

CRANFIELD UNIVERSITY

DUARTE O. DE M. A. RONDAO

**MULTIMODAL NAVIGATION FOR
ACCURATE SPACE RENDEZVOUS MISSIONS**



SCHOOL OF DEFENCE AND SECURITY
Centre for Electronic Warfare, Information and Cyber

Ph.D

Academic Year: 2020–2021

Supervisors: Prof Nabil Aouf, Prof Mark A. Richardson
May 2021

CRANFIELD UNIVERSITY

SCHOOL OF DEFENCE AND SECURITY
Centre for Electronic Warfare, Information and Cyber

Ph.D

Academic Year: 2020–2021

DUARTE O. DE M. A. RONDAO

**Multimodal Navigation for Accurate Space
Rendezvous Missions**

Supervisors: Prof Nabil Aouf, Prof Mark A. Richardson
May 2021

© Cranfield University 2021. All rights reserved. No part of
this publication may be reproduced without the written
permission of the copyright owner.

Acknowledgements

First, I want to express my gratitude to my supervisor and mentor, Prof Nabil Aouf, for his constant guidance, encouragement, and feedback. I have learnt an extraordinary amount during our technical discussions, from which most of the ideas presented herein have arisen, and without which this dissertation would not have been achievable.

I would also like to thank my co-supervisor, Prof Mark Richardson, for his continued advice and support, and for having reviewed my writing.

I gratefully acknowledge the funding received towards my PhD from the European Space Agency and Thales Alenia Space France. I particularly thank Dr Vincent Dubanchet for receiving me at Thales and for his review of my work conducted there.

I am indebted to the members of the Unmanned Autonomous System Lab and to my colleagues at Cranfield University, with a special mention to Akhil Kallepalli, Alejandro Dena, Amélie Grenier, Axel Beauvisage, Carole Belloni, Hugo Courtois, Raymond Vincent, Samuel Westlake, Shahmi Junoh, and Özgün Yılmaz. Thank you for motivating me every day.

Thank you to the 2019 cohort of “TASgaires” for making me feel welcome at Cannes, despite my ineptitude at speaking French: Amaury Knockaert, Étienne Puydebois, Grégory Gaudin, Nicky Rostan, Raúl Merino, Rodrigo González-O’Brien, Romain Mora, Tarik Errabih, and Victoire de Brosses. I will remember the quiz nights at Morrison’s fondly.

My thanks also go out to Maxwell Hogan at City, University of London, for his collaboration and support towards the final stage of my PhD.

Aside from my supervisory team and colleagues, I want to finally thank my friends and family back home, in particular my parents, Ana and Carlos, for their unwavering support and enthusiasm towards my decision to tackle this PhD. I am especially grateful to Rita for never ceasing to believe in me and for being there every step of the way.

ACKNOWLEDGEMENTS

Abstract

Relative navigation is paramount in space missions that involve rendezvousing between two spacecraft. It demands accurate and continuous estimation of the six degree-of-freedom relative pose, as this stage involves close-proximity-fast-reaction operations that can last up to five orbits. This has been routinely achieved thanks to active sensors such as lidar, but their large size, cost, power and limited operational range remain a stumbling block for en masse on-board integration. With the onset of faster processing units, lighter and cheaper passive optical sensors are emerging as the suitable alternative for autonomous rendezvous in combination with computer vision algorithms. Current vision-based solutions, however, are limited by adverse illumination conditions such as solar glare, shadowing, and eclipse. These effects are exacerbated when the target does not hold cooperative markers to accommodate the estimation process and is incapable of controlling its rotational state.

This thesis explores novel model-based methods that exploit sequences of monocular images acquired by an on-board camera to accurately carry out spacecraft relative pose estimation for non-cooperative close-range rendezvous with a known artificial target. The proposed solutions tackle the current challenges of imaging in the visible spectrum and investigate the contribution of the long wavelength infrared (or “thermal”) band towards a combined multimodal approach.

As part of the research, a visible-thermal synthetic dataset of a rendezvous approach with the defunct satellite Envisat is generated from the ground up using a realistic orbital camera simulator. From the rendered trajectories, the performance of several state-of-the-art feature detectors and descriptors is first evaluated for both modalities in a tailored scenario for short and wide baseline image processing transforms. Multiple combinations, including the pairing of algorithms with their non-native counterparts, are tested. Computational runtimes are assessed in an embedded hardware board.

From the insight gained, a method to estimate the pose on the visible band is derived from minimising geometric constraints between online local point and edge contour features matched to keyframes generated offline from a 3D model of the target. The combination of both feature types is demonstrated to achieve a pose solution for a tumbling target using a sparse set of training images, bypassing the need for hardware-accelerated real-time renderings of the model.

The proposed algorithm is then augmented with an extended Kalman filter which processes each feature-induced minimisation output as individual pseudo-measurements, fusing them to estimate the relative pose and velocity states at each time-step. Both the minimisation and filtering are established using Lie group formalisms, allowing for the covariance of the solution computed by the former

to be automatically incorporated as measurement noise in the latter, providing an automatic weighing of each feature type directly related to the quality of the matches. The predicted states are then used to search for new feature matches in the subsequent time-step. Furthermore, a method to derive a coarse viewpoint estimate to initialise the nominal algorithm is developed based on probabilistic modelling of the target's shape. The robustness of the complete approach is demonstrated for several synthetic and laboratory test cases involving two types of target undergoing extreme illumination conditions.

Lastly, an innovative deep learning-based framework is developed by processing the features extracted by a convolutional front-end with long short-term memory cells, thus proposing the first deep recurrent convolutional neural network for spacecraft pose estimation. The framework is used to compare the performance achieved by visible-only and multimodal input sequences, where the addition of the thermal band is shown to greatly improve the performance during sunlit sequences. Potential limitations of this modality are also identified, such as when the target's thermal signature is comparable to Earth's during eclipse.

Keywords

Spacecraft pose estimation; non-cooperative rendezvous; active debris removal; computer vision; thermal infrared imaging; multimodal imaging; optimisation; data fusion; Kalman filter; deep learning.

Contents

Acknowledgements	v
Abstract	vii
List of Figures	xiii
List of Tables	xvii
List of Acronyms	xix
Notation	xxiii
1 Introduction	1
1.1 Motivation	1
1.2 Vision-based Navigation	3
1.3 Space Rendezvous	7
1.4 The Camera as a Rendezvous Sensor	8
1.5 Active Debris Removal	11
1.6 Thermal Imaging	12
1.7 Overview of the Thesis	15
1.7.1 Objective	15
1.7.2 Outline and Contributions	16
1.7.3 Experimental Setup	17
1.7.4 Published and Submitted Manuscripts	22
1.7.5 Awards and Competitions	22
2 Theoretical Background and Tools	25
2.1 On Image Formation	25
2.1.1 Visible Wavelength	26
2.1.2 Infrared Wavelengths	26
2.1.3 Pinhole Camera Model	28
2.1.4 Perspective- n -Point Problem	32
2.1.5 Additional Considerations	32
2.2 On Frames of Reference	33
2.2.1 Spacecraft Body Frame	33
2.2.2 Earth-Centred Inertial Frame	35
2.2.3 Earth-Centred/Earth Fixed Frame	35

2.2.4	Local-Vertical/Local-Horizontal Frame	36
2.3	On Lie Groups	36
2.3.1	Composition of SE(3) Elements	40
2.3.2	Manifold Isomorphisms to SE(3)	40
2.4	On Artificial Intelligence	42
2.4.1	Machine Learning	42
2.4.2	Deep Learning	56
2.5	On Datasets	69
2.5.1	Open-Source Datasets for Motion Estimation	70
2.5.2	Simulation of Multimodal Trajectories for Space Rendezvous	71
3	Benchmarking of Detectors and Descriptors for Navigation	87
3.1	Motivation	87
3.2	Related Work	89
3.3	Methodology	91
3.3.1	Feature Detectors	91
3.3.2	Feature Descriptors	95
3.3.3	Performance Metrics	100
3.4	Experiments	103
3.4.1	Dataset	104
3.4.2	Ground Truth	104
3.4.3	Orbital Dynamics	107
3.4.4	Implementation	107
3.4.5	Results	108
3.5	Conclusions and Future Work	130
4	Markerless Multi-View Monocular Pose Estimation	133
4.1	Motivation	133
4.2	Related Work	135
4.3	Methodology	137
4.3.1	Local Feature Detection and Matching	138
4.3.2	Iteratively Reweighed Least Squares Minimisation	140
4.3.3	Biphasic Approach to Pose Estimation	144
4.4	Experiments	149
4.4.1	Dataset	149
4.4.2	Results	151
4.5	Conclusions and Future Work	155
5	Robust On-Manifold Optimisation	157
5.1	Motivation	157
5.2	Related Work	158
5.3	Methodology	160
5.3.1	Coarse Pose Estimation	162
5.3.2	Motion Estimation	171
5.3.3	Filtering	181
5.4	Experiments	191
5.4.1	Datasets	192

5.4.2	Testing	193
5.4.3	Evaluation of Coarse Pose Estimation	194
5.4.4	Evaluation of Fine Pose Estimation	200
5.5	Conclusions and Future Work	213
6	Pose Estimation for Multimodal Sequences via Deep Recurrent Convolutional Learning	217
6.1	Motivation	217
6.2	Related Work	220
6.3	Methodology	223
6.3.1	Architecture	224
6.3.2	Multistage Optimisation	231
6.3.3	Data Augmentation	236
6.4	Experiments	238
6.4.1	Datasets	238
6.4.2	Training	240
6.4.3	Testing	242
6.4.4	Evaluation of Multistage Optimisation	242
6.4.5	Evaluation of Recurrent Module	246
6.4.6	Evaluation of Multimodal Inputs	248
6.4.7	Summary of Performance on Astos Dataset	254
6.4.8	Evaluation on Laboratory Data	257
6.5	Conclusions and Future Work	259
7	Conclusion	263
7.1	Overview	263
7.2	Summary and Discussion	264
7.3	Future Work	267
	Bibliography	269

CONTENTS

List of Figures

1.1	Spacecraft relative pose estimation	2
1.2	Computer-generated renders of potential ADR approaches developed for the ESA e.Deorbit mission	12
1.3	Diversity of illumination conditions during RV and RVD/B	13
1.4	LIRIS-1 images of the ISS at multiple distances	14
1.5	Validation setup of the UASL at Cranfield University	18
1.6	Validation setup of the ASL at City, University of London	19
2.1	Pinhole camera model geometry	28
2.2	The PnP problem formulation	31
2.3	Geometric relationships between the chaser body frame \mathcal{F}_b , the camera frame \mathcal{F}_c , the target body frame \mathcal{F}_t in the context of landmark imaging in \mathcal{F}_Π	34
2.4	Definition of various orbital reference frames	35
2.5	A matrix Lie group \mathcal{G} and its tangent space at the identity, $T_I(\mathcal{G})$	37
2.6	Geometric correction and parametric fitting on manifolds	50
2.7	Sequential feature matching in false-color composite overlay	52
2.8	Comparison between the least squares cost function and the Huber and Tukey M-estimators	55
2.9	Diagram for a two-layer ANN	57
2.10	Typical ANN nonlinear activation functions	59
2.11	Comparison between the ReLU activation function and the leaky ReLU activation function	59
2.12	Illustration of a two-dimensional convolution operation	63
2.13	Diagram for a basic two-layer RNN	66
2.14	Characteristics of the SPEED dataset	71
2.15	Characteristics of the ASTOS dataset	72
2.16	Envisat body reference frame	73
2.17	Classical orbital elements	75
2.18	Schematic illustration of the three rotation scenarios of the target Envisat considered in the ASTOS dataset	76
2.19	Schematic illustration of the three guidance profiles of the chaser rendezvous with the target Envisat considered in the ASTOS dataset	79
2.20	Multimodal CAD models of Envisat for use in the ASTOS dataset and qualitative comparison with laboratory data acquired from UASL	80

LIST OF FIGURES

2.21	Key for the generated trajectories of the ASTOS dataset for chaser Guidance Profile 1	84
2.22	Key for the generated trajectories of the ASTOS dataset for chaser Guidance Profile 2	85
2.23	Key for the generated trajectories of the ASTOS dataset for chaser Guidance Profile 3	86
3.1	The domain of the present chapter	90
3.2	The difference of Gaussians pyramid structure	94
3.3	Distribution-based description and binary description	97
3.4	Illustration of feature correspondence	101
3.5	The ROC curve and its related rates	103
3.6	Key for the benchmarked trajectories of the ASTOS-B dataset	105
3.7	Homography computation example for a pair of wide baseline frames	106
3.8	Scenario specifications for the ASTOS-B dataset trajectory generation, centred in the chaser's body frame \mathcal{F}_b	107
3.9	Examples of detected features on hot case frames from the dataset . .	109
3.10	Examples of detected features on hot case frames from the dataset . .	109
3.11	Effect of contrast limited adaptive histogram equalisation on the visible cold case	110
3.12	Repeatability scores as a function of different benchmark parameters	112
3.13	Performance for ASTOS-B/01/01 rendezvous sequence: successive transformations, visible band, hot case	114
3.14	Performance for ASTOS-B/01/01 rendezvous sequence: large transfor- mations, visible band, hot case	115
3.15	Performance for ASTOS-B/02/01 rendezvous sequence: successive transformations, visible band, cold case	116
3.16	Performance for ASTOS-B/02/01 rendezvous sequence: large transfor- mations, visible band, cold case	117
3.17	Performance for ASTOS-B/01/02 rendezvous sequence: successive transformations, thermal infrared band, hot case	118
3.18	Performance for ASTOS-B/01/02 rendezvous sequence: large transfor- mations, thermal infrared band, hot case	119
3.19	Performance for ASTOS-B/02/02 rendezvous sequence: successive transformations, thermal infrared band, cold case	120
3.20	Performance for ASTOS-B/02/02 rendezvous sequence: large transfor- mations, thermal infrared band, cold case	121
3.21	Descriptor ROC curves for ASTOS-B/01/01 rendezvous sequence: visi- ble band, hot case	122
3.22	Descriptor ROC curves for ASTOS-B/02/01 rendezvous sequence: visi- ble band, cold case	124
3.23	Descriptor ROC curves for ASTOS-B/01/02 rendezvous sequence: ther- mal infrared band, hot case	125
3.24	Descriptor ROC curves for ASTOS-B/02/02 rendezvous sequence: ther- mal infrared band, cold case	126
3.25	Comparison of average feature detection times per frame	128

3.26	Comparison of average feature description and matching times per frame	129
3.27	Comparison of descriptor speed-up factors in different processors . . .	129
4.1	High level view of the proposed method	138
4.2	Visual representation of the employed multi-view sampling for training	145
4.3	Offline training	147
4.4	Temporally equidistantly sampled frames from the SIMPLESAT dataset	150
4.5	Offline-generated keyframes for validation of the proposed method with the SIMPLESAT dataset	151
4.6	Qualitative results for the SIMPLESAT dataset	152
4.7	Estimated and true values for the target relative position and orientation per axis in \mathcal{F}_c	152
4.8	Estimation error for the target relative translation and rotation . . .	153
4.9	Standard deviation of the target estimated relative translation and rotation	153
4.10	Figures of merit for point and edge features	154
5.1	High level view of the proposed method	161
5.2	The viewsphere for aspect sampling	164
5.3	Creating a training population	170
5.4	Minimisation of the reprojection function from the images of a randomly generated point cloud, averaged over 100 runs, with different amounts of contamination by outlying correspondences	180
5.5	Randomly sampled images from the BLENDER dataset	192
5.6	Histogram of results of the k -folds validation for the coarse pose classification on the BLENDER dataset	196
5.7	Cumulative performance of the k -folds validation for the coarse pose classification on the BLENDER dataset	197
5.8	Histogram of results of the k -folds validation for the coarse pose classification on the SPEED/TRAIN dataset	199
5.9	Nominal pose estimation errors for the ASTOS/G2/R1/VBAR/VIS/HOT trajectory	201
5.10	Feature statistics for nominal pose estimation sequence of the ASTOS/G2/R1/VBAR/VIS/HOT trajectory	202
5.11	Nominal velocity estimation errors for the ASTOS/G2/R1/VBAR/VIS/HOT sequence for the proposed framework	203
5.12	Qualitative results of the relative pose estimation for the ASTOS/G2/R1/VBAR/VIS/HOT sequence	203
5.13	Nominal pose estimation errors for the ASTOS/G1/R2/VBAR/VIS/HOT sequence	204
5.14	Nominal velocity estimation errors for the ASTOS/G1/R2/VBAR/VIS/HOT sequence for the proposed framework	204
5.15	Qualitative results of the relative pose estimation for the ASTOS/G1/R2/VBAR/VIS/HOT sequence	205
5.16	Image processing for the ASTOS/G2/R1/VBAR/VIS/COLD trajectory. . .	207

5.17	Nominal pose and velocity estimation errors for the ASTOS/G2/R1/- VBAR/VIS/COLD trajectory	208
5.18	Feature statistics for nominal pose estimation sequence of the ASTOS/- G2/R1/VBAR/VIS/COLD trajectory	209
5.19	Pose estimation errors for the UASL dataset	210
5.20	Velocity estimation errors for the UASL dataset	210
5.21	Results of the relative pose estimation for the UASL dataset	210
5.22	Sample keyframes rendered from the reconstructed target spacecraft of the SPEED dataset	212
5.23	Results of the relative pose estimation for the SPEED/REAL-TEST subset	213
6.1	ChiNet system overview	225
6.2	The Darknet CNN architecture	226
6.3	LSTM block diagram	228
6.4	ChiNet recurrent module design	229
6.5	Model points \mathbb{P} of the ASTOS dataset for use with ChiNet	235
6.6	Image augmentation transform operations in use with ChiNet	237
6.7	Sample images from the CITY/APPROACH-SLOW	240
6.8	Comparison of estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences in terms of training stages used	243
6.9	Effect of multistage optimisation illustrated on the ASTOS/G2/R1/- VBAR/HOT/VIS sequence	244
6.10	Comparison of estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences in terms of recurrence	247
6.11	Effect of recurrent module illustrated on the ASTOS/G3/R2/HOT/VIS sequence	248
6.12	Comparison of estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences in terms of imaging modality	249
6.13	Effect of multimodality illustrated on the ASTOS/G3/R2/HOT/ trajectory	251
6.14	Comparison of estimated position and attitude errors over time on the G2/R1/VBAR/COLD rendezvous sequence from the ASTOS dataset in terms of imaging modality	252
6.15	Effect of multimodality illustrated on eclipse sequences	253
6.16	Estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences for the complete multimodal DRCNN .	255
6.17	Qualitative pose estimation performance on frames of two sample ASTOS dataset rendezvous sequences for the complete multimodal DRCNN	256
6.18	Estimated position and attitude errors over time on the CITY dataset laboratory test rendezvous sequences	258
6.19	Qualitative pose estimation performance on frames of the CITY dataset laboratory test rendezvous sequences for the complete multimodal DRCNN	262

List of Tables

1.1	Comparison between flight-proven remote sensing devices	9
1.2	NASA measurement accuracy requirement for the Video Guidance Sensor	11
1.3	Technical data – mvBlueFOX MLC200wC	20
1.4	Technical data – DFK 22BUC03	20
1.5	Technical data – FLIR Vue Pro R	21
2.1	Envisat set of orbital elements at time $\tau = \tau_0$ for the ASTOS dataset generation	74
2.2	The three rotation scenarios of the target Envisat considered in the ASTOS dataset	76
2.3	Technical data – mvBlueFOX-MLC 202b	82
2.4	Technical data – FLIR Tau2	82
3.1	Characteristics of feature descriptors	96
3.2	Summary of experiments in Chapter 3	103
3.3	Simulated camera properties for the ASTOS-B dataset	104
3.4	The BeagleBone Black wireless single board computer	108
3.5	Average detection times per feature	127
3.6	Average description times per feature	127
4.1	Simulated camera properties for the SIMPLESAT dataset.	149
4.2	Average pose estimation computation times	155
5.1	Summary of experiments in Chapter 5	192
5.2	Settings used for k -folds validation of the coarse pose classification module on the BLENDER dataset	195
5.3	Average expected attitude error for the coarse pose classification on the BLENDER dataset	196
5.4	Mean computational execution times per image for the coarse pose classification on the BLENDER dataset	198
5.5	Average expected attitude error for the coarse pose classification on the SPEED/TRAIN dataset	199
5.6	Pose estimation pipeline configuration and numerical settings used for the experiments on the ASTOS and UASL datasets	200
5.7	Mean computational execution times per image for the fine pose estimation on the ASTOS dataset	208

LIST OF TABLES

5.8	Breakdown of mean computational execution times per image for the proposed fine pose estimation method on the ASTOS dataset	211
5.9	Achieved SPEED/REAL-TEST subset $\delta\tilde{T}_{\text{SPEC}}$ score in the context of the scores obtained by the SPEC top-5 rankers in this metric	212
6.1	Summary of experiments in Chapter 6	238
6.2	Train/test data split on the ASTOS dataset	241
6.3	Base learning rates used in the training of the complete multimodal DRCNN pipeline	241
6.4	Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of training stages used .	245
6.5	Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of recurrence	246
6.6	Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of imaging modality . .	250
6.7	Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of imaging modality . .	252
6.8	Summary of position and attitude error statistics on all ASTOS dataset rendezvous test sequences for the complete DRCNN pipeline	254
6.9	Summary of position and attitude error statistics on the CITY dataset laboratory test rendezvous sequences	259

List of Acronyms

6-DOF six degrees-of-freedom	togram equalisation
ADR active debris removal	CMOS complementary metal-oxide semi-conductor
AI artificial intelligence	CNES National Centre for Space Studies
AKAZE Accelerated KAZE	CNN convolutional neural network
ANN artificial neural network	COTS commercial-off-the-shelf
ASL Autonomous Systems Laboratory	CPU central processing unit
ATV Automated Transfer Vehicle	DNN deep neural network
AUC area under curve	DOF degree-of-freedom
BBB BeagleBone Black	DoG difference of Gaussians
BoVW bags-of-visual-words	DRCNN deep recurrent convolutional neural network
BPTT backpropagation through time	DS1 Deep Space 1
BRIEF Binary Robust Independent Elementary Features	ECC enhanced correlation coefficient
BRISK Binary Robust Invariant Scalable Keypoints	ECEF Earth-centred/Earth-fixed
CAD computer-aided design	ECI Earth-centred inertial
CCD charge-coupled device	EDL entry, descent and landing
CDF cumulative distribution function	EDLines Edge Drawing Lines
CenSurE Centre Surround Extrema	EKF extended Kalman filter
CLAHE contrast limited adaptive his-	ESA European Space Agency
	ETS-VII Engineering Test Satellite-VII

LIST OF ACRONYMS

FAST Features from Accelerated Segment Test	ISS International Space Station
FC fully connected	JAXA Japan Aerospace Exploration Agency
FLANN Fast Library for Approximate Nearest Neighbours	KDE kernel density estimation
FOV field of view	LED light-emitting diode
FPA focal plane array	LEO low Earth orbit
FPS frames per second	LIOP Local Intensity Order Pattern
FREAK Fast Retina Keypoint	LIRIS Laser and Infra-Red Imaging Sensors
GAN generative adversarial network	LM Levenberg-Marquardt
GEO geostationary orbit	LoG Laplacian of Gaussian
GFTT Good Features To Track	LOS line-of-sight
GMM Gaussian mixture modelling	LS least squares
GMST Greenwich mean sidereal time	LSTM long short-term memory
GNC guidance, navigation and control	LVLH local-vertical-local-horizontal
GNFIR Goddard Natural Feature Image Recognition	LWIR long-wavelength infrared
GNSS global navigation satellite system	MAD median absolute deviation
GPS Global Positioning System	ML machine learning
GPU graphics processing unit	MLE maximum likelihood estimate
HST Hubble Space Telescope	MLI multi-layer insulation
ILSVRC ImageNet Large Scale Visual Recognition Challenge	MLP multilayer perceptron
IMU inertial measurement unit	MSE mean squared error
IP image processing	MWIR medium-wavelength infrared
IR infrared	NASA National Aeronautics and Space Administration
IRLS iteratively reweighed least squares	NCRV non-cooperative rendezvous

NLLS nonlinear least squares	SGD stochastic gradient descent
NN nearest-neighbour	SIFT Scale Invariant Feature Transform
NNDR nearest-neighbour distance ratio	SLAM simultaneous localisation and mapping
NORAD North American Aerospace Defense Command	SPEC Satellite Pose Estimation Challenge
ORB Oriented FAST and Rotated BRIEF	SPEED Spacecraft PosE Estimation Dataset
PnP perspective- n -point	SURF Speeded-Up Robust Features
P3P perspective-3-point	SVD singular value decomposition
PDF probability density function	SVM support vector machine
PIL processor-in-the-loop	SWIR short-wavelength infrared
PMF probability mass function	TAS-F Thales Alenia Space France
PRISMA Hyperspectral Precursor of the Application Mission	TLE two-line element
RANSAC Random Sample Consensus	TriDAR Triangulation and Lidar Automated Rendezvous and Docking
ReLU rectified linear unit	UASL Unmanned Autonomous Systems Laboratory
RGB red-green-blue	UAV unmanned aerial vehicle
RGBT red-green-blue-thermal	URSO Unreal Rendered Spacecraft On-Orbit
RMSE root mean squared error	VO visual odometry
RNN recurrent neural network	VSLAM visual simultaneous localisation and mapping
ROC receiver operating characteristics	VVS virtual visual servoing
RV rendezvous	XOR exclusive-OR
RVD/B rendezvous and docking or berthing	ZM Zernike moment
RVS Rendezvous and Docking Sensor	
SAR synthetic aperture radar	
SFM structure from motion	

LIST OF ACRONYMS

Notation

This document follows the Style Guide for NASA History Authors and Editors (Garber, 2012) for referencing celestial bodies. With respect to mathematics, a notation based on I. Goodfellow et al. (2016), R. M. Murray et al. (1994), and Barfoot (2017) is adopted and tabled below.

Numbers and Arrays

a	A scalar (integer or real)
\mathbf{a}	A column vector
\mathbf{A}	A matrix
\mathbf{A}	A tensor
\mathbf{I}_n	Identity matrix with n rows and n columns
\mathbf{I}	Identity matrix with dimensionality implied by context
$\mathbf{0}_{n \times m}$	Matrix of zeros with n rows and m columns
$\mathbf{1}_{n \times m}$	Matrix of ones with n rows and m columns
a	A scalar random variable
\mathbf{a}	A vector-valued random variable

Sets

\mathbb{A}	A set
\mathbb{R}	The set of real numbers
$\{1, \dots, n\}$	The set of all integers between 1 and n
$[a, b]$	The real interval including a and b
$[a, b[$	The real interval including a but excluding b

$\mathbb{A} \setminus \mathbb{B}$ Set subtraction, i.e., the set containing the elements of \mathbb{A} that are not in \mathbb{B}

Indexing

a_i Element i of vector \mathbf{a} , with indexing starting at 1

$A_{i,j}$ Element i, j of matrix \mathbf{A}

$\mathbf{A}_{i,:}$ Row i of matrix \mathbf{A}

$\mathbf{A}_{:,i}$ Column i of matrix \mathbf{A}

$A_{i,j,k}$ Element (i, j, k) of a 3D tensor \mathbf{A}

$\mathbf{A}_{::,i}$ 2D slice of a 3D tensor

Linear Algebra Operations

\mathbf{A}^\top Transpose of matrix \mathbf{A}

$\mathbf{A} \odot \mathbf{B}$ Element-wise product of \mathbf{A} and \mathbf{B}

$\text{kron}(\mathbf{A}, \mathbf{B})$ Kronecker product of \mathbf{A} and \mathbf{B}

$\det(\mathbf{A})$ Determinant of \mathbf{A}

$\text{Tr}(\mathbf{A})$ Trace of \mathbf{A}

$\text{diag}(\mathbf{a})$ A square, diagonal matrix with diagonal entries given by \mathbf{a}

$\text{blockdiag}(\mathbf{A}, \mathbf{B})$ A square, diagonal matrix with diagonal blocks given by \mathbf{A} and \mathbf{B}

$\text{vec}(\mathbf{A})$ A vector built from stacking the columns of \mathbf{A}

\mathbf{a}^\wedge A 3×3 skew-symmetric matrix with entries given by $\mathbf{a} \in \mathbb{R}^3$

\mathbf{A}^\vee The inverse mapping of \mathbf{a}^\wedge

Calculus

$\frac{dy}{dx}$ Derivative of y with respect to x

$\frac{\partial y}{\partial x}$ Partial derivative of y with respect to x

$\nabla_{\mathbf{x}} y$ Gradient of y with respect to \mathbf{x}

$\frac{\partial f}{\partial \mathbf{x}}$ Jacobian matrix $\mathbf{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$\int f(\mathbf{x}) d\mathbf{x}$ Definite integral over the entire domain of \mathbf{x}

$\int_{\mathbb{S}} f(\mathbf{x}) d\mathbf{x}$	Definite integral with respect to \mathbf{x} over the set \mathbb{S}
$f \circ g$	Composition of the functions f and g
$\log x$	Natural logarithm of x
$\ \mathbf{x}\ _p$	L^p norm of \mathbf{x}
$\ \mathbf{x}\ $	L^2 norm of \mathbf{x} ; also $\ \mathbf{x}\ _2$
$ x $	Absolute value of x

Probability

$p(a)$	A probability distribution over a discrete or continuous variable
$\mathbb{E}[f(x)]$	Expectation of $f(x)$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Kinematics

τ	Continuous time
κ	Discrete time
\underline{a}	A vector quantity in three dimensions
$\underline{\mathcal{F}}_a$	A vectrix representing a frame of reference in three dimensions
$\text{SO}(3)$	The special orthogonal group, a matrix Lie group used to represent rotations
$\mathfrak{so}(3)$	The Lie algebra associated with $\text{SO}(3)$
$\text{SE}(3)$	The special Euclidean group, a matrix Lie group used to represent poses
$\mathfrak{se}(3)$	The Lie algebra associated with $\text{SE}(3)$
Ad	An operator producing the adjoint of an element from $\text{SO}(3)$ or $\text{SE}(3)$
ad	An operator producing the adjoint of an element from the Lie algebra of $\text{SO}(3)$ or $\text{SE}(3)$
\oplus	An operator producing the composition of two elements of $\text{SE}(3)$; also the composition between an element of $\text{SE}(3)$ and a point

NOTATION

\ominus	The inverse operator to \oplus
\boxplus	An operator producing the composition of an element of $\text{SE}(3)$ with a displacement screw in \mathbb{R}^6
\boxminus	The inverse operator to \boxplus

CHAPTER 1

Introduction

In this introductory chapter, the reader is made acquainted with the current context from which this work arises and is presented with an overview of how computer vision techniques adopted or developed herein were blended with space science to address the extant obstacles in spacecraft rendezvous missions. In particular, vision-based navigation is compared with its main competitor, i.e. lidar, drawbacks of visible wavelength cameras are discussed, and thermal infrared imaging is introduced as a candidate modality to improve reliability and accuracy.

1.1 Motivation

THIS dissertation aims to tackle the fundamental problem of providing a reliable and efficient solution for image-based *spacecraft relative pose estimation*. The objective of this task consists in determining the rigid transformation between two space bodies – one of which is controllable and carries the navigation sensors – in terms of their relative position and attitude (as shown in Fig. 1.1). Together, these two quantities define the six degrees-of-freedom (6-DOF) relative *pose*. Specifically, the navigator must methodically select and extract raw visual cues from a two-dimensional image of the *target* produced by the on-board camera using *image processing* (IP) techniques, which are then subjected to pattern recognition algorithms to recover the pose. In the present context, “reliable” means capable of providing an accurate solution that fulfils the subsystem’s requirements, whereas “efficient” refers to functioning in a manner that minimises the subsystem’s design budget, both physically and computationally. This remains an arduous problem, as space is one of the most extreme environments imaginable, and the target might be free to take up a multitude of possible pose configurations.

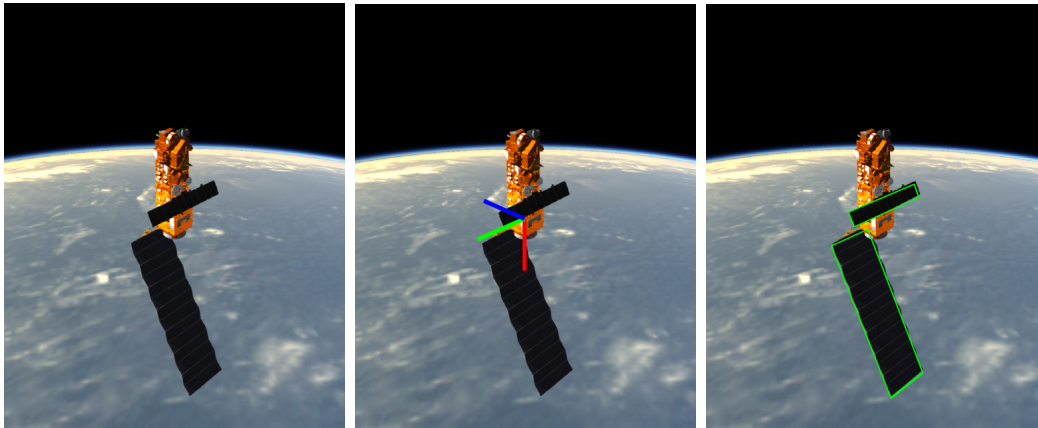


Figure 1.1: Spacecraft relative pose estimation. The goal is to take an image of the target captured by the on-board camera and, from it, infer the translation and rotation that aligns the target’s frame of reference with the chaser’s. The solution portrayed in this figure has been obtained with algorithms developed in this thesis. *(Left)* Target spacecraft. *(Centre)* The estimated origin and orientation of the target’s frame of reference. *(Right)* The 3D model of the target projected according to the attained solution.

In Earth-based applications, cameras are often deployed as navigation sensors for unknown environments. The objective is to estimate the vehicle’s egomotion in the uncharted world, to recover its three-dimensional structure, or both. Whereas this can and has been applied to the context of space, in many situations several a priori assumptions about the target body can be made. For example, most of Earth’s satellites are man-made crafts, fruit of previous and current missions, of which material, textural, and structural information is known. Indeed, this thesis is concerned with extending the state-of-the-art for the spacecraft relative pose estimation problem by formulating it as a *model-based* problem with the goal of deriving techniques to retrieve the robust and global solution.

Model-based formulations are not without their own challenges: often, information captured on-board must be matched to a reference that is fundamentally different in terms of imaging conditions. This thesis identifies and addresses these difficulties present in the road map towards pose estimation, and the developed implementations cover feature detection and matching, probabilistic modelling, shape-based classification, outlier rejection, optimisation, Kalman filtering, and deep learning. The focus is placed on *monocular* systems, in which images are acquired through a single camera, or equivalent. These have the potential for use at long range, but the scale cannot be recovered (as shown in Chap. 2); this dissertation explores the role of such model-based strategies in overcoming this challenge. This is in opposition to *stereo* systems, which can naturally extract 3D information from the scene by

triangulating points in the images acquired by both cameras. However, these become problematic at long range when the disparity is small and even minimal amounts of image noise can compromise the accuracy, limiting their use in spaceborne scenarios.

Relative pose estimation is a vital requirement for a space *rendezvous* (RV), i.e. to bring two space objects within close proximity of one another. Be it to service a space station, accomplish formation flying, or collect space junk, numerous RV tasks involve the management of relative velocities and minimal reaction times, justifying the need for autonomous operations. Others, such as landing on a small body, occur far away from Earth such that ground operators cannot reliably navigate the spacecraft in real-time. In general, the democratisation of space is evolving towards the mass-production of hardware and investment in research and development, cutting back on the reliance on ground control to save costs. Therefore, autonomous RV is set to become the norm rather than the exception. When the target is *non-cooperative*, redundancy for relative navigation sensors becomes a substantial requirement; accordingly, *passive* sensors such as digital cameras present themselves as a lower costing and lighter alternative to bulky *active* hardware.

Using regular cameras operating on the visible spectrum for navigation warrants particular care and combination with adequate IP due to the particular imaging conditions experienced in space. Often, orbiting spacecraft are covered in highly reflective materials which affect its perception as a target, and pointing a camera directly at the Sun will result in sensor failure. Conversely, during eclipse periods, important features on the target might not be visible at all. As such, alternative imaging modalities beyond the visible could improve relative navigation performance, and *multimodality* is thus investigated in this dissertation.

Nevertheless, optical cameras are gaining popularity as the suitable sensor for autonomous relative navigation in space due their attractive sizing and the fact that acquired images can be compressed to be processed aboard the spacecraft, avoiding the requirement of ground-processing the data. Additionally, the same sensor feeds used to fulfil the mission scientific objectives can be used for navigation and mapping, alleviating size and cost constraints associated with additional sensors.

1.2 Vision-based Navigation

In the field of *mobile robotics*, navigation is a necessary tool towards the fulfilment of autonomy. Undoubtedly, in order to accurately make a decision that leads it from one state to the next until the final goal is achieved, a machine must know how to interact with its environment, i.e. how to navigate it. Despite the fact that a spacecraft on an RV mission could technically be considered a mobile robot, and that

both subjects have shared common strategies on employing vision for navigation, they began by carving out individual paths. This section briefly outlines the main points of vision-based navigation for the former, and Section 1.4 does it specifically for space RV .

Detection and matching of *features* is the fulcrum of navigation solutions based on computer vision¹ (Torr and Zisserman, 2000). A feature is purely an interesting and distinguishable part of an image. In particular, Szeliski (2011) defines *interest points*, also known as *keypoints*, as one of the fundamental and most popular types of features; these are “specific locations in the images, such as mountain peaks, building corners, doorways, or interestingly shaped patches of snow... often described by the appearance of patches of pixels surrounding the point location”. Many different types of features exist, such as edges, circles, or even colours, but keypoints have remained consistently popular for navigation thanks to IP developments to efficiently detect and match them across images and to the clarity in establishing geometric constraints.

Vision-based navigation for mobile robots had its genesis with Moravec (1980), who developed a combined localisation-path-planning-collision-avoidance approach. An “interest operator” was applied to images from both cameras of a mounted stereo setup to detect keypoints, which were then matched and triangulated to find the corresponding 3D points in the scene. If clusters of these 3D points were determined to be too close to the robot, the path planner would send out a command to go around them for the next time-step. By matching these keypoints not only across cameras but also across time-steps, the vehicle’s own motion could be derived. This technique, termed *visual odometry* (VO), is one of the two archetypes of *model-free* navigation strategies, as no prior knowledge of the environment’s structure is needed to complete the task. The obvious limitation of VO is that only the vehicle’s own trajectory, i.e. its egomotion, is computed, and hence it is only usable in scenarios where scene understanding is not required. Furthermore, VO solutions tend to exhibit long-term trajectory drift due to errors in feature matching or triangulation. This is often mitigated by storing a sparse set of points from recent frames, whose 3D coordinates are jointly optimised with previous poses in a process termed bundle adjustment (Triggs et al., 2000). The term “visual odometry” was popularised by Nister et al. (2004), who formalised the framework for both monocular and stereo systems, and addressed the rejection of spurious feature matches. Overall, it is a less burdensome algorithm compared to other methods which do perform mapping, and

¹Whereas it may be accepted in the space literature to define “vision” as an umbrella term that also covers active sensors, in this dissertation it shall be exclusively used in reference to cameras.

has been adopted in circumstances where computational power is limited, such as the navigation of Mars exploration rovers (Maimone et al., 2007).

The second class of model-free navigation strategies is derived from *simultaneous localisation and mapping* (SLAM), which consists in navigating a robot through an unknown environment while building a map of it and avoiding obstacles, and had been traditionally accomplished using laser rangefinders (Thrun et al., 2005). In computer vision, the problem is known as *structure from motion* (SFM), involving the joint estimation of unordered camera poses and the geometry of the scene, although it is typically solved offline (Szeliski, 2011). The solution to the SLAM problem in real-time, with a monocular camera as the sole sensor, was devised by Davison (2003; 2007): as in VO, keypoints are associated to map landmarks via frame-to-frame matching, but their depth (and, effectively, 3D coordinates) are modelled according to a probabilistic extended Kalman filter (EKF) framework, in which the current robot pose is also included in the state vector. This formulation includes a motion model for the camera, and allows for the map to evolve dynamically due to the EKF updates, whereby the state is augmented when new features are observed, or likewise reduced if necessary. Importantly, the EKF formulation offers a natural way to estimate the uncertainty of the solution.

The major shortcoming of monocular SLAM is that the navigation solution can only be recovered up to an arbitrary scale. EKF-SLAM, in particular, was also afflicted by an ever-growing state vector (and covariance matrix) in proportion to the size of the map, which had to be limited before a certain number of landmarks was reached to avoid computational bottlenecks.

Later on, Karlsson et al. (2005) addressed these issues by employing FastSLAM's factored solution for larger map support (Montemerlo et al., 2002), and a second camera to form a stereo setup and derive scene depth information directly by triangulating features, solving for the absolute scale. The formulation made use of more modern keypoint descriptors (Lowe, 2004) rather than traditional feature patches and coined the term *visual simultaneous localisation and mapping* (VSLAM), despite also relying on data from the robot's wheel encoder odometry.

In an effort to further increase map building efficiency and volume, Klein and D. Murray (2007) broke away from the Bayesian formulation of VSLAM and introduced the Parallel Tracking and Mapping (PTAM) algorithm, which employed bundle adjustment to certain snapshots of the trajectory, or keyframes. Real-time performance was achieved by splitting tracking and mapping in two distinct tasks, each processed by a parallel thread of a dual-core central processing unit (CPU). The end result provided detailed maps bearing thousands of features with state-of-the-art

accuracy, thanks to the capability running both global and local bundle adjustment on its own thread.

Further research in the field has since continued to evolve with strategies such as visual loop-closure to reduce drift (Ho and Newman, 2006) and relocalisation after tracking failure (Williams et al., 2007). The state-of-the-art in feature-based SLAM condenses most of the improvements by its predecessors into a single pipeline which has been baptised as ORB-SLAM (Mur-Artal et al., 2015). The prefix refers to Oriented FAST and Rotated BRIEF, the feature detection and descriptor adopted to achieve real-time performance on the IP front-end, and which is explored further in Chapter 3. Furthermore, ORB-SLAM adds a third parallel thread exclusively dedicated to loop closure, and provides fully automatic initialisation.

It is clear that much of SLAM’s performance, especially in terms of mapping, was brought about by astute parallelisation strategies. Indeed, all of the algorithms describe above have been designed for desktop CPUs, making their implementation on mobile hardware more challenging, particularly for embedded systems which do not support multithreading. Added to the fact that, in close-range RV with artificial space objects, the “scene” normally consists of a single body of which the structure is known, this is a plausible explanation as to why most of such missions have bet on model-based methods instead.

Model-based methods are concerned with solving the task of determining the pose of the scene’s frame of reference relative to the camera’s frame of reference, given a model of the scene made up of 3D reference points and a 2D image of those points, without assuming any prior information on the correspondence between either the pose or feature correspondences. As described by David et al. (2004), this *model-to-image registration problem* entails, in fact, two coupled problems: the pose problem, consisting in retrieving the pose based on the set of 2D-3D corresponding points; and the correspondence problem, which involves establishing the matches between image and scene features. This approach therefore brings its own set of challenges which are mainly related to bridging the gap brought about by the extra dimension of the scene relative to its image. Additionally, model-based techniques normally require an initialisation strategy. Chapter 4 further reviews the evolution of model-based strategies, from their early industrial robotics applications until the spillover into the field of spacecraft relative navigation.

Either navigation strategy can alternatively be formulated as a *direct method*, whereby the parameters to be estimated are derived from measurable quantities at each image pixel rather than at selected local interest points (Irani and Anandan, 2000). Recently, direct methods have witnessed a resurgence due to advances in

deep learning, particularly *convolutional neural networks* (CNNs; LeCun et al., 1989). This type of architecture is capable of providing a navigation solution directly from a series of learned operations on image inputs and have been applied to VSLAM (Tateno et al., 2017), VO (S. Wang et al., 2017), and model-based methods (Kendall, Grimes, et al., 2015).

1.3 Space Rendezvous

A space rendezvous, or simply *RV*, is a set of operations that progressively manoeuvre an active vehicle (the *chaser*) into the vicinity of, and finally into contact with, a generally passive object (the target; Fehse, 2003; Wertz and Bell, 2003). Every *RV* begins with a launch, which typically involves the chaser departing Earth and being injected into orbit.² Initially, the chaser and the target are expected to be in entirely different orbits, tens of thousands of kilometres apart. Therefore, the chaser must first engage in absolute navigation, which entails the determination of its own orbit and inertial attitude. Orbit determination can be performed autonomously as part of the guidance, navigation and control (GNC) subsystem, via ground tracking, or by propagation of the ephemerides (Wertz, 1999). Earthbound *RVs* can typically make use of global navigation satellite systems (GNSS) to obtain complete positional information. For interplanetary trajectories, inertial measurement units (IMU) can provide integration-based position and velocity, but the solution becomes degraded over time, so typically other solutions are required (e.g. line-of-sight [LOS] measurements of other planets or asteroids). Attitude determination can be performed using rate gyros, magnetometers, Sun sensors, horizon sensors, star trackers, or a combination thereof (Wie et al., 2014).

Once both orbits are known, the transfer burn that sets the chaser on course towards the target can be calculated and executed. During this transfer orbit, the bodies are out of sight and out of contact from each other, meaning that absolute navigation is still in use. Once potential LOS is established, the far-range rendezvous is initiated. At this point, if the target is *cooperative*, either passively (e.g. has a fixed attitude, interfaces for communication or sensing) or actively (e.g. performs actual manoeuvres to facilitate the *RV*), both crafts may begin communication (Fehse, 2003). Alternatively, the target is said to be non-cooperative when it bears no such supportive equipment (Wertz and Bell, 2003).

The close-range rendezvous phase marks the beginning of the chaser's final approach to the target. It is divided into two subphases (Fehse, 2003):

²Special cases exist wherein the target may be launched towards the chaser, which is already in orbit (e.g. the Apollo 11 Lunar Module ascent to rendezvous back with the Command and Service Module; see Bennett, 1970).

- **Closing:** typically begins at a relative distance of a few kilometres to 100 m. Aims to attain the conditions leading to the final approach corridor. One of two methods of approach are usually followed: horizontally along the target spacecraft’s inertial velocity vector, or vertically along the radial vector; these are named *V-bar* and *R-bar* approaches, respectively (see Chap. 2, § 2.2 for a definition of the frames of reference involved).
- **Terminal:** begins when the chaser is between 10–100 m away from the target. The subsequent chain of operations will depend on the nature of the mission. In inspection missions, the chaser does not actually make contact with the target, but rather keeps a short separation distance, possibly circumnavigating it, for observation or study. For a scenario in which the target is another spacecraft, the mission may culminate in mating; in this case the chaser will proceed closer, reaching residual velocities, and ultimately match the necessary pose for contact, performing a *rendezvous and docking or berthing* (RVD/B).


A close-range **RV** marks the switch to the chaser’s relative navigation sensors. As the distance between the two bodies dwindles, attaining an accurate solution grows more important, requiring the estimation of the complete 6-DOF pose.³ For this reason, active sensors have traditionally been utilised on orbit. These function by irradiating the target with a signal and then registering the reflection that is bounced back. The main active sensors are radar and lidar⁴ (Chesley et al., 1999). The latter is a precise and flight-mature sensor capable of working at frequencies up to approximately 10 Hz, but is characterised by high mass, power consumption, and cost. Furthermore, most lidars used for **RV** are of the scanning type, which is affected by the stability of the platform (Wie et al., 2014). A short survey on the use of lidar for spacecraft rendezvous is provided by J. A. Christian and Cryan (2013).

1.4 The Camera as a Rendezvous Sensor

Table 1.1 provides some figures of merit for two scanning lidars which have been used aboard the European Space Agency’s (ESA) Automated Transfer Vehicle (ATV; Roux and da Cunha, 2004) for RVD/B with the International Space Station (ISS): the Rendezvous and Docking Sensor (RVS), and the LIRIS-2 (Laser and Infra-Red Imaging Sensors). For comparison, the on-board camera of the Earth-orbiting LightSail 2 CubeSat (Spencer et al., 2021) is also described. This camera was not

³Now that the estimation of this quantity is understood to exist in the context of relative navigation, it will be mostly referred to as simply “the pose” throughout this dissertation.

⁴A portmanteau of “light detection and ranging” which is sometimes stylised as an acronym instead.

Table 1.1: Comparison between flight-proven remote sensing devices (Fehse, 2003; Jena-Optronik, 2015; Micron Technology, 2006; Wagner, 2016; Wie et al., 2014).


Sensor		RVS scanning lidar	LIRIS-2 scanning lidar	MT9D131 visible camera
Manufacturer		Jena-Optronik	Jena-Optronik	Micron Technology ¹
Type		Active	Active	Passive
Power	W	35–75	25–55	<0.15
Mass	kg	14.7	13.3	0.01 ²
Volume	L	~27	~22	~0.009 ²
Max. range (cooperative)	km	1.3	2.5	0.2 ³
Max. range (non-cooperative)	km	n/a	~0.25	~100 ³
Max. frequency	Hz	1	3	30

¹ Now Onsemi.² Not including the lens.³ Subject to field of view (FOV) and IP algorithms.

used for GNC, but is similar in performance to those used in the experimentation validations of the solutions proposed within this dissertation, and highlights the low system requirements and, by extension, cost when compared to lidar, justifying its use for autonomous navigation in space.

Indeed, visible cameras have already proven their use for far-range navigation in space. In 1976, the Viking Orbiters 1 and 2 demonstrated that in-situ imaging of Mars’ natural satellites, Phobos and Deimos, could be used to perform fly-by manoeuvres (Duxbury and Callahan, 1988). In the following decade, the two Voyager spacecraft would use similar techniques along with radio tracking to navigate planets beyond the asteroid belt (Synnott et al., 1986). The maiden voyage of autonomous navigation was the Deep Space 1 (DS1) mission (Bhaskaran et al., 1998), which used a camera to provide LOS measurements of several “beacon” asteroids, alongside their predicted heliocentric orbits, to self-localise in an inertial frame, culminating in a fly-by with comet 19P/Borrelly in 2001. Four years later, the Deep Impact probe (Mastrodemos et al., 2005) would use an enhanced version of DS1’s autonomous navigation suite to estimate its position relative to the Tempel 1 comet with a camera

and successfully deploy on it an impactor craft. More recently, in 2014 the Rosetta mission (Castellini et al., 2015) used visual navigation to rendezvous with the comet 67P/Churyumov–Gerasimenko.

For close-range *RV*, however, cameras have essentially been used as a supporting sensor for cooperative scenarios, requiring fiducial markers on the target to estimate the pose (Tietz and T. E. Richardson, 1983). The typical design of such a system involves making these markers out of retroreflective material which is then illuminated by a light source on the chaser to facilitate detection by the on-board camera. Since the markers are arranged in a known pattern, their detected positions on the image can be used to compute the pose using model-based methods (Fehse, 2003). This principle has been used in multiple autonomous operations demonstrators with little variation, namely for Japan Aerospace Exploration Agency’s (JAXA) Engineering Test Satellite-VII (ETS-VII), which performed the first autonomous *RVD/B* between two unmanned spacecraft (Kawano et al., 2001); the Orbital Express mission (Leinz et al., 2008); and the Hyperspectral Precursor of the Application Mission (PRISMA; Bodin et al., 2012).

Efforts are being made towards the transition to cameras for use in close-range non-cooperative rendezvous (*NCRV*) scenarios. The SM4 mission to the Hubble Space Telescope (HST) performed in-flight tests of the National Aeronautics and Space Administration’s (NASA) Goddard Natural Feature Image Recognition (GNFIR) algorithm, which was able to track Hubble’s pose via model-based matching of contour edges of a 3D model made to resemble its shape projected onto the acquired images (Naasz et al., 2010). The algorithm was not tested on a tumbling case, though, as the relative attitude hardly changed. The method was also found to be highly dependent on the expected illumination conditions, failing on one of the three test trajectories. A keypoint-based method was also tested, but was unable to produce a continuous estimation track in any trajectory. Promising results have led NASA to develop Raven (Strube et al., 2015), an experiment aboard the ISS aimed at maturing GNFIR, with an augmented sensor suite including a thermal camera and a lidar, and testing the algorithm on incoming visiting vehicles.

Contrary to the mating phase, there are no explicit estimation accuracy requirements for the close-range *RV* phase. Fehse (2003) states that the rule of thumb is that the accuracy must be of the order of 1% of range or better. Table 1.2 summarises, as an example, NASA’s close-range requirements for the cooperative Video Guidance System, part of their Automated Rendezvous & Capture system (Howard et al., 1999).

Table 1.2: National Aeronautics and Space Administration (NASA) measurement accuracy requirement for the Video Guidance Sensor (Howard et al., 1999).

Operating Range (m)	Range Offset (mm)	Lateral Offset (mm)	Roll/Pitch/Yaw (deg)
10.5–30	±300	±100	±2
30–50	±1000	±200	±3
50–110	±3000	±2000	±5

1.5 Active Debris Removal

In 2007, the number of catalogued space objects orbiting Earth suddenly grew by approximately 26 % (McKnight, 2010). It has since been shown that this hike was due to the intentional destruction of the decommissioned Fengyun 1C spacecraft in low Earth orbit (LEO) by China during the testing of an anti-satellite missile system. This resulted in the exponentiation of the number of fragments in orbit, without which the increase in the historical catalogued debris count would have been only 6 % (Johnson et al., 2008).

This class of phenomena had been predicted over 30 years ago by Kessler and Cour-Palais (1978); the “Kessler Syndrome”, as it has been designated, suggests that space debris can grow irrespective of newer spacecraft launches simply due to cascading collisions between orbiting, most likely derelict, spacecraft, or fragments thereof. Such a phenomenon is capable of precipitating the arrival of a point of no return beyond which human intervention becomes futile, rendering space operations permanently unfeasible. However, the number of Earth-orbiting debris had been steadily growing even before 2007. In fact, they now outnumber active spacecraft by more than 5 to 1, inhabiting mainly the orbits commonly targeted for launches, i.e. LEO and geostationary orbits (GEOs; Andrenucci et al., 2011).

A potential chain reaction trigger is the Envisat spacecraft: a sizeable spacecraft in LEO weighing over 8000 kg, launched on 1st March 2002 and non-functional since 9th May 2012. The existence of such space objects justifies that debris mitigation strategies must be applied efficiently, whereas international rules state that at least five large space objects per year must be de-orbited in order to ensure long-term space operations (Bonnal et al., 2013).

One such mitigation strategy is termed *active debris removal* (ADR), whereby a chaser spacecraft is deployed to perform an NCRV with the target object in order to capture and de-orbit it. The e.Deorbit mission is set out to be the first ADR mission to be carried out by the ESA, demonstrating the removal of a large object



Figure 1.2: Computer-generated renders of potential active debris removal (ADR) approaches developed for the European Space Agency’s (ESA) e.Deorbit mission. (Left) Capture with net attached to tether. (Right) Capture with robotic arm.

from its current orbit and performing a controlled re-entry into the atmosphere. As one of the few ESA-owned debris in LEO, Envisat is a possible target for the mission (Biesbroek, Innocenti, et al., 2017).

ADR is thus a principal driver towards the development of affordable vision-based NCRV algorithms. e.Deorbit is part of ESA’s CleanSpace initiative, which is focused on outlining the required technology for this domain, including advanced IP for the relative navigation aspect of the rendezvous operations. A smaller scale in-orbit demonstration mission using CubeSats to test IP algorithms has been proposed: e.Inspector, as it is called, would visually inspect Envisat to determine its tumbling rate and axis. This data could then be used for validation purposes to use with e.Deorbit (Biesbroek, Wolahan, et al., 2017). Figure 1.2 illustrates two potential approaches for capturing an Envisat-type satellite with e.Deorbit.

In 2018, the first ever ADR demonstration was conducted by the RemoveDebris mission (Forshaw et al., 2016). The main satellite was launched from the ISS, and in turn deployed two 2U CubeSats to act as space debris simulators. Several novel payloads were successfully tested on the CubeSats, namely a net and harpoon capture system, as well as a drag sail to amplify the atmospheric friction, allowing for a quicker de-orbiting. A model-based pose estimation algorithm based on an enhanced, GPU-accelerated (graphics processing unit) version of GNFIR was run in-situ on images captured by the chaser’s on-board camera of one of the targets during a slow, controlled rotation lasting 210 s (Aglietti et al., 2020).

1.6 Thermal Imaging

The space environment must not be considered amiable towards vision-based navigation systems. Elementary RV or RVD/B manoeuvres are capable of triggering complex effects as seen by the on-board camera, such as shadowing or glare, by

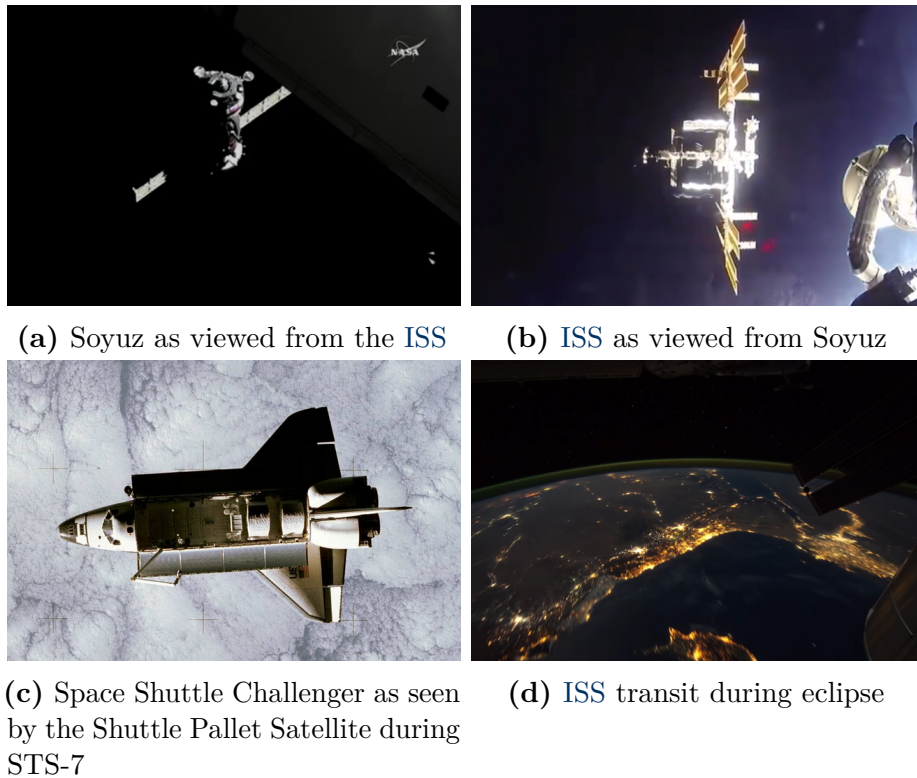


Figure 1.3: Diversity of illumination conditions during rendezvous (RV) and rendezvous and docking or berthing (RVD/B).

changing the relative configuration between the chaser, the target, and the Sun. In the case of the manoeuvre happening during an eclipse period of the orbit, direct illumination from the latter is not available at all, and the only source of natural brightness comes from Earth’s albedo. Specific materials required on artificial targets for thermal control exacerbate these effects (Meseguer et al., 2014). One such example is *multi-layer insulation (MLI)*, a “thermal blanket” composed of multiple layers highly reflecting shields with the goal of providing the spacecraft with radiative insulation. Radiators, i.e. systems that reject heat from the system to outer space, must also have a high solar reflectance to curb incoming heat. Furthermore, tumbling targets in NCRV scenarios further aggravate the problem by changing the angle of incidence of incoming light. Figure 1.3 exemplifies some of these dynamic effects as imaged by on-board cameras on the visible wavelength (0.38–0.70 μm).

An alternative approach to image an object would be to do it in terms of its emitted radiation rather than detecting what it reflects. All matter at a non-zero absolute temperature emits radiation in the wavelength range 0.1–100 μm , but the region in which that emission is concentrated moves towards longer wavelengths as the temperature decreases (Meseguer et al., 2014). For the temperature range of a spacecraft in orbit, and for many ground-based applications as well, this peak

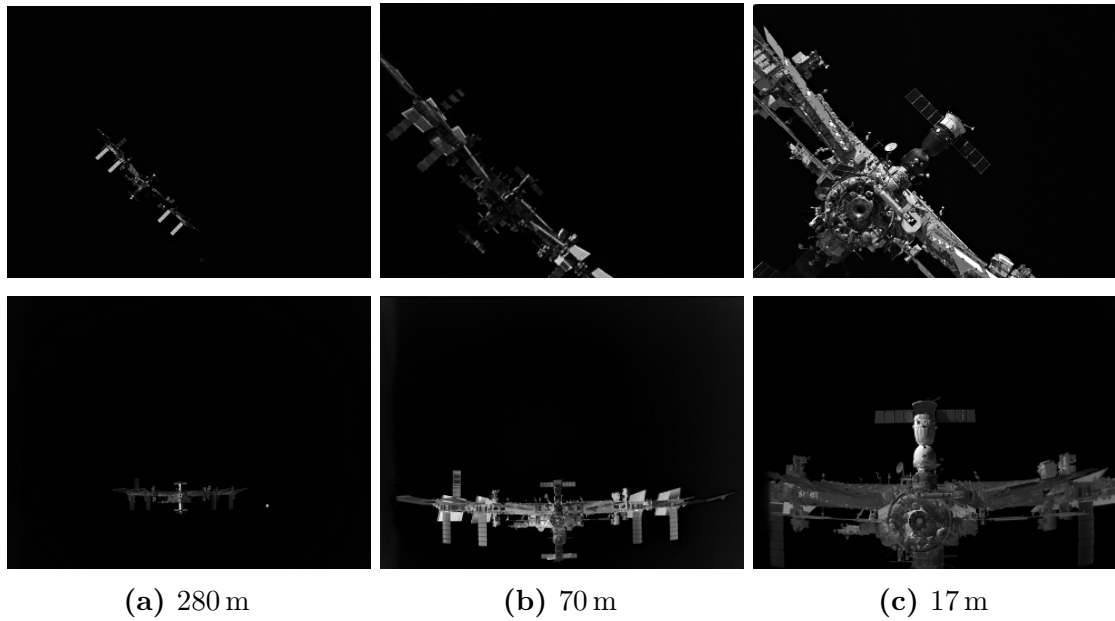


Figure 1.4: LIRIS-1 (Laser and Infra-Red Imaging Sensors) images of the International Space Station (ISS) at multiple distances (Cavrois et al., 2015). (*Top Row*) Visible modality. (*Bottom Row*) Long-wavelength infrared (LWIR) modality.

corresponds to the *long-wavelength infrared* (LWIR) band (8–14 μm). Even though thermal radiation refers to all emissions independently of the wavelength, in the LWIR region of the spectrum specialised sensors can obtain a passive image of the target bypassing the need for illumination, and for this reason it is colloquially known as “thermal infrared”. Likewise, sensors that acquire images in this modality are dubbed thermal cameras.

Thermal imagers have already been recognised as a means towards the enhancement of vision-based RV suites. Figure 1.4 showcases sample images captured by LIRIS-1 during an RVD/B sequence with the ISS in the visible and LWIR modalities, where the latter exhibits significantly more constancy relative to the former. In the image taken at 70 m distance, in particular, the centre portion of the target is mostly obscured for the visible, but distinguishable in the LWIR. While the LIRIS-1 mission objective was to collect sensor data for posterior study, thermal imaging has been used by Neptec’s Triangulation and Lidar Automated Rendezvous and Docking (TriDAR) system on an RV of the Space Shuttle with the ISS for target detection and LOS relative navigation (Ruel et al., 2012). Furthermore, the Raven experiment (Strube et al., 2015, see § 1.4) is investigating the use of its model edge-tracking algorithm for pose estimation on thermal imagery. Despite these initiatives, the use of thermal cameras navigation in space is still at a relatively underdeveloped stage and the potential of using it for 6-DOF estimation remains untrodden.

1.7 Overview of the Thesis

This thesis was the product of an industrial collaboration with [ESA](#) and the Research & Technology division of the Avionics and Power group for Thales Alenia Space France ([TAS-F](#)).

1.7.1 Objective

Given the background established by this introductory chapter, the goal of the thesis can now be defined. The main objective of this work is to design model- and vision-based solutions capable of retrieving the [6-DOF](#) pose of an object with respect to a camera in the context of space [NCRV](#) sequences, with a focus on investigating the potential of multiple imaging modalities, particularly the visible and the [LWIR](#). This camera would be mounted on the chaser vehicle, and by extension the proposed methods allow the required localisation of the target to accomplish relative navigation.

Some assumptions are made regarding the task at hand. Firstly, this thesis concentrates on the close-range [RV](#) problem, where the [6-DOF](#) pose estimation becomes relevant; in particular, the terminal subphase is tackled, with a relative distance bounded at 100 m. The target itself is considered to be known, artificial, but non-cooperative, meaning that it does not supply any equipment nor procedure to aid in the task. A special focus is placed on the tumbling case, where the target rotates uncontrollably. Furthermore, the target is assumed to be the only navigable object in the [FOV](#). Lastly, it is assumed that cameras are the only on-board sensor available to estimate the relative state. The imaging system is monocular, or follows a monocular configuration, meaning that either only one camera is used or they are configured to share the same boresight.

Throughout this dissertation, the objective is further divided into a set of research questions with the aim of answering them sequentially:

[RQ1] How do low-level image processing algorithms behave on images acquired during a space rendezvous?

Feature extraction is normally the first operation performed on an image for machine learning and pattern recognition applications. The past two decades have witnessed an upsurge in the development of innovative feature detection, description, and matching algorithms, particularly for keypoints, as the bedrock of vision-based pose estimation algorithms for both model-based and model-free approaches. However, these algorithms have been developed with ground-based applications in mind, or at most drone-based ones, where the imaging conditions differ vastly from those

experienced in orbit. As such, the performance and limitations, if any, of such IP techniques must be clearly benchmarked before incorporating a navigation solution.

[RQ2] Can a contribution be made towards model-based spacecraft relative pose estimation in the visible wavelength?

On-orbit demonstrators such as [GNFIR](#) have blazed the trail for model-based [NCRV](#) using images from cameras operating on the visible modality, albeit with limited success. Vision-based navigation algorithms used in space generally still lag behind in complexity with respect to those used on the ground. The potential for closing this gap must be investigated while bearing in mind the budget limitations of a spaceborne [GNC](#) system. Navigating complex targets under complex conditions remains a potential research avenue.

[RQ3] Can the long-wavelength infrared modality improve vision-based six degrees-of-freedom relative navigation? If so, how?

Thermal imaging has been targeted as a novel addition for new generation [GNC](#) suites, but it is currently only used as a supporting input for visible imaging or other [RV](#) sensors. Determining how [LWIR](#) features could be used explicitly to obtain a [6-DOF](#) pose solution provides an open pathway towards innovation.

1.7.2 Outline and Contributions

The structure of the thesis along with each chapter’s contribution are presented as follows:

Chapter 2 encompasses a summary of the required background concepts for the current research, including the topics of image formation, frames of reference, Lie groups, and artificial intelligence ([AI](#)) and machine learning ([ML](#)). A novel multimodal dataset of a [NCRV](#) with [Envisat](#), which will be used throughout the thesis, is introduced along with the tools utilised for its generation.

Chapter 3 uses the dataset generation tools established in the previous chapter to create a framework to benchmark keypoint detectors and descriptors for navigation. The attained results are used to provide a first-of-its-kind analysis of the performance of these algorithms on space [NCRV](#) trajectories, for both visible and [LWIR](#) modalities.

Chapter 4 employs the results obtained in the preceding chapter to derive a relative navigation solution on the visible modality using combined point and edge

features, demonstrating that 6-DOF pose estimation relative to a tumbling target is possible using a small set of reference images (i.e. keyframes).

Chapter 5 proposes a method to address the problem of pose initialisation from a single image of the target using global features. Furthermore, it improves the algorithm introduced in Chapter 4 by introducing a Kalman filter that fuses the pose estimates obtained by minimising the geometric constraints from local point and edge correspondences. Both modules are parametrised using the formalism of Lie groups which allows the automatic computation of the feature minimisation process covariances to be interpreted as the filter measurement noise. The method is validated on realistic simulations, as well as on laboratory data, and compared to competitor methods.

Chapter 6 represents this dissertation’s contribution in the field of deep learning for NCRV, in which a CNN architecture is reinforced with a recurrent neural network (RNN) back-end to yield the first full deep recurrent convolutional neural network (DRCNN) for spacecraft pose estimation. The resulting framework is used on synthetic trajectories to compare the performance of the visible-only modality against visible-thermal multimodal inputs. The capability of the network to generalise to unseen eclipse trajectories is analysed. Moreover, the performance of the network trained on limited data is assessed on laboratory data.

Chapter 7 recapitulates the work performed with a conclusion and extends recommendations towards future work.

1.7.3 Experimental Setup

Experimental validations of the solutions proposed in this thesis have been conducted on real data custom-acquired for this purpose. Two different experimental setups were created, in which NCRV sequences were simulated in the laboratory with distinct tumbling targets.

1.7.3.1 Unmanned Autonomous Systems Laboratory

The Unmanned Autonomous Systems Laboratory (UASL) is Cranfield University’s facility for testing autonomous ground and air vehicles. The lab features a 1:17 scaled mock-up of Envisat capable of one degree-of-freedom (DOF) rotation about the horizontal axis at a constant rate of 5.73 deg s^{-1} . The room has been fitted with blackout curtains to simulate a deep space background, and a custom-built

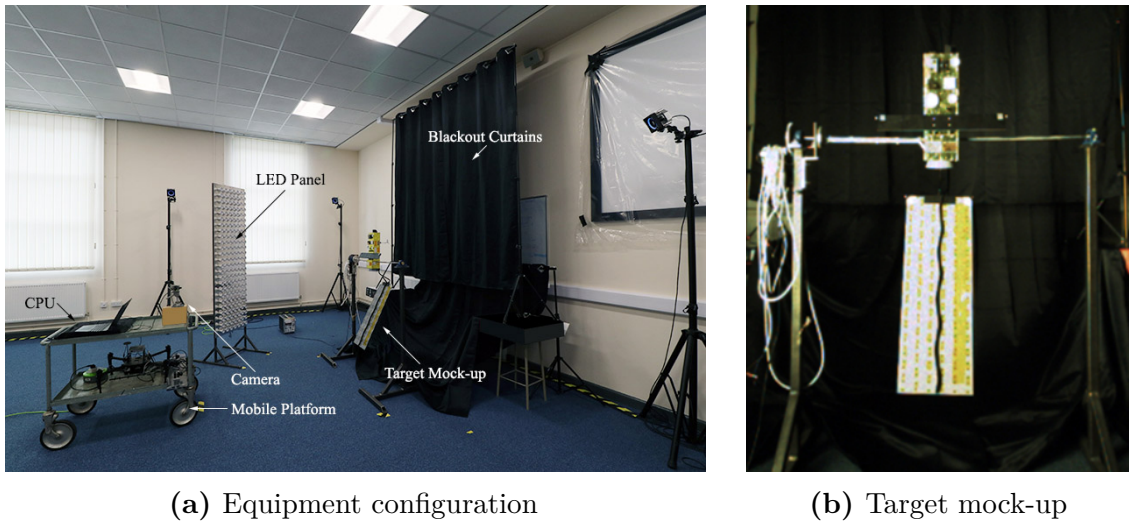


Figure 1.5: Validation setup of the Unmanned Autonomous Systems Laboratory (UASL) at Cranfield University.

light-emitting diode (LED) illumination panel running at 900 W is used to simulate direct sunlight acting on the target.

A visible camera is mounted on a platform which remains static for the data acquisition. The camera is a Matrix Vision mvBlueFOX MLC200wC colour camera with a 1/3 in format complementary metal-oxide semiconductor (CMOS) sensor (Onsemi MT9V034) and a native resolution of 752 px \times 480 px, fitted with a 2.97 mm focal length Matrix Vision IRCB5M29740N lens.

The ground truth is obtained via manual registration of the first frame and propagation of the state according to the target’s rotational mode. Since only short data acquisition sequences are considered (maximum of three revolutions), the target mechanism is not active long enough for any measurable drift to occur. Figure 1.5 illustrates the experimental setup. Table 1.3 summarises the camera’s physical characteristics.

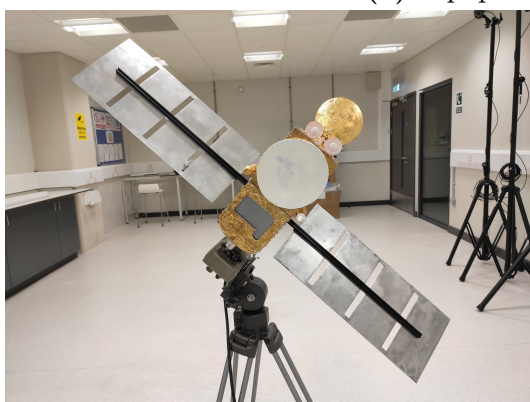
1.7.3.2 Autonomous Systems Laboratory

The Autonomous Systems Laboratory (ASL) is City, University of London’s facility dedicated to the research activities of the Robotics, Autonomy and Machine Intelligence group. A volume of approximately 5 m \times 5 m \times 3 m is available for the testing of autonomous vehicle trajectories with ground truth provided by an OptiTrack system.

OptiTrack is a motion capture system that can record 6-DOF pose data of rigid and flexible bodies by detecting, tracking, and triangulating passive near infrared markers placed on targets. The data can be saved or stream over a local network



(a) Equipment configuration



(b) Target mock-up



(c) Camera configuration

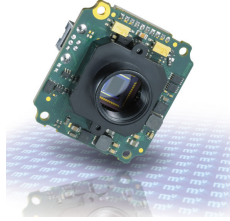
Figure 1.6: Validation setup of the Autonomous Systems Laboratory (ASL) at City, University of London.

in real-time. The OptiTrack setup at City consists in six PrimeX 13 cameras with a resolution of $1280 \text{ px} \times 1024 \text{ px}$ running at a native framerate of 240 Hz, capable of achieving positional errors less than $\pm 0.20 \text{ mm}$ and rotational errors less than 0.5 deg .

A 1:4 scale mock-up of the National Aeronautics and Space Administration (NASA; United States) and National Centre for Space Studies (CNES; France) satellite Jason-1 is considered for data acquisition. The mock-up rotates along its vertical axis at a constant rate of 6 deg s^{-1} . It is placed inside the capture volume with blackout curtains behind it to simulate a deep space background. Illumination is guaranteed by a 400 W directional floodlight. A visible camera and a thermal camera are used for multimodal data acquisition:

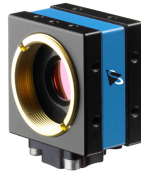
- The Imaging Source DFK 22BUC03 colour camera with a 1/3 in format CMOS sensor (Onsemi MT9V024) and a native resolution of $744 \text{ px} \times 480 \text{ px}$, fitted with a Kowa LM4NCL 3.5 mm focal length lens;

Table 1.3: Technical data – mvBlueFOX MLC200wC



Parameter	Dimensions	Value
Resolution	px	752×480
Frame rate	Hz	90
Focal length	mm	2.97
Horizontal FOV	deg	74.4
Vertical FOV	deg	51.7

Table 1.4: Technical data – DFK 22BUC03



Parameter	Dimensions	Value
Resolution	px	744×480
Frame rate	Hz	76
Focal length	mm	3.5
Horizontal FOV	deg	65.6
Vertical FOV	deg	44.7

- FLIR Vue Pro R uncooled VOx microbolometer, fixed lens 9 mm focal length, $640 \text{ px} \times 512 \text{ px}$ native resolution.

Due to the lack of a beamsplitter setup, the cameras are mounted side to side with a very small baseline to minimise image disparity. The rig is placed on a mobile platform that navigates the capture volume during data acquisition. The setup is illustrated in Figure 1.6. Tables 1.4 and 1.5 summarise the physical characteristics of the cameras.

Table 1.5: Technical data – FLIR Vue Pro R

Parameter	Dimensions	Value
Resolution	px	640×512
Frame rate	Hz	30
Focal length	mm	9
Horizontal FOV	deg	69
Vertical FOV	deg	56
Spectral band	μm	7.5–13.5

1.7.4 Published and Submitted Manuscripts

Conferences

- [C1] D. Rondao and N. Aouf (Jan. 2018). “Multi-View Monocular Pose Estimation for Spacecraft Relative Navigation”. In: *2018 AIAA Guidance, Navigation, and Control Conference*. Kissimmee, FL: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2018-2100
- [C2] D. Rondao, N. Aouf, and O. Dubois-Matra (Oct. 2018). “Multispectral Image Processing for Navigation Using Low Performance Computing”. In: *69th International Astronautical Congress (IAC) 2018*. Bremen, Germany: IAF. URL: <https://dspace.lib.cranfield.ac.uk/handle/1826/13558>
- [C3] M. Hogan et al. (June 2021). “Using Convolutional Neural Networks for Relative Pose Estimation of a Non-Cooperative Spacecraft with Thermal Infrared Imagery”. In: *11th International ESA Conference on Guidance, Navigation and Control Systems*. Accepted manuscript. Virtual conference: ESA

Journals

- [J1] D. Rondao, N. Aouf, M. A. Richardson, and O. Dubois-Matra (July 2020). “Benchmarking of local feature detectors and descriptors for multispectral relative navigation in space”. In: *Acta Astronautica* 172, pp. 100–122. DOI: 10.1016/j.actaastro.2020.03.049
- [J2] D. Rondao, N. Aouf, M. A. Richardson, and V. Dubanchet (2021). “Robust On-Manifold Optimization for Uncooperative Space Relative Navigation with a Single Camera”. In: *Journal of Guidance, Control, and Dynamics*. Article in advance, pp. 1–26. DOI: 10.2514/1.G004794
- [J3] D. Rondao, N. Aouf, and M. A. Richardson (2021). “ChiNet: Deep Recurrent Convolutional Learning for Multimodal Spacecraft Pose Estimation”. In: *IEEE Transactions on Aerospace and Electronic Systems*. Manuscript in submission

1.7.5 Awards and Competitions

- [A1] ESA student sponsorship for the attendance of the 69th International Astronautical Congress in Bremen, Germany, 2018.
- [A2] Finalist (placed as one of four shortlisted candidates) of the Eric Beverley Bursary competition organised by the Worshipful Company of Coachmakers and Coach Harness Makers, 2018.

- [A3] Finalist (placed as one of three shortlisted candidates) of the 69th International Astronautical Congress Interactive Presentation competition for Category C – Technology, 2018.
- [A4] European Union Erasmus+ grant to support industrial placement at Thales Alenia Space, Cannes, France, 2019.

CHAPTER 2

Theoretical Background and Tools

In this chapter, the fundamental concepts that serve as the basis for the contributions made throughout this thesis are introduced. It begins with the field of computer vision by reviewing how an image is formed in a camera, in the visible and thermal infrared modalities, and both in terms of the underlying physical processes and adopted mathematical models. Next, the frames of reference encountered in a typical space rendezvous scenario are described, followed by a characterisation of the six degree-of-freedom pose space in the context of Lie groups. An in-depth delineation of artificial intelligence is presented, covering classical machine learning techniques and the recently-popularised field of deep learning, as well as their role in the estimation of the relative pose from captured images. Lastly, the tools used in the development of a synthetic rendezvous dataset are covered.

2.1 On Image Formation

A camera is a device that captures a (usually) dynamic scene onto a static frame. Typically, when considering a camera, one intuitively draws an equivalence to a light-sensing mechanism, since images replicating the output of the human eye, i.e. photographs, are commonplace in today's society and cover a vast range of applications. However, imaging is certainly not limited to this specific portion of the electromagnetic spectrum as different methods exist to capture a scene beyond the visible. This section briefly describes image formation mechanisms for cameras in the visible wavelength and in the long-wavelength infrared (LWIR).

2.1.1 Visible Wavelength

The term “camera” is nowadays actually a synecdoche for *solid-state* camera: one that comprises arrays of photosensitive elements mounted on integrated circuits, in opposition to the preceding standard of vacuum-tube cameras which supplied an analogue voltage proportional to the intensity of incident light on a photoconductive electrode. Solid-state cameras brought about multiple improvements, such as a smaller form factor, additional robustness, and unlikelihood of damage from high illumination intensities; they are broadly available as a commercial-off-the-shelf (COTS) product at relatively inexpensive cost (Painter et al., 1994).

Such cameras include area image sensors that are engineered to function in a two-step manner: conversion of incoming photons to electric charge at each pixel; and charge transfer to an output amplifier following by conversion to an electric signal (Tredwell, 1995). The former is performed by a silicon semiconductor image-sensing element, which can be a photodiode, photocapacitor, or photoconductor, absorbing the photon and resulting in the generation of an electron-hole pair. The latter is achieved with a readout element, of which there are two main types in use: the charge-coupled device (CCD) and the complementary metal-oxide semiconductor (CMOS). The CCD operates by transferring charge packets from the vicinity of the image-sensing element’s surface through physical storage areas towards the output to be converted into a voltage. Anti-blooming circuits are incorporated to attenuate the phenomenon of excess charge bleeding into the readout element or the substrate and corrupting the image. CMOS sensors, on the other hand, integrate one or more transistors into each pixel, allowing them to be individually read and amplified, and to perform on-chip image processing. This makes it more compact than a CCD sensor, which requires clocks and signal processing in separate hardware (Holst, 1995). Despite the prevalence of CCDs for traditional applications due to their quality, CMOS are the default choice nowadays for digital cameras (Szeliski, 2011).

2.1.2 Infrared Wavelengths

Despite imaging a scene’s emitted thermal radiation, infrared (IR) detector arrays have a parallel structure with respect to their visible wavelength counterparts (Kozlowski and Kosonocky, 1995). The front-end of an IR detector is the focal plane array (FPA), which can be either scanning or staring-based.¹ Scanning FPAs are composed of linear one-dimensional arrays that scan the scene across the horizontal field of view (FOV) by means of a rotating mirror to build a two-dimensional image

¹Some sources consider an FPA to be exclusively synonymous with staring arrays.

over time. Staring **FPA**s, on the other hand, encompass dedicated pixels at each resolution element and are analogous to **CCD**s or **CMOS** in the visible wavelength.

Infrared detection is achieved on **FPA**s with either photon or thermal detectors which convert the incoming photons into electrical signals. The former can be photovoltaic or photoconductive elements and need to be actively cooled to achieve moderate performance; within these, intrinsic detectors operate at higher temperatures while dissipating less power when compared to extrinsic ones. Intrinsic detectors are typically mercury cadmium telluride-based (**HgCdTe**) or indium antimony-based (**InSb**), whereas the most popular extrinsic photoconductive material is doped silicon (**Si**). Thermal detectors, or bolometers, process incident radiation by absorbing the energy and registering the consequent change in the temperature of the system, and are typically uncooled. Resistive bolometers are composed of a thin sheet of resistive material placed over a silicon readout, where incident **IR** radiation changes the resistance of the detector element proportionally to the change in local temperature. Capacitive bolometers are most commonly based on the pyroelectric effect and thus use materials such as lithium tantalate (**LiTaO**) and barium strontium titanate (**BaSrTiO**).

The above-mentioned technology is available for short-wavelength infrared (**SWIR**; 1–3 μm), medium-wavelength infrared (**MWIR**; 3–5 μm), and **LWIR** (8–14 μm). For Earth-based **IR** imaging applications, the **MWIR** and **LWIR** bands are the most used since atmospheric transmission is maximised therein; **SWIR** has limited use but has been applied to astronomical settings, such as the Hubble Space Telescope (**HST**). Whereas **MWIR** cameras provide a scene image with higher contrast, **LWIR** cameras are favoured by higher resistance to atmospheric turbulence and reduced **IR** emission rates originating from colder backgrounds; the latter is especially important in military applications against high-temperature countermeasures such as flares, and in spaceborne scenarios against solar glint reflected off insulating satellite materials. For Earth-based applications, as the wavelength is increased in the electromagnetic scale (up to a few microns), the influence of reflected solar radiation diminishes. This is because the background radiation increases, thus decreasing the contrast, e.g. in the visible wavelength, daytime contrast is higher than in **LWIR** as for the former the source is light reflected off objects at ambient temperature generally under 290 K and for the latter the background flux is equivalent to sunlight. On the other hand, at nighttime, natural visible light is non-existent which brings the level of contrast to zero, but in **LWIR** target vs. background metrics are comparable to daytime. In space, **LWIR** images will generally witness a higher contrast variability, though, since the scene is mainly comprised of the deep space background measuring a baseline

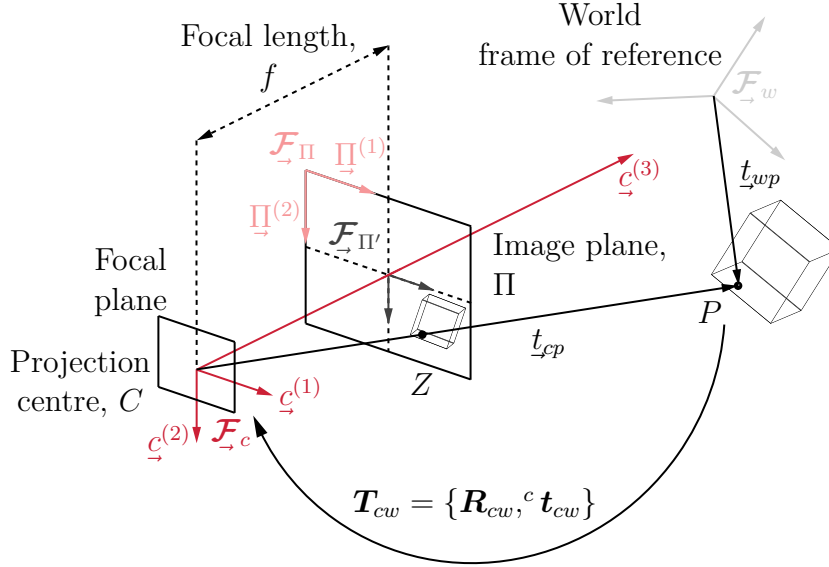


Figure 2.1: Pinhole camera model geometry. The 3D point P is projected onto the image plane Π as Z , i.e. the intersection of \overline{CP} with Π .

temperature of 2.7 K.

2.1.3 Pinhole Camera Model

Regardless of the used device or spectral band considered, all cameras must be able to cluster the incident light rays on the sensor in order to generate a good enough signal to warrant image formation. This is achieved using a *lens*. For most computer vision applications, complex optical models are usually avoidable and it is sufficient to model the lens as an ideal pinhole at a certain distance, f , from the focal plane. It is consequently known as the *pinhole camera model* (Szeliski, 2011), after the “camera obscura” phenomenon named by Johannes Kepler in 1604 — but which had been known since antiquity — in which a small hole on the wall of a darkened room allows rays travelling from different points on the outside in, forming an image on the wall opposite the hole (Dupré, 2008).

The pinhole camera model is described by its optical centre, or camera projection centre, C , and the *image plane*, Π (see Figure 2.1). The perpendicular distance from C to Π is termed the *focal length*, denoted by f ; the line intersecting C that is perpendicular to Π is the optical, or principal, axis of the camera; and the plane parallel to the image plane containing C is the focal, or principal, plane. The centre of projection is the origin of the three-dimensional Cartesian coordinate system, the *camera reference frame*, represented by the vectrix (Barfoot, 2017) $\mathcal{F}_c = [\underline{c}^{(1)} \ \underline{c}^{(2)} \ \underline{c}^{(3)}]^\top$, of which $\underline{c}^{(3)}$ is the optical axis. The relationship between the coordinate of a 3D point expressed in \mathcal{F}_c and the coordinates of its projection in the

image plane is characterised by a *perspective projection*.

Let P denote such a 3D point in the scene. The vector indicating the position of P relative to the origin of $\underline{\mathcal{F}}_c$ is denoted as \underline{t}_{cp} . Its representation in coordinates of $\underline{\mathcal{F}}_c$ is the 3×1 column vector ${}^c\mathbf{t}_{cp}$. With some loss of generality but no detriment to the theory herein established, the term ‘‘point’’ shall also be used to refer directly to the column vector representation ${}^x\mathbf{t}_{xp}$ of the position of any point P relative to the origin of some reference frame $\underline{\mathcal{F}}_x$ in coordinates of $\underline{\mathcal{F}}_x$.

Furthermore, let $\mathbf{p}' = [p'_1 \ p'_2 \ p'_3]^\top := {}^c\mathbf{t}_{cp}$ for brevity. Using perspective projection, points are projected onto the image plane by dividing them by the p'_3 component:

$$p'_3 \tilde{\mathbf{z}}' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{p}}', \quad (2.1)$$

where $\mathbf{z}' = [z'_1 \ z'_2]^\top := {}^\Pi\mathbf{t}_{\Pi'z}$ is the position of the projected point Z relative to the centre of Π in image plane coordinates, and $\tilde{\mathbf{x}} := [\mathbf{x} \ 1]^\top$ is the conversion of a vector \mathbf{x} to *homogeneous coordinates*.

The transformation illustrated in Equation (2.1), however, does not take into account certain nuances of image formation, in particular the pixel spacing in the camera sensor. In effect, for computer vision applications, the coordinates of points in the image plane must be mapped to the frame buffer, $\underline{\mathcal{F}}_\Pi$. This can be achieved if the intrinsic camera parameters are known, forming the *intrinsic camera*, or calibration, *matrix*:

$$\mathbf{K} = \begin{bmatrix} f/s_1 & s & c_1 \\ 0 & f/s_2 & c_2 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

where s_1, s_2 are the spacing of the sensor pixels in the horizontal and vertical directions, respectively, s is the skew between the sensor axes and the optical axis, and c_1, c_2 are the coordinates in $\underline{\mathcal{F}}_\Pi$ of the intersection of Π with $\underline{c}^{(3)}$. A 3D point can then be mapped directly to frame buffer coordinates as

$$\lambda \tilde{\mathbf{z}} = \mathbf{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{p}}', \quad (2.3)$$

where $\mathbf{z} := {}^\Pi\mathbf{t}_{\Pi z}$ and λ is the resulting scale factor. The intrinsic camera matrix is obtained through a *camera calibration* process preceding the mission. However,

Equation (2.3) is only valid for linear projection (i.e. straight lines in the scene generate straight lines in the image), whereas most lenses have noticeable radial distortion (i.e. curvature). It is therefore habitual to estimate a vector of radial distortion parameters alongside \mathbf{K} during camera calibration, then undistort the image and work with a post-undistortion \mathbf{K} . Throughout this thesis, linear projection is assumed valid and \mathbf{K} is assumed to be known; additional information on camera calibration and radial distortion models can be found in Szeliski (2011, § 2.1.6, § 6.3).

Three-dimensional scene points, though, are not defined a priori in the camera reference frame, \mathcal{F}_c , but rather in terms of coordinates of the *world frame*, \mathcal{F}_w . The two reference frames are related by a rotation, which can be represented by a 3×3 *rotation matrix* \mathbf{R}_{cw} , and by a 3×1 *translation vector* ${}^c\mathbf{t}_{cw}$. Regarding the notation, the former should be read as “the rotation mapping a vector expressed in \mathcal{F}_w coordinates to one expressed in \mathcal{F}_c coordinates”, whereas the latter is “the vector translating the origin of \mathcal{F}_c to the origin of \mathcal{F}_w , expressed in coordinates of \mathcal{F}_c ”. In the context of estimating the pose of a rigid body, these quantities are also typically called the *attitude* and *position*, respectively. Let $\mathbf{p} := {}^w\mathbf{t}_{wp}$ represent a 3D point P expressed in \mathcal{F}_w coordinates. Then, one has:

$$\begin{aligned}\tilde{\mathbf{p}}' &= \begin{bmatrix} \mathbf{R}_{cw} & {}^c\mathbf{t}_{cw} \\ \mathbf{0}^\top & 1 \end{bmatrix} \tilde{\mathbf{p}} \\ &= \mathbf{T}_{cw} \tilde{\mathbf{p}}.\end{aligned}\tag{2.4}$$

The matrix \mathbf{T}_{cw} is the relative *pose*, having six degrees-of-freedom (DOF) equally distributed by \mathbf{R} and \mathbf{t} , where the subscripts have been dropped for brevity. It is also termed the *extrinsic camera matrix* (in contrast to \mathbf{K}). Intrinsic and extrinsic parameters are often combined in a single 3×4 *projection matrix*:

$$\mathbf{P} = \mathbf{K}\mathbf{T}_{1:3,1:4}.\tag{2.5}$$

The projection of a 3D point in the world reference frame to a pixel coordinate in the frame buffer is therefore given by

$$\lambda \tilde{\mathbf{z}} = \mathbf{P} \tilde{\mathbf{p}}.\tag{2.6}$$

It is common to rewrite Equation (2.6) by defining a projective function $\pi(\tilde{\mathbf{z}}) := \tilde{\mathbf{z}}_{1:2}/\tilde{\mathbf{z}}_3$ that applies the mapping from the 2D projective space \mathbb{P}^2 to \mathbb{R}^2 to a point expressed in homogeneous coordinates, thus yielding the *reprojection equation*:

$$\mathbf{z} = \pi(\mathbf{K}\mathbf{T} \oplus \mathbf{p}).\tag{2.7}$$

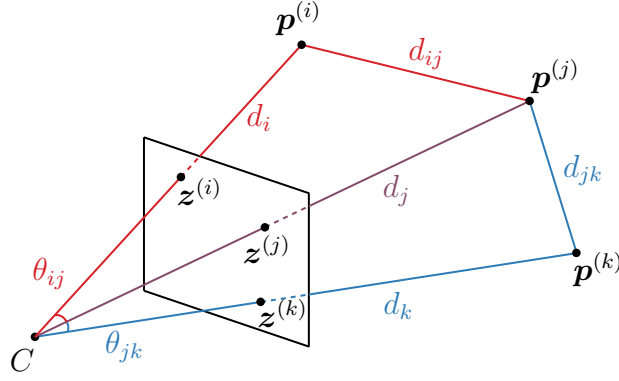


Figure 2.2: The perspective- n -point (PnP) problem formulation. 3D points in the scene P_i , known in \mathcal{F}_w coordinates as $\mathbf{p}^{(i)}$, and their corresponding projection Z_i , known in \mathcal{F}_Π as $z^{(i)}$, define a ray passing through the optical centre C . From at least three known correspondences, the distances d_i to C can be estimated, which is equivalent to estimating their representation in \mathcal{F}_c coordinates. The recovery of the pose \mathbf{T} then comes from aligning the two point clouds.

The \oplus operator denotes pose-point composition and it is used to avoid limiting the reprojection equation and similar formulations to the matrix representation of the pose (i.e. writing $\mathbf{T} \oplus \mathbf{p}$ implicitly assumes the form of Eq. [2.4] and the necessary division by the homogeneous scale factor). It shall also be used for pose-pose composition (see § 2.3.1).

Remark 2.1: On the Projective Transform

Any two homogeneous points $\{\tilde{z}^{(i)}, \tilde{z}^{(j)}\} \in \mathbb{P}^2$ that differ only by scale are equivalent, i.e. $\tilde{z}^{(i)} = \lambda \tilde{z}^{(i)} = \tilde{z}^{(j)}$ is true for any $\lambda \in \mathbb{R} \setminus \{0\}$. This effectively means that a point in the 2D image plane is treated as a ray in projective space. When applying the transformation $\pi(\tilde{z}) \mapsto z$, one has $z \in \mathbb{R}^2$, meaning that the depth of a 3D point is lost after reprojection.

Indeed, inverting Equation (2.6) yields:

$$\mathbf{p} = -\mathbf{R}^{-1}\mathbf{t} + \lambda\mathbf{R}^{-1}\mathbf{K}^{-1}\tilde{z}, \quad \lambda \in \mathbb{R}, \quad (2.8)$$

which shows that the resulting \mathbf{p} is not a 3D point, but an optical ray that passes through it since λ is unknown.

2.1.4 Perspective- n -Point Problem

Given a set of $n \geq 3$ correspondences $\{\mathbb{Z}, \mathbb{P}_w\}$, where $\mathbb{Z} = \{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(n)}\}$ is the set of projected points in \mathcal{F}_Π and $\mathbb{P}_w = \{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n)}\}$ is the set of scene points in \mathcal{F}_w , the pose \mathbf{T} that best aligns them can be resolved. If the intrinsic matrix \mathbf{K} is known, the question consists in solving the *perspective- n -point* (P n P; Szeliski, 2011) problem. Consider Figure 2.2, which represents the line segments connecting the camera’s optical centre, C , to each of the world points in \mathbb{P}_w , passing through each image point in \mathbb{Z} . Taking each world point pair yields a triangle $\triangle P_i C P_j$ to which the cosine law can be applied:

$$d_i^2 + d_j^2 - 2d_i d_j \cos \theta_{ij} - d_{ij}^2 = 0, \quad (2.9)$$

where d_i is the length of $\overline{CP_i}$ and θ_{ij} is the angle of $\triangle P_i C P_j$ corresponding to C . With three such equations, the scenario is reduced to the perspective-3-point (P3P) problem, which solves for the three distances $\{d_i, d_j, d_k\}$. The result is equivalent to the determination of the scene points in camera frame coordinates, and the remainder of the problem consists then in estimating the alignment between the two point cloud sets $\{\mathbb{P}_c, \mathbb{P}_w\}$: the centroids of each set are used to solve for the translation, whereas the rotation is determined by solving the constrained Procrustes problem, for which the singular value decomposition (SVD) is used (Schönemann, 1966). The latter is equivalent to Wahba’s (1965) problem, which was originally posed in the context of spacecraft attitude determination.

P3P yields four real solutions for \mathbf{T} , so in practice a fourth point is often included to disambiguate. The P n P problem has a closed-form solution for $n = \{3, 4, 5\}$ points, while for $n < 3$ it is not well-defined, and for $n > 5$ it can be solved iteratively (Tang et al., 2008). A P n P solver is the staple of any computer vision programming library, and over the penultimate decade algorithmic advances have brought on powerful solutions with $O(n)$ time complexity capable of dealing with many more than five points, such as EP n P, which is based on a closed-form solution for four control points, followed by an efficient Gauss-Newton optimisation step (Lepetit, Moreno-Noguer, et al., 2008), or UP n P, which discards the calibrated camera assumption (Moreno-Noguer et al., 2007).

2.1.5 Additional Considerations

The theory presented above is sufficient to model the relation between a three-dimensional scene and a two-dimensional image of it. However, it is important to highlight a few additional parameters that will affect image formation in a digital

camera (Szeliski, 2011). After concentrating the incoming rays in the lens, the speed of the shutter will dictate the amount of light that hits the sensor, i.e the *exposure time*; an exposure too low may cause the image to be undesirably dark, whereas one too high may lead to overexposure. Directly related to the intrinsic parameters s_1 and s_2 in Equation (2.2), the *sensor pitch* also affects the light sensitivity, since it represents the physical space between two adjacent sensor cells (smaller pitch leads to smaller area which leads to reduced photon exposure, but also to a larger resolution). The values c_1 and c_2 are dependent on the *sensor size*, where a larger one means added photo-sensitivity. The sensor signal amplification gain is controlled by the *ISO*,² which is defined in standardised units of $\{100, 200, 400, \dots\}$. A higher ISO enhances an image acquired under low lighting conditions, which is especially useful when the exposure cannot be further increased, but it also amplifies the *sensor noise*. Other noise sources apart from the amount of incoming light and sensor gain include fixed pattern noise (minute differences in the individual sensitivity of each sensor element), shot noise (or photon noise, which depends on the particle nature of light), and quantisation noise (arising from the analog-to-digital conversion bit resolution). Image noise is arguably the parameter that most affects image processing (IP) tasks, such as feature detection, feature matching, and, of course, denoising, and it is thus commonplace to estimate the noise level; this is typically achieved with resort to Gaussian models or, more rarely, Poisson noise models.

2.2 On Frames of Reference

This section defines the frames of reference necessary to describe the motion of spacecraft in the context of a space rendezvous. A frame is defined by its origin (a 3D point in space, A) and by a set of three orthogonal unit vectors ($\underline{a}^{(1)}, \underline{a}^{(2)}, \underline{a}^{(3)}$). In this thesis, Barfoot’s (2017) more compact vectrix representation $\underline{\mathcal{F}}_a = [\underline{a}^{(1)} \ \underline{a}^{(2)} \ \underline{a}^{(3)}]^\top$ is commonly adopted. Furthermore, other terms such as “reference frame”, “coordinate frame”, or simply “frame”, are treated as synonyms.

2.2.1 Spacecraft Body Frame

The body frame has its origin typically fixed at the center of mass of the spacecraft and the axes rotate with it. The exact configuration depends on the spacecraft manufacturer or on the flight dynamics engineers working on the mission. The body frames are of critical importance in a close-range rendezvous. Figure 2.3 illustrates

²Not an acronym itself, but an eponym in reference to the International Organisation for Standardisation, which defined levels of brightness analogous to film that were adopted by digital camera manufacturers.

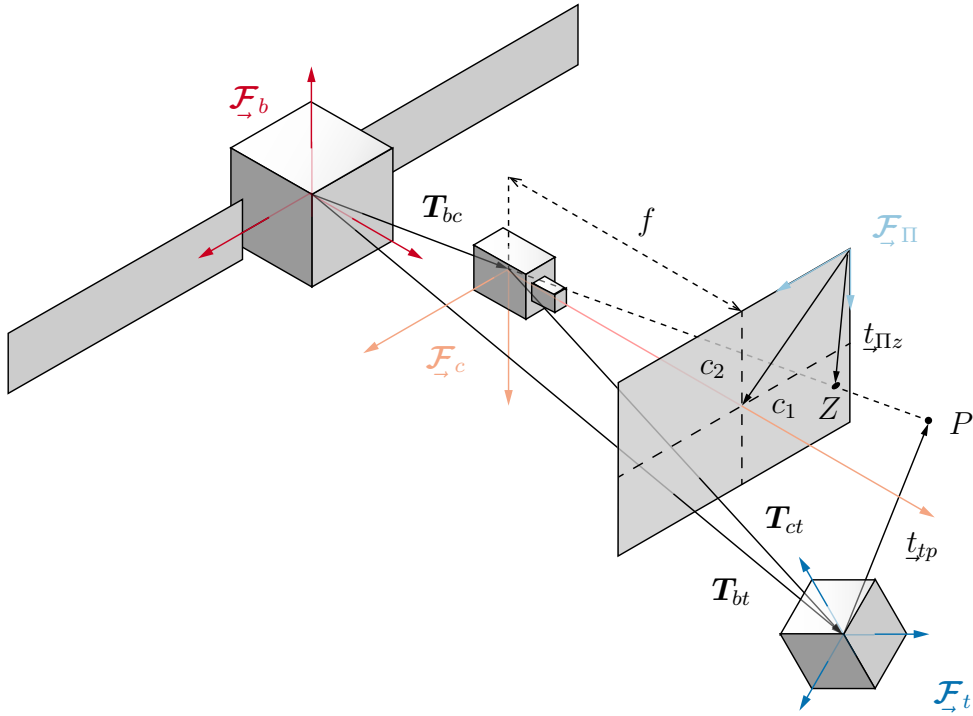


Figure 2.3: Geometric relationships between the chaser body frame \mathcal{F}_b , the camera frame \mathcal{F}_c , the target body frame \mathcal{F}_t in the context of landmark imaging in \mathcal{F}_{II} (cf. Fig. 2.1).

the geometric configuration between a chaser spacecraft body frame, \mathcal{F}_b , and a target object body frame, \mathcal{F}_t , in the context of an on-board imager adopting the pinhole camera model. The spacecraft relative pose estimation problem can be distilled in terms of finding the rigid body transformation represented by the matrix \mathbf{T}_{tb} . The computer vision system, however, operates in terms of the camera frame, \mathcal{F}_c . In practice, the relative pose between the camera frame and the chaser body frame, \mathbf{T}_{bc} is fixed and can be determined in pre-mission calibration. Therefore, for the scope of this thesis, it is assumed known and equal to $\mathbf{T}_{bc} = \mathbf{I}$, i.e. the chaser frame coincides with the camera frame. Similarly, as the purpose of the work is relative pose estimation in the context of a spacecraft rendezvous, the world frame \mathcal{F}_w (§ 2.1.3) is often taken to be coincident with \mathcal{F}_t unless otherwise stated.

Note that the centre of mass of the spacecraft, and hence the origin of \mathcal{F}_b , may shift during the mission due to changes in propellant levels (Fehse, 2003). This shift is further assumed to be outside of the scope of this thesis and therefore \mathcal{F}_b is considered fixed to the spacecraft geometry.

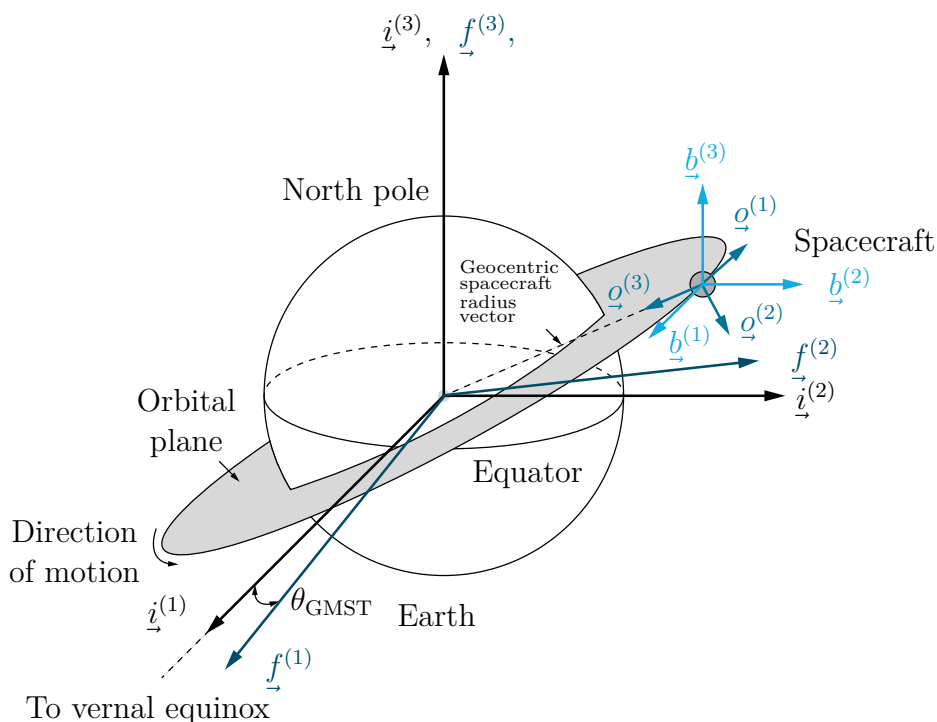


Figure 2.4: Definition of various orbital reference frames (adapted from Ronda, 2016).

2.2.2 Earth-Centred Inertial Frame

The Earth-centred inertial (**ECI**) frame of reference (denoted by \mathcal{F}_i , see Figure 2.4) is one defined to establish the spacecraft orbital equations of motion in a non-accelerated or non-rotating frame (Markley and Crassidis, 2014). The $\underline{i}^{(1)}$ axis is aligned with the vernal equinox direction, i.e. the intersection of Earth’s equatorial plane with the plane of its orbit around the Sun, in the direction of the Sun’s position relative to the Earth on the first day of spring. The $\underline{i}^{(3)}$ axis is aligned with Earth’s north pole, whereas the $\underline{i}^{(2)}$ completes the right-handed triad. Since neither the polar axis nor the vernal equinox direction are inertially fixed, the **ECI** axes are defined to be mean orientations at a fixed epoch time (in particular, the standard epoch is J2000).

2.2.3 Earth-Centred/Earth Fixed Frame

For completeness, the Earth-centred/Earth-fixed (**ECEF**) frame of reference (denoted by \mathcal{F}_f) is described herein. Its origin is also fixed at the centre of Earth but, unlike the **ECI** frame, it rotates with the planet: the $\underline{f}^{(3)}$ axis is aligned with the north pole, $\underline{f}^{(1)}$ points in the direction of Earth’s prime meridian, and $\underline{i}^{(2)}$ completes the right-handed triad. The angular difference between \mathcal{F}_f and \mathcal{F}_i is the angle θ_{GMST} (see Figure 2.4), i.e. the Greenwich mean sidereal time (**GMST**; Markley and Crassidis, 2014). θ_{GMST} is defined in terms of the number of Julian centuries elapsed from the

J2000 reference epoch, and the **ECEF** frame is useful to define physical properties that affect the orbit of a satellite, such as the nonspherical mass distribution of Earth, an effect often condensed into the term “ J_2 ”: a reference to the largest-weighting term of the spherical harmonic expansion used to correct the planet’s gravitational potential function; see (Rondao, 2016, Chap. 3) for additional details.

2.2.4 Local-Vertical/Local-Horizontal Frame

The local-vertical-local-horizontal (**LVLH**) frame of reference (denoted by \mathcal{F}_o), also occasionally termed local orbital frame, is used to describe the orbital motion of a body. The origin coincides with the centre of mass, the $\underline{o}^{(3)}$ axis points along the nadir vector towards the centre of the Earth, the $\underline{o}^{(2)}$ axis is aligned with the negative orbit normal, and the $\underline{o}^{(1)}$ axis completes the right-handed system. The set $\{\underline{o}^{(1)}, -\underline{o}^{(2)}, \underline{o}^{(3)}\}$ is often referred to as the V-bar, H-bar, and R-bar approach vectors, respectively.³

2.3 On Lie Groups

It is often useful to perform operations on poses, such as computing a combination, a difference, or a representation change. Unlike a translation vector, though, a rigid body transformation does not represent a vector space, meaning that the sum of two poses is not a valid pose. Indeed, the matrix \mathbf{T} is the homogeneous representation of an element of the 3-dimensional *special Euclidean group* (R. M. Murray et al., 1994):

$$\text{SE}(3) := \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mid \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4 \times 4}. \quad (2.10)$$

Analogously, the rotation matrix \mathbf{R} is the homogeneous representation of an element of the 3-dimensional *special orthogonal group*, $\text{SO}(3)$. $\text{SE}(3)$ is a 6-dimensional smooth manifold with matrix multiplication as the group operation, meaning that for $\{\mathbf{T}^{(1)}, \mathbf{T}^{(2)}\} \in \text{SE}(3)$, one has:

$$\mathbf{T}^{(1)}\mathbf{T}^{(2)} = \mathbf{T}^{(3)} \in \text{SE}(3). \quad (2.11)$$

Furthermore, it is a non-abelian Lie group, meaning that chaining poses is possible but order-dependant:

$$\mathbf{T}_{ca} = \mathbf{T}_{cb}\mathbf{T}_{ba}. \quad (2.12)$$

³Named after the variable symbols $\{\mathbf{v}, \mathbf{h}, \mathbf{r}\}$ commonly used to describe an orbit’s radius, angular momentum, and velocity vectors, with which they are aligned.

Remark 2.2: On Key Concepts

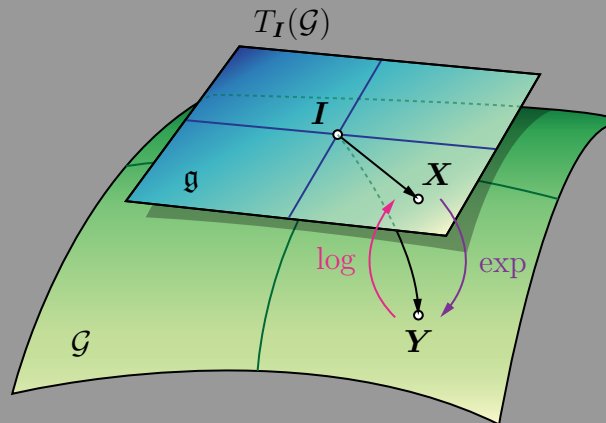


Figure 2.5: A matrix Lie group \mathcal{G} and its tangent space at the identity, $T_I(\mathcal{G})$. The exponential map $\exp(\mathbf{X}) = \mathbf{Y}$ maps an element of the latter to the former (Eq. [2.13]). The inverse transform exists and is termed the logarithmic map.

This section makes use of concepts such as differential geometry and group theory, which represent full mathematical domains on their own. A few key concepts are summarised below, as opposed to diverging too much from the context of relative pose estimation. For a broader consultation on the topic, the reader is directed towards Stillwell (2008) and Hall (2015), which provide an introduction to Lie groups without necessarily delving into manifold theory. Additionally, Gallier (2011, Chap. 18) provides the basics of manifolds and Lie groups, whereas Selig (2004) discusses Lie groups in the context of robotics.

A *manifold* is a space that is topologically Euclidean (i.e. \mathbb{R}^n) on a local scope, but which is not necessarily isomorphic to \mathbb{R}^n on a global scale (an *isomorphism* $\mathcal{M} \cong \mathcal{N}$ between two structures means that there exists a mapping between them that conserves relations among elements and that can be inverted). The *tangent space* $T_x(\mathcal{M})$ of an n -dimensional manifold \mathcal{M} is an n -dimensional vector space of tangent vectors to \mathcal{M} at point $x \in \mathcal{M}$. A *group* is a set with a binary operation admitting the properties of associativity, inversion, identity element, and closure. The general linear group over the real numbers, $\text{GL}(n, \mathbb{R})$, is the group under multiplication of all $n \times n$ invertible matrices containing real entries, and is also a manifold embedded in \mathbb{R}^{n^2} . A *matrix Lie group* \mathcal{G} is a closed subgroup of $\text{GL}(n, \mathbb{R})$. The tangent space at the identity of a matrix Lie group, $T_I(\mathcal{G})$ bears an algebraic structure — the *Lie algebra* \mathfrak{g} — which provides a linearisation of \mathcal{G} and conserves many of its properties without significant information loss (see Figure 2.5).

The Lie algebra is defined as $\mathfrak{g} = \{\mathbf{X} \in M(n, \mathbb{R}) \mid \exp(t\mathbf{X}) \in \mathcal{G}, \forall t \in \mathbb{R}\}$, where $M(n, \mathbb{R})$ is the set of all real $n \times n$ matrices, invertible or not, and is closed under the *Lie bracket* $[\bullet, \bullet]$ defined such that $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA}$ for all $\mathbf{A}, \mathbf{B} \in M(n, \mathbb{R})$.

Nonetheless, it is possible, and useful, to characterise the tangent space at the identity of both $\text{SO}(3)$ and $\text{SE}(3)$. The retraction mapping $T_I(\mathcal{G}) \rightarrow \mathcal{G}$ of any Lie group \mathcal{G} is the *exponential map*, and for matrix Lie groups it corresponds to matrix exponentiation:

$$\exp(\mathbf{X}) = \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{X}^k, \quad \mathbf{X} \in \mathbb{R}^{n \times n}. \quad (2.13)$$

Whereas the Lie algebra \mathfrak{g} can be thought of as a linearisation of \mathcal{G} near the identity element, the exponential map provides a “delinearisation” back onto \mathcal{G} .

The $(\bullet)^\wedge$ operator is used to map a vector $\phi \in \mathbb{R}^3$ to the Lie algebra of $\text{SO}(3)$:

$$\begin{aligned} (\bullet)_{\mathfrak{so}(3)}^\wedge : \mathbb{R}^3 &\rightarrow \mathfrak{so}(3) \\ \phi^\wedge := \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\wedge &\mapsto \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}. \end{aligned} \quad (2.14)$$

This is occasionally found in the literature with the analogous representation $(\bullet)^\times$ since the mapping yields a 3×3 skew-symmetric matrix such that $\mathbf{a} \times \mathbf{b} = \mathbf{a}^\times \mathbf{b}$. The inverse mapping $\mathfrak{so}(3) \rightarrow \mathbb{R}^3$ is performed with the $(\bullet)^\vee$ operator. These two operators are overloaded to achieve a mapping between \mathbb{R}^6 and the Lie algebra of $\text{SE}(3)$:

$$\begin{aligned} (\bullet)_{\mathfrak{se}(3)}^\wedge : \mathbb{R}^6 &\rightarrow \mathfrak{se}(3) \\ \xi^\wedge := \begin{bmatrix} \rho \\ \phi \end{bmatrix}^\wedge &\mapsto \begin{bmatrix} \phi^\wedge & \rho \\ \mathbf{0} & 0 \end{bmatrix} \quad \rho, \phi \in \mathbb{R}^3. \end{aligned} \quad (2.15)$$

For $\text{SO}(3)$ and $\text{SE}(3)$, Equation (2.13) has a known closed form expression (R. M. Murray et al., 1994):

$$\begin{aligned} \exp_{\text{SE}(3)} : \mathfrak{se}(3) &\rightarrow \text{SE}(3) \\ \xi^\wedge &\mapsto \begin{bmatrix} \exp_{\text{SO}(3)}(\phi^\wedge) & \mathbf{N}(\phi)\rho \\ \mathbf{0} & 1 \end{bmatrix}, \end{aligned} \quad (2.16)$$

with:

$$\mathbf{N}(\boldsymbol{\phi}) := \mathbf{I}_3 + (1 - \cos \|\boldsymbol{\phi}\|)\boldsymbol{\phi}^\wedge / \|\boldsymbol{\phi}\|^2 + (\|\boldsymbol{\phi}\| - \sin \|\boldsymbol{\phi}\|)\boldsymbol{\phi}^\wedge{}^2 / \|\boldsymbol{\phi}\|^3. \quad (2.17)$$

The map $\exp_{\text{SO}(3)}$ is given by the Rodrigues rotation formula:

$$\begin{aligned} \exp_{\text{SO}(3)}: \mathfrak{so}(3) &\rightarrow \text{SO}(3) \\ \boldsymbol{\phi}^\wedge &\mapsto \mathbf{I}_3 + \frac{\sin \|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|} \boldsymbol{\phi}^\wedge + \frac{1 - \cos \|\boldsymbol{\phi}\|}{\|\boldsymbol{\phi}\|^2} (\boldsymbol{\phi}^\wedge)^2. \end{aligned} \quad (2.18)$$

The exponential map admits an inverse which is termed the *logarithmic map*:

$$\begin{aligned} \log_{\text{SE}(3)}: \text{SE}(3) &\rightarrow \mathfrak{se}(3) \\ \mathbf{T} &\mapsto \begin{bmatrix} \log_{\text{SO}(3)}(\mathbf{R}) & \mathbf{N}^{-1}\mathbf{t} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \end{aligned} \quad (2.19)$$

$$\begin{aligned} \log_{\text{SO}(3)}: \text{SO}(3) &\rightarrow \mathfrak{so}(3) \\ \mathbf{R} &\mapsto \frac{\theta}{2 \sin \theta} (\mathbf{R} - \mathbf{R}^\top), \end{aligned} \quad (2.20)$$

with $2 \cos \theta + 1 = \text{Tr}(\mathbf{R})$.

It is occasionally convenient to use the *adjoint* action of a Lie group on its Lie algebra (Selig, 2004). For $\text{SE}(3)$:

$$\begin{aligned} \text{Ad}_{\text{SE}(3)}: \text{SE}(3) &\rightarrow \mathbb{R}^{6 \times 6} \\ \mathbf{T} &\mapsto \begin{bmatrix} \mathbf{R} & \mathbf{t}^\wedge \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}. \end{aligned} \quad (2.21)$$

Let u denote a generic element of $\text{SE}(3)$. If $\mathbf{T} = \mathbf{T}(u)$ is the homogeneous representation of the group element u , then then $(\boldsymbol{\xi}')^\wedge = \mathbf{T} \boldsymbol{\xi}^\wedge \mathbf{T}^{-1}$ also yields an element of $\mathfrak{se}(3)$ and the relation can be written linearly in \mathbb{R}^6 as $\boldsymbol{\xi}' = \text{Ad}(\mathbf{T})\boldsymbol{\xi}$. Furthermore, the adjoint action of the Lie algebra on itself is:

$$\begin{aligned} \text{ad}_{\text{SE}(3)}: \mathfrak{se}(3) &\rightarrow \mathbb{R}^{6 \times 6} \\ \boldsymbol{\xi}^\wedge &\mapsto \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho}^\wedge \\ \mathbf{0} & \boldsymbol{\phi}^\wedge \end{bmatrix}, \end{aligned} \quad (2.22)$$

such that the expression for the Lie bracket of $\mathfrak{se}(3)$ can be written as $[\boldsymbol{\xi}^{(0)}, \boldsymbol{\xi}^{(1)}] := \boldsymbol{\xi}^{(0)\wedge} \boldsymbol{\xi}^{(1)} - \boldsymbol{\xi}^{(1)\wedge} \boldsymbol{\xi}^{(0)} = (\text{ad}[\boldsymbol{\xi}^{(0)\wedge}] \boldsymbol{\xi}^{(1)\wedge})^\wedge$.

2.3.1 Composition of SE(3) Elements

The \oplus operator is used to denote the composition of two poses, regardless of their representation:

$$u_{ca} = u_{cb} \oplus u_{ba}, \quad (2.23)$$

where u_{ba} maps the pose from \mathcal{F}_a to \mathcal{F}_b . The inverse operation is defined:

$$u_{ca} \ominus u_{ba} = u_{ca} \oplus u_{ba}^{-1} = u_{cb}, \quad (2.24)$$

where u_{ba}^{-1} maps the pose from \mathcal{F}_b to \mathcal{F}_a . The \oplus, \ominus operators are overloaded to designate pose-point composition:

$${}^b\mathbf{p} = u_{ba} \oplus {}^a\mathbf{p}, \quad (2.25)$$

$${}^a\mathbf{p} = {}^b\mathbf{p} \ominus u_{ba}. \quad (2.26)$$

If a Lie group \mathcal{U} is a manifold obtained through the semi-direct product of some isomorphism of $\text{SO}(3)$ and \mathbb{R}^3 (see § 2.3.2), then \mathcal{U} is isomorphic to $\text{SE}(3)$ as a manifold, but not as a group (Gallier, 2011). The “box-plus” operator (Hertzberg, 2008) $\boxplus: \mathcal{U} \times \mathbb{R}^6 \rightarrow \mathcal{U}$ is adopted to generalise a composition of a group element $u \in \mathcal{U}$ representing a pose and an element $\boldsymbol{\xi}$ which is the compact representation in \mathbb{R}^6 of $\boldsymbol{\xi}^\wedge \in \mathfrak{se}(3)$:

$$u' = u \boxplus \boldsymbol{\xi}, \quad u, u' \in \mathcal{U}, \quad \mathcal{U} \cong \text{SE}(3), \quad (2.27)$$

which is equivalent, in homogeneous form, to:

$$\mathbf{T}(u') = \exp(\boldsymbol{\xi}^\wedge) \mathbf{T}(u), \quad \mathbf{T}(u'), \mathbf{T}(u) \in \text{SE}(3). \quad (2.28)$$

Likewise, one defines the inverse operation $\boxminus: \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}^6$ that yields the compact representation of an element of the Lie algebra.

2.3.2 Manifold Isomorphisms to SE(3)

It is also useful to see $\text{SE}(3)$ as a semi-direct product of manifolds $\text{SO}(3) \times \mathbb{R}^3$, as one might be interested in working with isomorphic representations of $\text{SO}(3)$, such as the special unitary group $\text{SU}(2)$ of *unit quaternions*, with the well-known isomorphism (Markley and Crassidis, 2014):

$$\mathbf{R}(\mathbf{q}) = (q^2 - \|\mathbf{e}\|^2) \mathbf{I}_3 - 2q\mathbf{e}^\wedge + 2\mathbf{e}\mathbf{e}^\top, \quad \mathbf{q} \in \text{SU}(2), \mathbf{R} \in \text{SO}(3), \quad (2.29)$$

where \mathbf{e} and q are the vector and scalar parts of the quaternion, respectively, which takes the form:

$$\mathbf{q} := \begin{bmatrix} \mathbf{e} \\ q \end{bmatrix}, \quad \|\mathbf{q}\| = 1. \quad (2.30)$$

The unit-norm constraint in Equation (2.30) must be included in order to be an isomorphism to $\text{SO}(3)$. Henceforth, whenever a quaternion is mentioned, it is implied to be a quaternion of rotation without loss of generality. The map $\text{SU}(2) \rightarrow \text{SO}(3)$ is a 2-to-1 homomorphism, since $\{-\mathbf{q}, \mathbf{q}\}$ represent the same attitude. It can also be seen as the unit 3-sphere group \mathbb{S}^3 , i.e. the points at distance 1 from the origin in \mathbb{R}^{n+1} (Stillwell, 2008).

There are two diverging schools of thought on the definition of a quaternion for attitude representation based on the group operation of $\text{SU}(2)$. This thesis follows the norm in spacecraft attitude determination literature, i.e. Shuster's, or the "flipped", quaternion multiplication (Shuster, 1993):

$$\mathbf{q}^{(0)} \otimes \mathbf{q}^{(1)} = \begin{bmatrix} q^{(0)}\mathbf{I}_3 - \mathbf{e}^{(0)\wedge} & \mathbf{e}^{(0)} \\ -\mathbf{e}^{(0)\top} & q^{(0)} \end{bmatrix} \mathbf{q}^{(1)}, \quad (2.31)$$

such that $\mathbf{R}(\mathbf{q}^{(0)} \otimes \mathbf{q}^{(1)}) = \mathbf{R}(\mathbf{q}^{(0)})\mathbf{R}(\mathbf{q}^{(1)})$.

The quaternion has the lowest dimensionality possible for a globally non-singular attitude representation. In some applications, however, it can be seen to take the form of a three-dimensional *rotation vector* parametrisation, with the map $\text{SU}(2) \rightarrow \mathbb{R}^3$ given by (Diebel, 2006):

$$\boldsymbol{\vartheta}(\mathbf{q}) = \frac{2 \arccos(q)}{(1 - q^2)} \mathbf{e}, \quad (2.32)$$

which is in turn related to the *axis-angle* representation by:

$$\boldsymbol{\vartheta}(\alpha, \mathbf{n}) = \alpha \mathbf{n}, \quad \alpha \in \mathbb{R}, \mathbf{n} \in \mathbb{S}^2. \quad (2.33)$$

The rotation vector avoids the gimbal lock issues experienced when using Euler angles, but it is not free of singularities of its own: if the rotation angle α is zero, the axis \mathbf{n} is not uniquely defined. Furthermore, at $\alpha = \pi$ both \mathbf{n} and $-\mathbf{n}$ represent the same rotation. Note that the map $\boldsymbol{\vartheta} \mapsto \mathbf{R}$ is given by Rodrigues' rotation formula

(Eq. [2.18]). In practical terms, however, one typically has $\|\boldsymbol{\vartheta}\| \gg \|\boldsymbol{\phi}\|$, meaning that the addition of two rotation vectors does not constitute a rotation, and it should not be confused with the exponential coordinates of $\text{SO}(3)$.

2.4 On Artificial Intelligence

The notion of *artificial intelligence* (AI) is well-established amongst science, technology, engineering, and mathematics disciplines, but has recently gained notoriety in the popular media despite being often used in loose ways to refer to somewhat abstract concepts. Although historically several approaches to define AI have been followed, arguably the most relatable is a human-centred one, aiming to create machines that will be able to perform the same tasks as humans do, ideally surpassing them in performance. The epitome of this philosophy is the well-known Turing Test (Turing, 1950), devised to determine an objective baseline for machine intelligence. Today, the field has branched out in such a way that specific key areas may be identified as necessary for a computer to pass off as intelligent (Russell and Norvig, 2013):

- (1) Natural language processing to enable communication.
- (2) Knowledge representation to store what it knows or learns.
- (3) Automated reasoning to rely on the stored information to answer questions and infer new information.
- (4) *Machine learning* (ML) to adapt to novel situations and to recognise and extrapolate patterns.

The reader is directed to Russell and Norvig (2013, Chap. 1) for alternative approaches to the definition of AI.

2.4.1 Machine Learning

ML is the key component responsible for automatically processing data inside an AI. In today's age of *big data*, i.e. the increased availability of massive datasets in technology, information has never been so plentiful and available. As such, ML techniques must adapt to maximise the potential of such large volumes of data.

ML can be divided into two central types: *supervised learning* and *unsupervised learning* (Murphy, 2012). In supervised learning, the objective is to learn a mapping from inputs to outputs given labelled pairs. On the other hand, unsupervised learning has the goal to recognise noteworthy patterns in the input data, neither labels are

given nor are outputs available for comparison. Naturally, this type of ML is far less well-defined, and it is typical to observe more frequently problems posed in terms of the former approach, although the latter is more applicable. For example, much of the learning done by humans is performed simply by visual observation.

Regardless of the approach, *computer vision* is the machine-analogue for perceiving objects, and alongside *robotics* and the four other above-mentioned points, it makes up one of the six disciplines that compose most of AI (Russell and Norvig, 2013). An ability to capture raw, rich information about the environment in a simple way, allied to the fact that small, compact cameras can be bought as off-the-shelf items at affordable prices, make computer vision a very frequent candidate method to acquire input data for robotics applications.

Classical machine learning methods typically consist of two stages. The first one, *feature extraction*, involves deriving informative and discriminative subsets from the raw data, normally reducing the size of the information that needs to be post-processed. In the context of relative pose estimation using a monocular camera as the sensor, the relationship between scene and image points (§ 2.1.3, § 2.1.4) is suggestive of a need for mechanisms to extract meaningful points from said image. The second stage involves selecting a *model* that will process the engineered features to produce the desired output. Models are generated through optimisation procedures, i.e. *algorithms*, that attempt to minimise its prediction error on the data. For instance, after extracting feature points from the image of the scene, the model dictates how they are matched to the three-dimensional world points. Naturally, there are different approaches that can be taken, and in classical ML not only the model but also the feature extractor must be carefully tailored to the problem at hand. In Chapter 3, several state-of-the-art point feature detectors and descriptors are benchmarked; these search an image for interest points in scale space and encode them in a vector so that they can be afterwards matched to other features. This matching process is natively constrained to operate in the two-dimensional domain, and thus the remaining chapters are dedicated to the development of models that extend it to the three-dimensional structure of the scene.

2.4.1.1 Optimisation

The algorithmic procedures to train a ML model are embodied in a cost, or *loss function*, of which the value dictates the goodness of the fit for said model. Formally, the loss is a real valued function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and the problem is posed as:

$$\begin{aligned} & \text{minimise} && f(\boldsymbol{\theta}) \\ & \text{subject to} && \boldsymbol{\theta} \in \Omega. \end{aligned} \tag{2.34}$$

Here, $\boldsymbol{\theta}$ are independent decision variables defining the model and $\Omega \subset \mathbb{R}^n$ is the constraint set. Optimisation therefore consists in finding the best values for $\boldsymbol{\theta}$ within Ω that result in the smallest $f(\boldsymbol{\theta})$. Real-world scenarios are consistently characterised by nonlinear phenomena, in which case Equation (2.34) cannot be solved algebraically, but rather must be solved iteratively. Indeed, the projection $\pi(\tilde{\mathbf{z}}) \mapsto \mathbf{z}$ in Equation (2.7) — the basis for the pinhole camera model — is also nonlinear.

The optimisation procedure for a nonlinear function comes directly from applying the first-order necessary condition for local minimisers in the form of *Newton-Raphson's* method (Chong and Zak, 2013):

$$\boldsymbol{\theta}^{(\kappa+1)} = \boldsymbol{\theta}^{(\kappa)} - \alpha_\kappa \mathbf{F}(\boldsymbol{\theta}^{(\kappa)})^{-1} \nabla f(\boldsymbol{\theta}^{(\kappa)}), \tag{2.35}$$

where $\boldsymbol{\theta}^{(\kappa)}$ is the value of $\boldsymbol{\theta}$ obtained at time-step $\tau = \tau_\kappa$, $\nabla f(\boldsymbol{\theta})$ is the gradient of $\boldsymbol{\theta}$, and $\mathbf{F}(\boldsymbol{\theta})$ is the Hessian matrix of $\boldsymbol{\theta}$, i.e. $F_{i,j} := \partial^2 f / \partial \theta_i \partial \theta_j$. The factor α_κ is an added *step size* controlling the amount travelled in the descent direction. Newton's method (for short) is characterised by superior convergence properties when starting out with an initial guess $\boldsymbol{\theta}^{(0)}$ that is close to the minimiser $\boldsymbol{\theta}^*$. However, there is no guarantee of convergence towards decreasing values if $\mathbf{F}(\boldsymbol{\theta})$ is not positive-definite.

Remark 2.3: Conditions for Local Minimisers

The conditions leading to Newton-Raphson's method for optimisation are summarised below (Chong and Zak, 2013).

The optimisation problem can be viewed as a decision problem that involves finding the best vector $\boldsymbol{\theta}^*$ of the decision variables over all possible $\boldsymbol{\theta} \in \Omega$ such that $f(\boldsymbol{\theta})$ is minimal, i.e. $\boldsymbol{\theta}^*$ is the minimiser. If $\boldsymbol{\theta}^*$ is a local minimiser, then for any feasible direction \mathbf{s} at $\boldsymbol{\theta}^*$, one has the first-order necessary condition:

$$\mathbf{s}^\top \nabla f(\boldsymbol{\theta}^*) \geq 0 \Rightarrow \frac{\partial f}{\partial \mathbf{s}}(\boldsymbol{\theta}^*) \geq 0, \tag{2.36}$$

i.e. the rate of increase of f at $\boldsymbol{\theta}^*$ in any feasible direction \mathbf{s} inside Ω is non-negative. Taking the Taylor series expansion of f about a current point $\boldsymbol{\theta}^{(\kappa)}$

yields:

$$f(\boldsymbol{\theta}) \approx f(\boldsymbol{\theta}^{(\kappa)}) + (\boldsymbol{\theta} - \boldsymbol{\theta}^{(\kappa)})^\top \mathbf{g}^{(\kappa)} + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}^{(\kappa)})^\top \mathbf{F}(\boldsymbol{\theta}^{(\kappa)}) (\boldsymbol{\theta} - \boldsymbol{\theta}^{(\kappa)}) := q(\boldsymbol{\theta}), \quad (2.37)$$

where $\mathbf{g}^{(\kappa)} := \nabla f(\boldsymbol{\theta}^{(\kappa)})$. If $\boldsymbol{\theta}^*$ is an interior point of Ω , then any \mathbf{s} is feasible:

$$\mathbf{s}^\top \nabla f(\boldsymbol{\theta}^*) \wedge -\mathbf{s}^\top \nabla f(\boldsymbol{\theta}^*) \geq 0 \Rightarrow \mathbf{s}^\top \nabla f(\boldsymbol{\theta}^*) = 0 \Rightarrow \nabla f(\boldsymbol{\theta}^*) = \mathbf{0}. \quad (2.38)$$

Applying this condition to Equation (2.37) gives:

$$0 = \nabla q(\boldsymbol{\theta}) = \mathbf{g}^{(\kappa)} + \mathbf{F}(\boldsymbol{\theta}^{(\kappa)}) (\boldsymbol{\theta} - \boldsymbol{\theta}^{(\kappa)}), \quad (2.39)$$

which, after re-ordering, leads to Equation (2.35). The second-order necessary condition for $\boldsymbol{\theta}^*$ to be a minimiser of f asserts:

$$\mathbf{s}^\top \mathbf{F}(\boldsymbol{\theta}^*) \geq 0, \quad (2.40)$$

i.e. $\mathbf{F}(\boldsymbol{\theta}^*)$ is positive semidefinite. Both of these conditions, however, are not sufficient; the second order sufficient condition dictates that if $\nabla f(\boldsymbol{\theta}^*) = \mathbf{0}$, then

$$\mathbf{F}(\boldsymbol{\theta}^*) > 0 \quad (2.41)$$

is sufficient (albeit not necessary) for $\boldsymbol{\theta}^*$ to be a local minimiser of f . Furthermore, it is a strict local minimiser of f .

Newton's method is commonly modified with the addition of a step size α_κ controlling the magnitude of descent along \mathbf{s} such that

$$\alpha_\kappa = \arg \min_{\alpha \geq 0} f\left(\boldsymbol{\theta}^{(\kappa)} - \alpha \mathbf{F}(\boldsymbol{\theta}^{(\kappa)})^{-1} \mathbf{g}^{(\kappa)}\right), \quad (2.42)$$

in order to ensure that $f(\boldsymbol{\theta}^{(\kappa+1)}) - f(\boldsymbol{\theta}^{(\kappa)}) < 0$ for $\mathbf{g}^{(\kappa)} \neq \mathbf{0}$.

To combat this and robustify the optimisation procedure, other iterative methods have been developed. The *Levenberg-Marquardt* (LM) method, in particular, is a technique that ensures the search direction $\mathbf{s}^{(\kappa)} = -\mathbf{F}(\boldsymbol{\theta}^{(\kappa)})^{-1} \mathbf{g}^{(\kappa)}$ is a descent direction by introducing low-magnitude noise along the diagonal of the Hessian before inversion:

$$\boldsymbol{\theta}^{(\kappa+1)} = \boldsymbol{\theta}^{(\kappa)} - \alpha_{\kappa} (\mathbf{F}(\boldsymbol{\theta}^{(\kappa)}) + \mu_{\kappa} \mathbf{I})^{-1} \mathbf{g}^{(\kappa)}, \quad (2.43)$$

where $\mu_{\kappa} > 0$ is a small, perturbative constant, and \mathbf{I} is the identity matrix of appropriate dimension. The idea behind the algorithm is that adding the term $\mu \mathbf{I}$ increments the eigenvalues of \mathbf{F} by μ . Therefore, for a sufficiently large μ , the eigenvalues of $\mathbf{F} + \mu \mathbf{I}$ are always positive and hence the new matrix is positive-definite.

Nonlinear Least Squares

As introduced in Section 2.1.4, in the face of flawless geometric correspondence between world points and image points, the true relative camera pose satisfies the relationship between the two (i.e. Eq. [2.7] is verified). However, the consideration of real, physical scenarios implies that sensor data \mathbf{x} are in fact subject to measurement errors and, hence, non-ideal. Therefore, it becomes necessary to model such an error in order to obtain an optimal estimate of the parameters $\boldsymbol{\theta}$.

Due to the ensuing tractability, it is commonplace to assume that such measurement errors follow a Gaussian probability distribution (Hartley and Zisserman, 2004). More specifically, the measurement error can be modelled as a zero-mean isotropic Gaussian variable, independent on each feature x_i constituting \mathbf{x} :

$$x_i = \bar{x}_i + \Delta x, \quad \Delta x \sim \mathcal{N}(0, \sigma^2), \quad (2.44)$$

where σ is the standard deviation and \bar{x}_i represents the i -th component of the true measured quantity $\bar{\mathbf{x}}$. The probability density function (PDF) of the measurement \mathbf{x} is then:

$$p(\mathbf{x}) = \left(\frac{1}{2\pi\sigma^2} \right) \exp \left(-\frac{1}{2\sigma^2} (\mathbf{x} - \bar{\mathbf{x}})^{\top} (\mathbf{x} - \bar{\mathbf{x}}) \right). \quad (2.45)$$

Obtaining a solution for $\boldsymbol{\theta}$ can then be approached as a task minimising the distance between the measurement, \mathbf{x} , and the image of a function $h(\boldsymbol{\theta}) \mapsto \bar{\mathbf{x}}$ that generates the ideal data point based on the model parameters. This formulation gives meaning to the ‘‘optimality’’ condition from the above-mentioned paragraph as it provides the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$. The proof is almost direct, based on the Gaussian error assumption; since the error on each measurement is considered to be independent, the PDF of the full set $\mathbb{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ of noisy data is the product of the PDF of each individual correspondence. Denoting the *residual* of the i -th correspondence as the difference $\mathbf{r}^{(i)} := \mathbf{x}^{(i)} - h(\boldsymbol{\theta})$, one can write:

$$p(\mathbb{X} | \boldsymbol{\theta}) = \prod_i \left(\frac{1}{2\pi\sigma^2} \right) \exp \left(-\frac{1}{2\sigma^2} \mathbf{r}^{(i)\top} \mathbf{r}^{(i)} \right). \quad (2.46)$$

Taking the log-likelihood of Equation (2.46) yields:

$$\log [p(\mathbb{X} | \boldsymbol{\theta})] = -\frac{1}{2\sigma^2} \sum_i \mathbf{r}^{(i)\top} \mathbf{r}^{(i)} + \text{const.} \quad (2.47)$$

Therefore, maximising the log-likelihood is equivalent to the formulation:

$$\min \sum_i \mathbf{r}^{(i)\top} \mathbf{r}^{(i)}, \quad (2.48)$$

and the **MLE** thus minimises the residual sum of squares. For this reason, the problem is also termed one of *nonlinear least squares* (**NLLS**).

Concatenating the individual residuals into a single vector $\mathbf{r}^\top = [\mathbf{r}^{(1)\top} \ \dots \ \mathbf{r}^{(m)\top}] = [r_1^{(1)} \ \dots \ r_d^{(1)} \ \dots \ r_d^{(m)}]$, where d is the dimensionality of \mathbf{x} , Newton's method becomes:

$$\boldsymbol{\theta}^{(\kappa+1)} = \boldsymbol{\theta}^{(\kappa)} - (\mathbf{J}(\boldsymbol{\theta})^\top \mathbf{J}(\boldsymbol{\theta}) + \mathbf{S}(\boldsymbol{\theta}))^{-1} \mathbf{J}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}), \quad (2.49)$$

where the step size has been omitted. The matrix $\mathbf{J}(\boldsymbol{\theta})$ is the *Jacobian* of \mathbf{r} with respect to $\boldsymbol{\theta}$:

$$\mathbf{J}(\boldsymbol{\theta}) := \begin{bmatrix} \frac{\partial r_1}{\partial \theta_1}(\boldsymbol{\theta}) & \dots & \frac{\partial r_1}{\partial \theta_n}(\boldsymbol{\theta}) \\ \vdots & \ddots & \vdots \\ \frac{\partial r_{md}}{\partial \theta_1}(\boldsymbol{\theta}) & \dots & \frac{\partial r_{md}}{\partial \theta_n}(\boldsymbol{\theta}) \end{bmatrix}, \quad (2.50)$$

and the matrix $\mathbf{S}(\boldsymbol{\theta})$ is defined in index notation as:

$$S(\boldsymbol{\theta})_{j,k} := \sum_{i=1}^{md} r_i(\boldsymbol{\theta}) \frac{\partial^2 r_i}{\partial \theta_j \partial \theta_k}(\boldsymbol{\theta}). \quad (2.51)$$

Analogously, the **LM** formulation becomes:

$$\boldsymbol{\theta}^{(\kappa+1)} = \boldsymbol{\theta}^{(\kappa)} - (\mathbf{J}(\boldsymbol{\theta})^\top \mathbf{J}(\boldsymbol{\theta}) + \mu_\kappa \mathbf{I})^{-1} \mathbf{J}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}). \quad (2.52)$$

The term $\mu \mathbf{I}$ can therefore be interpreted as an approximation to $\mathbf{S}(\boldsymbol{\theta})$ avoiding the computation of second-order derivatives.

Remark 2.4: Derivation of Equation (2.49)

First, the cost function (see Eq. [2.34]) is rewritten as:

$$f(\boldsymbol{\theta}) = \mathbf{r}^\top \mathbf{r}. \quad (2.53)$$

Then, the gradient of f can be computed in index notation:

$$\begin{aligned} (\nabla f(\boldsymbol{\theta}))_j &= \frac{\partial f}{\partial \theta_j}(\boldsymbol{\theta}) \\ &= 2 \sum_{i=1}^{md} r_i(\boldsymbol{\theta}) \frac{\partial r_i}{\partial \theta_j}(\boldsymbol{\theta}) \\ &\Rightarrow \nabla f(\boldsymbol{\theta}) = 2\mathbf{J}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}). \end{aligned} \quad (2.54)$$

To develop an expression for the Hessian, index notation is again used with the results from above:

$$\begin{aligned} F(\boldsymbol{\theta})_{j,k} &= \frac{\partial^2 f}{\partial \theta_j \partial \theta_k}(\boldsymbol{\theta}) \\ &= 2 \left(\sum_{i=1}^{md} \frac{\partial r_i}{\partial \theta_k}(\boldsymbol{\theta}) \frac{\partial r_i}{\partial \theta_j}(\boldsymbol{\theta}) + \sum_{i=1}^{md} r_i(\boldsymbol{\theta}) \frac{\partial^2 r_i}{\partial \theta_k \partial \theta_j}(\boldsymbol{\theta}) \right) \\ &\Rightarrow \mathbf{F}(\boldsymbol{\theta}) = 2(\mathbf{J}(\boldsymbol{\theta})^\top \mathbf{J}(\boldsymbol{\theta}) + \mathbf{S}(\boldsymbol{\theta})). \end{aligned} \quad (2.55)$$

Substituting the derived quantities in Equation (2.35) finally leads to Equation (2.49).

On-Manifold Optimisation

In most applications, it is prevalent to assume that the parameter vector is Euclidean (i.e. $\boldsymbol{\theta} \in \mathbb{R}^n$). In this case, the iterative NLLS update derived from Equation (2.48) yields a correction that can be added to the previous estimate of $\boldsymbol{\theta}$ to obtain the next one. Taking the LM normal equations, for example:

$$\Delta \boldsymbol{\theta} = -(\mathbf{J}(\boldsymbol{\theta})^\top \mathbf{J}(\boldsymbol{\theta}) + \mu_\kappa \mathbf{I})^{-1} \mathbf{J}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}), \quad (2.56)$$

$$\boldsymbol{\theta}^{(\kappa+1)} = \boldsymbol{\theta}^{(\kappa)} + \Delta \boldsymbol{\theta}. \quad (2.57)$$

Concretely, the corrective step of Equation (2.57) takes for granted that $\boldsymbol{\theta} \in \Omega$ (cf. Eq. [2.34]), which is not necessarily true if the domain Ω is non-Euclidean. On the

other hand, the theory developed in this chapter has so far highlighted the importance of the tangent space as a local vector space approximation for the pose manifold $\text{SE}(3)$. Let R_x be a retraction at $x \in \mathcal{M}$ that maps $T_x(\mathcal{M}) \rightarrow \mathcal{M}$. In addition, let 0_x be the 0-element of $T_x(\mathcal{M})$ such that $R_x(0_x) = x$, and ϕ a real-valued function acting in \mathcal{M} . Then, if \mathcal{M} is endowed with a Riemannian metric, one can write:

$$\nabla(\phi \circ R_x)(0_x) = \nabla\phi(x), \quad (2.58)$$

i.e. the retraction preserves gradients at x (Absil et al., 2008). This means that optimisation problems based on Euclidean spaces relying on the computation of gradients (or some approximation thereof), such as NLLS, can be generalised to (nonlinear) manifolds via retraction mappings. Since any Lie group can be endowed with a Riemannian metric (Gallier and Quaintance, 2019), Equation (2.58) applies to the problem of pose estimation, and the exponential map could be used as the bridge to locally convert an optimisation problem stated in terms of \mathbf{T} to the more tractable vector space of the corresponding Lie algebra element ξ^\wedge (or simply its compact representation $\xi \in \mathbb{R}^6$), using the composition theory of Section 2.3.1, where methods of Euclidean analysis can be used. This section therefore reformulates the NLLS problem into the general case where Ω is a manifold (e.g. $\text{SE}(3)$).

Let u represent the object to be determined. Its domain is a manifold $\mathcal{U} \subset \mathbb{R}^m$ which defines the *parameter space*. Let \mathbf{x} be the vector of measurements in \mathbb{R}^n . Consider also the general case where \mathbf{x} is observed in the presence of noise with a covariance matrix $\Sigma_{\mathbf{x}}$ (cf. Eq. [2.44]), and let $\bar{\mathbf{x}}$ be its true value, i.e. $\mathbf{x} = \bar{\mathbf{x}} + \Delta\mathbf{x}$. Let $h: \mathbb{R}^m \rightarrow \mathbb{R}^n$ be a mapping such that, in the absence of noise, $h(\bar{u}) = \bar{\mathbf{x}}$. Varying the value of \bar{u} traces out a manifold $\mathcal{X} \subset \mathbb{R}^n$ defining the set of allowable measurements, i.e. the *measurement space*. The objective is, given a measurement \mathbf{x} , to find the vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ lying on \mathcal{X} that is closest to \mathbf{x} (Figure 2.6).

Given the form of the multivariate Gaussian probability density function, under the assumption of Gaussian noise with covariance matrix $\Sigma_{\mathbf{x}}$, the MLE now minimises the squared *Mahalanobis distance* (cf. Eq. [2.45]):

$$d_{\Sigma}(\mathbf{x}, h(\hat{u}))^2 := (\mathbf{x} - h(\hat{u}))^\top \Sigma_{\mathbf{x}}^{-1} (\mathbf{x} - h(\hat{u})). \quad (2.59)$$

As prefaced in Section 2.3, it is assumed that, in the neighbourhood of \mathbf{x} , the surface of \mathcal{X} is essentially planar and well approximated by the tangent space; concretely, this approximation is reasonable within the order of magnitude of the measurement noise variance (Hartley and Zisserman, 2004). Then, the maximum likelihood corrected measurement $\hat{\mathbf{x}}$ is the foot of the perpendicular from \mathbf{x} onto the tangent plane.

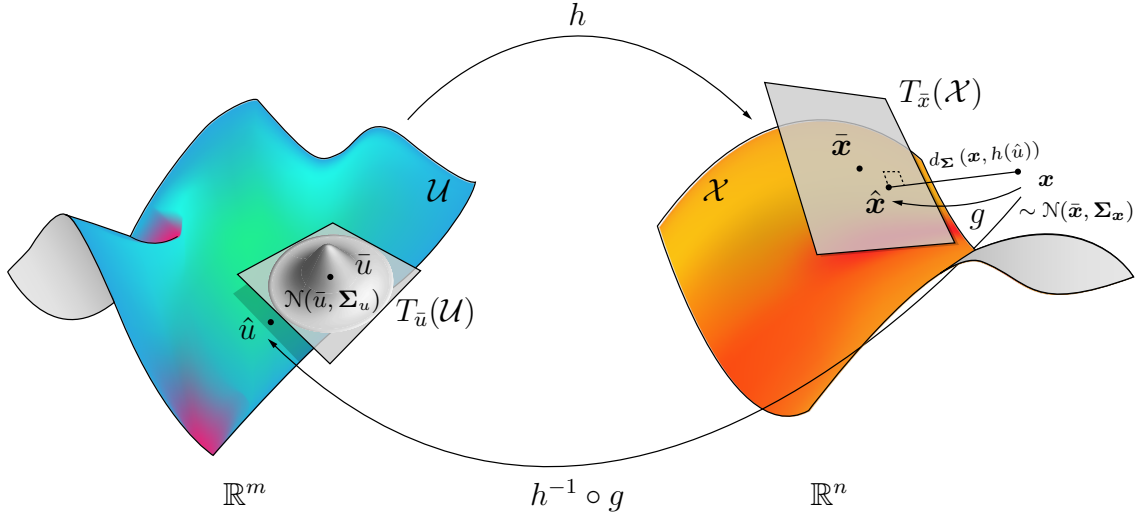


Figure 2.6: Geometric correction and parametric fitting on manifolds. The measurement space manifold \mathcal{X} in \mathbb{R}^n is the image of the parameter space \mathcal{U} through h . Under assumptions of Gaussian noise, the optimal solution involves estimating a corrected measurement $\hat{\mathbf{x}}$ from the noisy \mathbf{x} as the perpendicular to the tangent space of \mathcal{X} at the true value $\bar{\mathbf{x}}$. This yields the maximum likelihood parameters \hat{u} in \mathbb{R}^m which is a random variable with a local distribution on the tangent space of \mathcal{U} at \hat{u} .

The benefit of this approximation is that it allows the measurement residual error to be modelled as a Gaussian distribution in the normal space of \mathcal{X} , whereas the measurement estimation error is a Gaussian distribution in the tangent space $T_{\bar{\mathbf{x}}}(\mathcal{X})$. In the domain of computer vision, since image (or world) points are taken as measured variables, typically one can safely write the estimation error as $\hat{\mathbf{x}} - \bar{\mathbf{x}}$. Analogously, the parameter estimation error $\delta \mathbf{u}$ is, to a first approximation, constrained to be in the tangent space $T_{\bar{u}}(\mathcal{U})$. In general terms, the assumption for this thesis shall be that this local distribution approximation is valid for small errors whenever dealing with the probability distribution of a variable constrained to a manifold (Kanatani, 1996). Let $g: \mathbb{R}^n \rightarrow \mathcal{X}$ map a point to the surface of the measurement space, as defined in Equation (2.59). Assuming that h is invertible such that $h^{-1}: \mathcal{X} \rightarrow \mathbb{R}^m$, then the mapping $h^{-1} \circ g$ can be used to propagate the measurement noise covariance $\Sigma_{\mathbf{x}}$ to obtain the covariance matrix of the MLE \bar{u} , which has the following form:

$$\Sigma_u = \mathbf{P}_{\bar{u}}^{\mathcal{U}} \mathbb{E} [(u - \bar{u})(u - \bar{u})^\top] \mathbf{P}_{\bar{u}}^{\mathcal{U}}, \quad (2.60)$$

where $\mathbf{P}_{\bar{u}}^{\mathcal{U}}$ is the projection matrix onto $T_{\bar{u}}(\mathcal{U})$ and $\mathbb{E}[\bullet]$ represents the expected value. This implies that the null space of Σ_u coincides with the normal space of \mathcal{U} at \bar{u} .

Taking $f(u) = d_{\Sigma}(\mathbf{x}, h(u))^2$, the general NLLS problem can be posed as:

$$\hat{u} = \arg \min_{u \in \mathcal{U}} f(u), \quad (2.61)$$

where, in the particular case of $\Sigma = \sigma^2 \mathbf{I}$ and \mathcal{U} being identical to the Euclidean space, the update equations are given in Equation (2.57). Otherwise, one possible solution is to nevertheless apply the correction via standard addition and then project the result back to the parameter manifold \mathcal{U} , which could introduce additional noise in the system and drive the result away from the MLE \hat{u} . An example of this would be parametrising the attitude quaternion as a vector in \mathbb{R}^4 , correcting it with vector addition and then enforcing quaternion normalisation on the result (see § 2.3.2). A more elegant alternative solution is to exploit the local Euclidean structure of \mathcal{U} around $u^{(\kappa)}$ at the current time-step $\tau = \tau_\kappa$ to generate a new set of normal equations. Taking $\mathcal{U} \cong \text{SE}(3)$ and using the composition operator from Section 2.3.1, linearising $f(u)$ yields:

$$f(u) \approx f(u^{(\kappa)}) + (u \boxminus u^{(\kappa)})^\top \nabla f \Big|_{u \ominus u^{(\kappa)} = \mathbf{0}}. \quad (2.62)$$

Equation (2.62) thus motivates working with the pose estimation error $\delta \mathbf{u} = u \boxminus u^{(\kappa)}$ explicitly, which is an element of $\mathfrak{se}(3)$. The generalised normal equations take the form:

$$\mathbf{J}(u)^\top \Sigma_{\mathbf{x}} \mathbf{J}(u) \delta \mathbf{u} = -\mathbf{J}(u)^\top \Sigma_{\mathbf{x}} \mathbf{r}(u), \quad (2.63)$$

where $\mathbf{r} = h(u^{(\kappa)}) - \mathbf{x}$ is the residual vector. The elementary building block for on-manifold optimisation is the 12×6 Jacobian of $\text{SE}(3)$ (Blanco, 2019; Hertzberg, 2008). Taking $\boldsymbol{\xi} = \delta \mathbf{u}$, it has the form:

$$\frac{\partial \exp(\boldsymbol{\xi}^\wedge)}{\partial \boldsymbol{\xi}} \Big|_{\boldsymbol{\xi}=\mathbf{0}} = \frac{\partial \text{vec}[\exp(\boldsymbol{\xi}^\wedge)]}{\partial \boldsymbol{\xi}} \Big|_{\boldsymbol{\xi}=\mathbf{0}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{e}^{(1)\wedge} \\ \mathbf{0}_{3 \times 3} & -\mathbf{e}^{(2)\wedge} \\ \mathbf{0}_{3 \times 3} & -\mathbf{e}^{(3)\wedge} \\ \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad (2.64)$$

where $\mathbf{e}^{(1)\top} = [1 \ 0 \ 0]$, $\mathbf{e}^{(2)\top} = [0 \ 1 \ 0]$, $\mathbf{e}^{(3)\top} = [0 \ 0 \ 1]$, and the $\text{vec}(\bullet)$ operator vertically stacks the column vectors of a matrix:

$$\text{vec}(\mathbf{A}) := \begin{bmatrix} \mathbf{A}_{:,1} \\ \vdots \\ \mathbf{A}_{:,n} \end{bmatrix}, \quad \mathbf{A} \in \mathbb{R}^{n \times m}, \text{vec}(\mathbf{A}) \in \mathbb{R}^{nm \times 1}. \quad (2.65)$$

The other advantage of Equation (2.63) is that the Jacobian matrix is computed

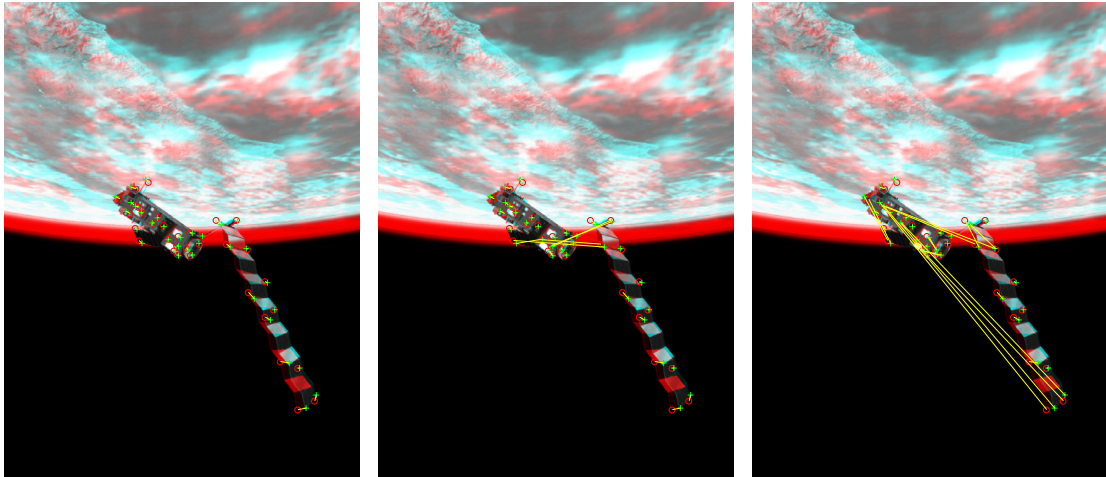


Figure 2.7: Sequential feature matching in false-color composite overlay. The first frame is shown as red and the second as cyan. Feature points are detected in both frames and the matches are shown in yellow. (*Left*) 0% outliers; all matches are correct and the yellow lines represent accurately the motion of the target. (*Centre*) 25% outliers; some features are matched incorrectly, but the overall motion can still be discerned. (*Right*) 50% outliers; the apparent trajectory becomes increasingly garbled.

with respect to the basis of $\mathfrak{se}(3)$:

$$\mathbf{J}(u) := \left. \frac{\partial h(u^{(\kappa)} \boxplus \boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right|_{\boldsymbol{\xi}=\mathbf{0}} \quad (2.66)$$

This class of Jacobian matrices is solvable by resorting to Equation (2.64) and the chain rule. At the end of each iteration, the updated parameter is obtained via the exponential map (Eq. [2.27]), thus ensuring it naturally remains an element of \mathcal{U} .

Robust Estimation

When the measurements are assumed to be equivariant with respect to scale, Equation (2.61) falls back to the minimisation of the residual sum of squares, with the underlying metric being the L^2 distance, as the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}}$ vanishes. However, the ordinary least squares problem is not robust to *outliers*, i.e. spurious data that may contaminate the measurements. If measurements correspond to detected features, outliers correspond in this case to gross 2D localisation errors in the image plane, such that the true correspondences to the world points become compromised (cf. Fig. 2.2). In the case that the world points are very accurately known (e.g. from a model of the target), these errors can be assumed to be contained in the feature matching process, and outliers correspond in effect to erroneous correspondences. In the scope of this thesis, this can arise in a typical space rendezvous scenario, as

imaged by a camera, from a solar panel which might resemble a repeating pattern that yields many identically-looking features, or from intense illumination from the Sun acting on the spacecraft, changing its local aspect with respect to a model image. Consider Figure 2.7, where sequential frames of a simulated rendezvous sequence are represented with a false-color overlay. Point feature matches are also represented and connected by lines for several levels of outlier contamination. Whereas, from the reader's perspective, it may seem that for a 25% outlier level (middle image) the true trajectory can still be determined, in theory the presence of a single outlier is enough to make the least squares estimate diverge (Rousseeuw and Leroy, 1987).

Robustness with respect to outliers can be achieved by generalizing Equation (2.61) into an *M-estimator*:

$$\hat{u} = \arg \min_{u \in \mathcal{U}} \sum_{i=1}^N \rho \left(\frac{r_i}{\hat{\sigma}} \right), \quad (2.67)$$

where ρ is a symmetric, positive-definite function with subquadratic growth, and $\hat{\sigma}^2$ is an estimate of the variance, or scale, of \mathbf{r} . Solving Equation (2.67) implies

$$\sum_{i=1}^N \psi \left(\frac{r_i}{\hat{\sigma}} \right) \frac{dr_i}{du} \frac{1}{\hat{\sigma}} = 0, \quad (2.68)$$

where $\psi(x) := d\rho(x)/dx$ is defined as the influence function of the M-estimator. This function measures the influence that a data point has on the estimation of the parameter u . A robust M-estimator $\rho(x)$ should meet two constraints: convexity in x , and a bounded influence function (Z. Zhang, 1997). By acknowledging the latter point, it becomes clear why the general least squares is not robust, since $\rho(x) = x^2/2$ and therefore $\psi(x) = x$.

There are two possible approaches to define the normal equations for M-estimation that avoid the computation of the Hessian (Holland and Welsch, 1977):⁴

$$\mathbf{J}^\top \mathbf{J} \delta \mathbf{u} = -\mathbf{J}^\top \psi \left(\frac{\mathbf{r}}{\hat{\sigma}} \right) \hat{\sigma}, \quad (2.69)$$

$$\mathbf{J}^\top \mathbf{W} \mathbf{J} \delta \mathbf{u} = -\mathbf{J}^\top \mathbf{W} \mathbf{r}, \quad (2.70)$$

where $\mathbf{W} = \text{diag}(w(r_1/\hat{\sigma}), \dots, w(r_n/\hat{\sigma}))$ and $w(x) := \psi(x)/x$. The first method was developed by P. Huber (1977) and generalises the normal equations through the

⁴Holland and Welsch (1977) define a third formulation, which is based on Newton's method, but is difficult to implement since it requires the computation of $d\psi(x)/dx$ and the Hessian risks being negative definite.

modification of the residuals via ψ and $\hat{\sigma}$. Huber proposed a specific loss function, the Huber M-estimator:

$$\rho_{\text{Hub}}(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq c, \\ c \left(|x| - \frac{c}{2} \right) & \text{otherwise,} \end{cases} \quad (2.71)$$

where c is a tuning constant. Huber's algorithm provides a way to jointly estimate the scale σ alongside the parameter u with proven convergence properties. The minimisation algorithm (e.g. LM) is simply appended with the procedure:

$$\sigma_{\kappa+1}^2 = \frac{1}{(n-p)\beta} \sum_i^n \left(\frac{r_i}{\sigma_\kappa} \right)^2 \sigma_\kappa^2 \quad (2.72)$$

where β is a bias-correcting factor and p is the dimensionality of u . The second method was developed by Beaton and Tukey (1974) and is commonly known as *iteratively reweighed least squares* (IRLS), due to the inclusion of the weights matrix \mathbf{W} that assumes the role of $\Sigma_{\mathbf{x}}$ (cf. Eq. [2.63]). Tukey proposed an alternative robust loss function:

$$\rho_{\text{Tuk}}(x) = \begin{cases} \frac{c^2}{6} \left(1 - \left(1 - \left(\frac{x}{c} \right)^2 \right)^3 \right) & \text{if } |x| \leq c, \\ \frac{c^2}{6} & \text{otherwise,} \end{cases} \quad (2.73)$$

Each robust loss function, $\rho_{\text{Hub}}(x)$ and $\rho_{\text{Tuk}}(x)$, can be compared regardless of the formulation. The Huber M-estimator is considered to be adequate for almost all situations, but does not eliminate completely the influence of large errors (Z. Zhang, 1997). On the other hand, the Tukey M-estimator is non-convex, but is a “hard redescender”, meaning that its influence function tends to zero quickly so as to aggressively reject outliers, explaining its frequent use in computer vision applications, where the outliers typically have small residual magnitudes (Stewart, 1999).

Remark 2.5: On Influence Functions

The advantage of M-estimators is perhaps better visualised by plotting their response against the classical least squares. From Figure 2.8, it can be seen that the Huber and Tukey functions behave like least squares in a local neighbourhood around the origin. However, by looking at the influence function $\psi(x)$, it can be seen that when the residual grows too much away from zero,

the derivative becomes constant. On the other hand, the influence function of least squares grows continually away from zero, meaning that a single outlier can have an unbounded influence on the estimation.

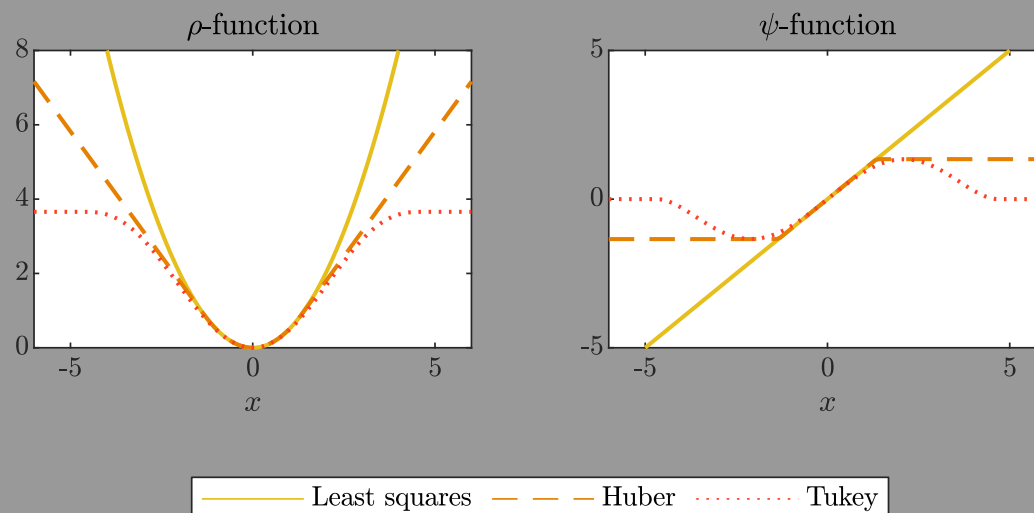


Figure 2.8: Comparison between the least squares cost function and the Huber (Eq. [2.71]) and Tukey (Eq. [2.73]) M-estimators, and their respective influence functions. The curves are plotted for $c_{\text{Hub}} = 1.345$ and $c_{\text{Tu}} = 4.6851$, which corresponds to 95% asymptotic efficiency on the standard normal distribution (Z. Zhang, 1997).

Remark 2.6: On RANSAC

One popular method for outlier rejection is Random Sample Consensus (RANSAC; Fischler and Bolles, 1981), which works by iteratively sampling random data points according to the minimum quantity needed to solve the problem in closed form (i.e. four features in the case of PnP) and measuring the quality of the derived model using the remaining points (e.g. the reprojection error in pixels). The model providing the best fit is kept. By defining a threshold for the allowable error in each point, a set of inliers can be defined. These are then often used in a post-processing step based on iterative least squares (LS) to refine the model.

RANSAC is non-deterministic and the probability of reducing the model error increases with the number of allowed iterations. Conversely, M-estimation combines the benefits of outlier rejection and optimality by iteratively finding a cost function minimum. Therefore, when using M-estimation, RANSAC

becomes unnecessary provided that a good initial estimate of the solution can be assumed.

2.4.2 Deep Learning

Deep learning is a subset of **ML** in which a model learns directly from the raw data as input, skipping the traditional custom feature extraction step. It is a term used tantamount to *artificial neural networks* (**ANNs**), as the earliest iterations were vaguely inspired by the biological brain’s functions (I. Goodfellow et al., 2016). The basic **ANN** model is the *multilayer perceptron* (**MLP**; Bishop, 2006): to generate a vector of outputs $\mathbf{y} = [y_1 \dots y_K]^\top$, M linear combinations of the elements of the input $\mathbf{x} = [x_1 \dots x_D]$ are established and the result is passed through a nonlinear activation function h :

$$a_j = \sum_{i=1}^D W_{j,i}^{(1)} x_i + b_j^{(1)}, \quad j = 1, \dots, M \quad (2.74)$$

$$z_j = h(a_j), \quad (2.75)$$

where $W_{j,i}, b_j$ are learnable parameters of the network, i.e. weights and biases,⁵ and the superscript (1) refers to the first layer of the **MLP**. For the simplest case in which the **ANN** only has one *hidden layer*, the output activations are then directly obtained from z_j through another linear combination:

$$a_k = \sum_{j=1}^M W_{k,j}^{(2)} z_j + b_k^{(2)}, \quad k = 1, \dots, K. \quad (2.76)$$

Depending on the nature of the responses, the output activations can be passed through another nonlinear activation unit, or taken as the identity $y_k = a_k$ (see Fig. 2.9). Due to the development of efficient optimisation techniques and advances in consumer-grade computing capability, mainly regarding *graphics processing units* (**GPUs**), models with a larger number of hidden layers (i.e. more depth) were capable of being trained in reasonable timespans, prompting a resurgence of research in the field occurring in the late 2000s, and hence popularising the use of “deep” learning, or *deep neural networks* (**DNNs**), as a synonym for neural networks.

Like in classical **ML**, the training of a **DNN** is formulated in terms of minimising a scalar loss function $f(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of learnable parameters of the

⁵Note that Equation (2.74) is simply a linear regression, where b_j corresponds to the y -intercept; hence, the same **ML** terminology is conserved.

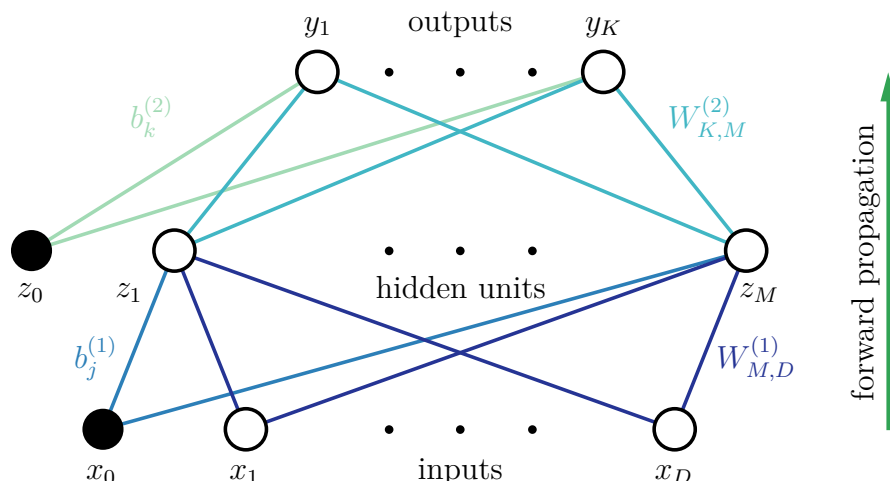


Figure 2.9: Diagram for a two-layer artificial neural network (ANN). The open nodes represent input (\mathbf{x}), hidden (\mathbf{z}), and output (\mathbf{y}) variables, whereas weights (\mathbf{W}) are represented by links connecting nodes. Biases (b) are also represented as links which come from additional closed nodes (x_0, z_0). Cf. Equations (2.74) to (2.76). Adapted from Bishop (2006).

network and the dependency on the inputs and outputs has been made explicit. As a DNN is nonlinear by design (recall Eq. [2.75]), finding the optimal θ is achieved through iterative methods by searching for critical points in f using the gradient information; the simplest approach is taking a small step in θ -space in the direction of the negative gradient, i.e. *gradient descent* (Bishop, 2006):

$$\theta^{(\kappa+1)} = \theta^{(\kappa)} - \alpha_\kappa \nabla_{\theta} f, \quad (2.77)$$

where α_κ is the learning rate at time-step $\tau = \tau_\kappa$. The main advantage of Equation (2.77) is that it does not require the computation of the Hessian (cf. Eq. [2.35]).

One forward pass through the DNN will yield the sufficient conditions to compute the gradient of the loss with respect to the output, i.e. $\nabla_{\mathbf{y}} f$. The gradient of f with respect to the weights in each layer, necessary for the gradient descent optimisation in Equation (2.77), can be found by successively chaining the local gradients of each layer in the reverse direction until the desired layer is reached (*backpropagation*; Rumelhart et al., 1985). If a unit j in one layer sends connections to k units in the next layer, then the local gradient at j is given by:

$$\frac{\partial f}{\partial a_j} = \sum_k \frac{\partial f}{\partial a_k} \frac{\partial a_k}{\partial a_j}. \quad (2.78)$$

Since the values of the gradient for the output units are known with a forward pass, by recursive backpropagation the gradients for every hidden layer can be efficiently

computed regardless of the [DNN](#) model.

As gradient-based parameter optimisation for large datasets is often prohibitive from a memory standpoint, at each step a *minibatch* $\mathbb{B} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ of m inputs can be sampled from the training data set and used instead. In this case, the procedure becomes a *stochastic gradient descent* ([SGD](#)), and the gradient is estimated as:

$$\nabla_{\boldsymbol{\theta}} f \approx \frac{1}{m} \sum_{i=1}^m \nabla_{\boldsymbol{\theta}} f(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \boldsymbol{\theta}). \quad (2.79)$$

An *epoch* passes once the [SGD](#) processes all minibatches comprising the entire dataset of inputs. Typically, a [DNN](#) requires several epochs to train.

A linear output such as Equation (2.76) allows a neural network to learn regression problems. However, much of the popularity exuded by deep learning today comes from its potential performance for *classification* tasks (i.e. a discrete and generally mutually exclusive output), namely in those where a selection is performed from a large pool of choices. Whereas the sigmoid function (Figure 2.10, middle) is used to model an output obeying a Bernoulli distribution (e.g. a coin toss, telling a dog apart from a cat, or distinguishing between an operational and a failed satellite), the *softmax* function, given by

$$\text{softmax}(\mathbf{a})_k := \frac{\exp(a_k)}{\sum_{\ell} \exp(a_{\ell})}, \quad (2.80)$$

is used to model categorical, or generalised Bernoulli, distributions (e.g. distinguishing between several breeds of cat, or differentiating between multiple satellite models).

2.4.2.1 Activation Functions

The earliest [ANNs](#) generally made use of either sigmoid or hyperbolic tangents (Figure 2.10, middle and left, resp.) as the activation function (I. Goodfellow et al., 2016). The former saturate over most of their domain, being only sensitive to an input near zero, which makes learning difficult through the phenomenon of vanishing gradients during backpropagation. Despite their initial widespread use, they are nowadays discouraged from being used, save in specific contexts (§ 2.4.2.5). The latter is usually easier to train, but also relies on the activation inputs being small.

A common modern default choice for an activation function is the rectified linear unit ([ReLU](#)), defined as:

$$\text{relu}(x) := \max(0, x). \quad (2.81)$$

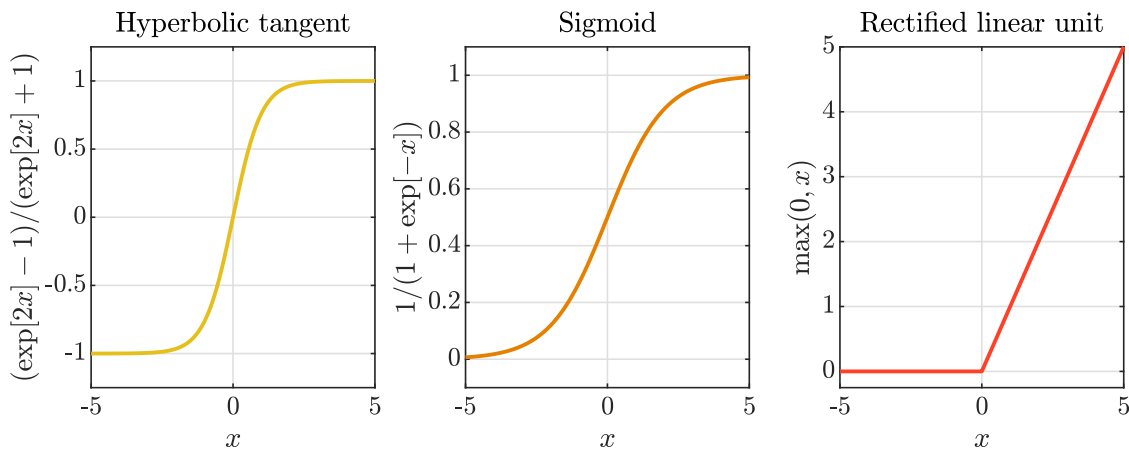


Figure 2.10: Typical artificial neural network (ANN) nonlinear activation functions. *Left:* $h(x) = \tanh(x)$. *Middle:* $h(x) = \text{sigm}(x)$. *Right:* $h(x) = \text{relu}(x)$.

The function is characterised by outputting a value of zero when the input is negative, and having a linear response otherwise (Figure 2.10, right), being for this reason easy to optimise. Other advantages include sparse activation (approximately 50% of the hidden units will have a non-zero output for a randomly initialised ANN, which is good to prevent overfitting and hence generalise well to new examples, see § 2.4.2.6) and better gradient propagation when compared to the sigmoid and hyperbolic tangent functions which saturate in both directions.

Remark 2.7: Dying ReLU Problem

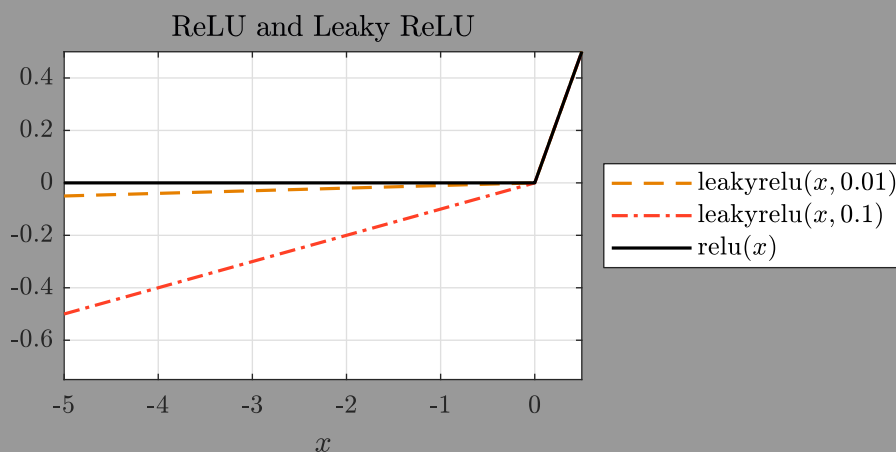


Figure 2.11: Comparison between the rectified linear unit (ReLU) activation function and the leaky ReLU activation function. The latter is plotted for two distinct values of η , the parameter defining the slope of the response when $x < 0$.

Despite its advantages, the ReLU activation can run into specific problems

during backpropagation, namely converging to states of inactivity regardless of the input. This is known as the dying ReLU problem and can occur when the network learns a large negative bias, driving the activations' output to zero (see Figure 2.10, right), effectively blocking the backward gradient flow. To mitigate this, some generalisations of the ReLU function have been introduced. The leaky ReLU (Maas et al., 2013), in particular, results in a small gradient when the unit is not active:

$$\text{leakyrelu}(x, \eta) := \begin{cases} x & \text{if } x > 0, \\ \eta x & \text{otherwise.} \end{cases} \quad (2.82)$$

The function is illustrated in Figure 2.11. Other approaches exist, such as the parametric ReLU, which treats η as a learnable parameter (He et al., 2015).

2.4.2.2 Optimisation

The beginning of Section 2.4.2 introduced SGD as the “default” learning method for DNNs. Despite its simplicity, however, it usually leads to a slow learning process (I. Goodfellow et al., 2016). To accelerate the optimisation, momentum can be used to enhance SGD by introducing an additive variable taking the role of velocity which defines the direction and speed of the motion through parametric space. The parameter update approach becomes (cf. Eqs. [2.77] and [2.79]):

$$\mathbf{v}^{(\kappa+1)} = \gamma \mathbf{v}^{(\kappa)} - \alpha \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^m f(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \boldsymbol{\theta}) \right), \quad (2.83)$$

$$\boldsymbol{\theta}^{(\kappa+1)} = \boldsymbol{\theta}^{(\kappa)} + \mathbf{v}^{(\kappa+1)}. \quad (2.84)$$

Equation (2.83) shows that SGD with momentum conserves the influence of past steps by keeping a moving average of the previous gradients, where $\gamma \in [0, 1[$ determines the rate of decay of their contributions. Momentum is useful for cases where the Hessian matrix is poorly condition, e.g. the topology of the objective function is highly varying in directions perpendicular to the path toward a minima, where regular SGD would be highly affected by such.

Momentum does not, however, solve the problem of choosing the arguably most important hyperparameter (i.e. the learning rate); in fact, it introduces an additional one. An alternative approach consists in designing optimisation strategies which

adapt the learning rates of model parameters. The RMSProp algorithm, unpublished and introduced by Geoffrey Hinton in 2012 during his lectures,⁶ modifies the learning rate of each individual parameter in θ by scaling them in inverse proportion to the accumulated squared gradient of past steps. The accumulation process follows an exponentially decaying average in order to neglect the influence from the distant past to ensure fast convergence after finding a convex bowl (I. Goodfellow et al., 2016). The update is given by:

$$\mathbf{g}_{\text{acc}}^{(\kappa+1)} = \gamma \mathbf{g}_{\text{acc}}^{(\kappa)} + (1 - \gamma) \mathbf{g}^{(\kappa+1)} \odot \mathbf{g}^{(\kappa+1)}, \quad (2.85)$$

$$\Delta \theta^{(\kappa+1)} = - \frac{\alpha}{\sqrt{\mathbf{g}_{\text{acc}}^{(\kappa+1)} + \epsilon}} \odot \mathbf{g}^{(\kappa+1)}, \quad (2.86)$$

$$\theta^{(\kappa+1)} = \theta^{(\kappa)} + \Delta \theta^{(\kappa+1)}, \quad (2.87)$$

where the operator \odot denotes element-wise product, ϵ is a small constant to stabilise division by potentially small numbers, the division and square root are applied element-wise, and \mathbf{g} denotes the gradient (continuing the notation introduced by Eq. [2.37]).

The Adam⁷ algorithm (Kingma and J. Ba, 2014) goes one step further adds another term, resembling momentum (cf. Eq. [2.83]) based on the non-squared gradient. It also includes an estimation of the bias corrections to both gradient-dependent terms. The full update is summarised as:

$$\mathbf{g}_{1\text{st}}^{(\kappa+1)} = \gamma_1 \mathbf{g}_{1\text{st}}^{(\kappa)} + (1 - \gamma_1) \mathbf{g}^{(\kappa+1)}, \quad (2.88)$$

$$\mathbf{g}_{2\text{nd}}^{(\kappa+1)} = \gamma_2 \mathbf{g}_{2\text{nd}}^{(\kappa)} + (1 - \gamma_2) \mathbf{g}^{(\kappa+1)} \odot \mathbf{g}^{(\kappa+1)}, \quad (2.89)$$

$$\hat{\mathbf{g}}_{1\text{st}}^{(\kappa+1)} = \frac{\mathbf{g}_{1\text{st}}^{(\kappa+1)}}{1 - \gamma_1^\kappa}, \quad (2.90)$$

$$\hat{\mathbf{g}}_{2\text{nd}}^{(\kappa+1)} = \frac{\mathbf{g}_{2\text{nd}}^{(\kappa+1)}}{1 - \gamma_2^\kappa}, \quad (2.91)$$

$$\Delta \theta^{(\kappa+1)} = -\alpha \frac{\hat{\mathbf{g}}_{1\text{st}}}{\sqrt{\hat{\mathbf{g}}_{2\text{nd}} + \epsilon}}, \quad (2.92)$$

where the divisions and square root are applied element-wise and the parameter update is identical to Equation (2.87). Note that during the bias update steps the decay parameters γ_1 and γ_2 are exponentiated by the current iteration value κ . In

⁶http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

⁷Not an acronym but a portmanteau of “adaptive” and “moment”.

this way, Adam features added robustness to the choice of hyperparameters when compared to **SGD** with momentum or RMSProp.

2.4.2.3 Convolutional Neural Networks

Classical **ML** strategies imply a pre-processing step to obtain the inputs to the training algorithm, which are seldom raw, but are instead the product of some feature detection and extraction algorithm that ultimately yields a set of observations $\mathbb{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$. This step is of crucial importance as it must be ensured that the chosen features are an adequate representation of the observed reality, and often this adequacy can be highly dependent on the application. For instance, an algorithm specialised in detecting interest points $\mathbf{z}^{(i)}$ on the image plane that will then be matched to solve a **PnP** problem (see § 2.1.4) or the on-manifold **LM** normal equations (see Eq. [2.63]) must be robust to the extreme variability experienced in on-orbit imaging due to sunlight hitting spacecraft's materials. Additionally, there is no guarantee that such algorithms, if developed for ground applications, will work just as well for space operations.

The end-to-end training capability of **DNNs**, on the other hand, represents a vast advantage for image-based applications as the optimal feature representation can be optimally learned in an unsupervised fashion, a step which is painstakingly present in classical **ML**. Autonomous vision-based spacecraft navigation, in particular, is one key area with the potential of largely benefiting from **DNN**-based estimation methods. Using such models would adequately capture the intrinsic nonlinearities between the input sensor data and the 6-DOF pose estimates. The hidden layer model inherent to the **MLP** depicted in Figure 2.9 has as a basis the linear combination between every possible input and output; each layer is, as such, alternatively termed a *fully connected* (**FC**) layer. Training an **FC** layer applied to an image input would be prohibitively expensive due to the sheer number of pixels encompassed in even modest resolutions by today's standards (each pixel would be an input to the **DNN**). Instead, *convolutional neural networks* (**CNNs**) are naturally tailored to process such image inputs (LeCun et al., 1989).

A convolution operation slides a 2D kernel \mathbf{K} (typically a relatively small, square matrix, with odd dimensions) across a 2D input image $\bar{\mathbf{I}}^{\text{in}}$ to produce a 2D output, $\bar{\mathbf{I}}^{\text{out}}$ according to the operation:

$$\bar{I}_{i,j}^{\text{out}} = (\bar{\mathbf{I}}^{\text{in}} * \mathbf{K})_{i,j} = \sum_u \sum_v \bar{I}_{i+u,j+v}^{\text{in}} K_{u,v}, \quad (2.93)$$

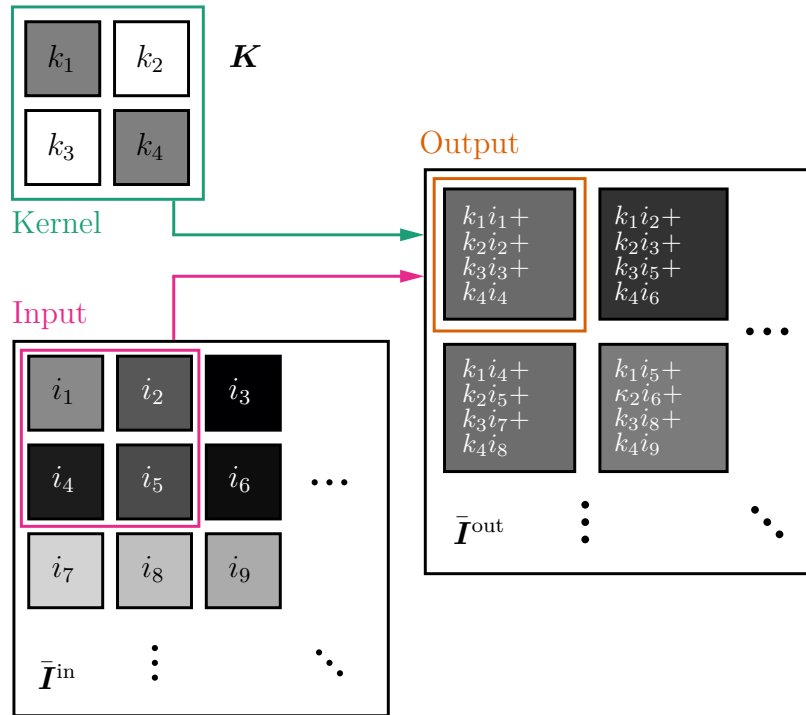


Figure 2.12: Illustration of a two-dimensional convolution operation. A single channel input image $\bar{\mathbf{I}}^{\text{in}}$ is convolved with a 2×2 kernel \mathbf{K} to produce a single channel output image $\bar{\mathbf{I}}^{\text{out}}$. The example is restricted to the top left 3×3 sub-matrix of $\bar{\mathbf{I}}^{\text{in}}$, where each input pixel has been abbreviated as $\{i_1, \dots, i_9\}$. Analogously, the kernel elements are denoted by $\{k_1, \dots, k_4\}$. The colour of each element represents the image intensity level, ranging from 0 (black) to 1 (white), which is also reflected on the output image, or feature map. Adapted from I. Goodfellow et al. (2016).

where index notation has been used.⁸ Figure 2.12 demonstrates the convolution operation and the expected output. The advantage of convolutional layers is undoubtedly the sparsity of learnable parameters, which dramatically decrease in number when compared to an FC layer, lowering memory requirements. This is accompanied inherently by a sharing trait, meaning that a single set of parameters is learned regardless of the input location, since the kernel is slid over the whole image. This has the same effect as looking for localised features in an image, where the resulting output works takes the role of a *feature map*, representing the location and intensity of the detected kernel features. Indeed, CNNs are frequently used as feature extraction front-ends, where spatial information is sequentially reduced while additional feature maps are generated. Here, the feature maps can be seen as images with multiple channels (the first two dimensions corresponding to spatial ones) resulting from the

⁸Equation (2.93) actually denotes the operation of cross-correlation, which is related to convolution through the flipping of the kernel with respect to the input. However, most ML libraries do implement convolution as such (I. Goodfellow et al., 2016), and hence this thesis shall follow the same convention.

convolution with an extension of the 2D kernel to a 4D tensor, \mathbf{K} , with dimensions given by:

$$\dim \mathbf{K} = C_{\text{in}} \times F \times F \times C_{\text{out}}, \quad (2.94)$$

where C_{in} is the number of input channels, C_{out} is the (desired) number of output channels, and it has been assumed that the kernel is spatially square and of dimensions $F \times F$. Spatial reduction can be achieved by applying a *pooling* operation after the convolution, where the feature map of spatial dimensions W_{in} is subdivided into bins to generate an output of dimension

$$W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} - Q}{Q} \right\rfloor + 1, \quad (2.95)$$

where Q is the pooling window size. Typical pooling operations take either the maximum or average value of each bin. Modern DNNs, however, have mostly abandoned pooling layers in favour of increasing the *stride*, S , of the convolution, i.e. the number of skipped rows and columns of the input when sliding the kernel. In this case, the spatial output size can be controlled according to the formula:

$$W_{\text{out}} = \frac{W_{\text{in}} - F + 2P}{S} + 1, \quad (2.96)$$

where P is the spatial padding applied to the input.

2.4.2.4 Transfer Learning

DNNs that translate into complex models may not only require large training times but also vast training datasets. A possible approach towards mitigating these two obstacles is *transfer learning*: the assumption that some factors responsible for influencing the outcome of one task are relevant to the outcome of a different task. Concretely, in the case of CNNs, it is expected that several of the learned kernels converge towards detecting generalised visual features. In practice, it has been verified experimentally in image classification tasks that the kernels of the first CNN layers⁹ are optimised towards wide-domain, broad features such as corners and edges, whereas the kernels of the last layers specialise in more problem-specific shapes (Zeiler and Fergus, 2013). Thus, it is common to adopt a pre-existing CNN architecture as the backbone of the model in which the early layers up to a point have been trained on a large, general dataset. Then, the few last layers (or new added ones) can be trained from the ground up using smaller domain-specific data.

⁹In the context of this thesis, the first layers of a DNN shall refer to the ones closest to the input.

A dataset commonly used in transfer learning for visual tasks is the ImageNet dataset, which consists of more than 14 million images hand-annotated into bins of more than 20 000 categories via crowd-sourcing. In the past decade, many advances in CNN architecture design came from participations in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC),¹⁰ in which the objective is to build a ML pipeline to correctly classify images into 1000 different classes of ImageNet. Notably, the trampoline jump of DNNs into public view can be traced back to Krizhevsky et al.’s (2012) performance on ILSVRC 2012, which it won with a top-5 classification error of 15.3%, more than almost 11 p.p. lower than the runner-up, using a variant of LeCun et al.’s (1989) own CNN trained on a GPU, which massively accelerated the process. Since then, new CNN designs have competed in the challenge each year, resulting in successive dramatic improvements while showcasing new advances, such as GoogLeNet (Szegedy, W. Liu, et al., 2015), which was characterised not only by a very deep architecture, but also by parallel layers to detect features at different scales; and ResNet (He et al., 2016), which introduced residual connections allowing the breakthrough to even deeper architectures. Most of these state-of-the-art DNN architectures have been open-sourced, with pre-trained ImageNet weights made available, contributing to the rapid advancement of the field and the adoption of such models as CNN front-ends.

2.4.2.5 Recurrent Neural Networks

The DNN architectures described so far include different, assumed unrelated, inputs at each forward pass. A different type of architecture, the *recurrent neural network* (RNN), explicitly attempts to model any existing temporal relationship between inputs at each time-step, containing loops that allow information to be passed from one to the next (see Figure 2.13). The basic RNN architecture learns an additional set of parameters to weigh the contribution of a previous output (Rumelhart et al., 1986):

$$a_j^{(\kappa+1)} = \sum_{i=1}^D W_{j,i}^{(x)} x_i + \sum_{k=1}^M W_{j,k}^{(z)} z_k^{(\kappa)}, \quad j = 1, \dots, M \quad (2.97)$$

$$z_j^{(\kappa+1)} = h\left(a_j^{(\kappa+1)}\right), \quad (2.98)$$

where the bias has been omitted for brevity (cf. Eq. [2.74]). This design is therefore adequate to model information organised in lists or sequences, like image captioning,

¹⁰<http://www.image-net.org/challenges/LSVRC>.

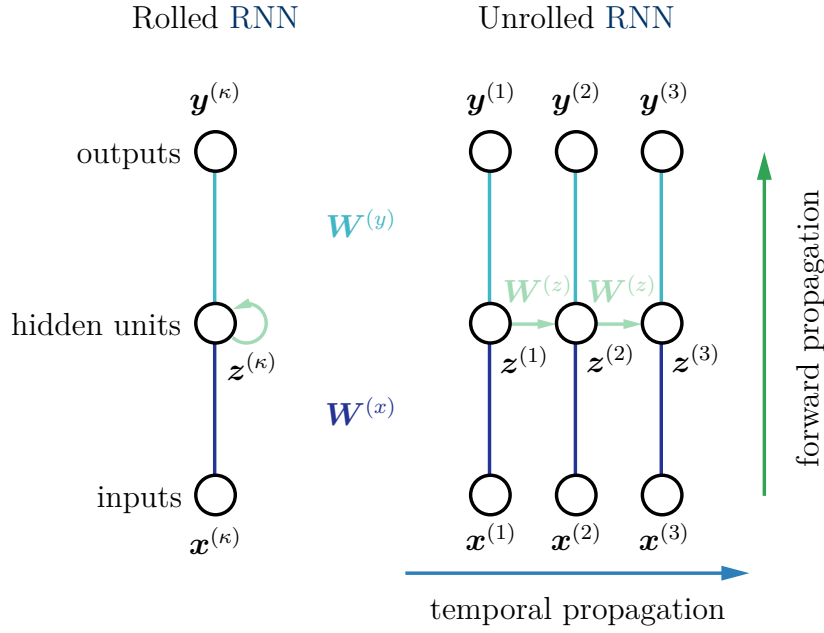


Figure 2.13: Diagram for a basic two-layer recurrent neural network (RNN). For this type of network, the training of the parameters at time-step $\tau = \tau_\kappa$ does not depend only on the current input $\mathbf{x}^{(\kappa)}$, but also on the previous output $\mathbf{z}^{(\kappa-1)}$. The unrolled representation of the network thus resembles a chain-like structure, and the gradient computation involves a forward propagation pass moving along a temporal axis. The network elements are represented with vector notation using single nodes. Cf. Figure 2.13.

natural language processing, and time series data such as a space rendezvous trajectory. The recurrent weights $\mathbf{W}^{(z)}$ are obtained via backpropagation through time (BPTT), which is similar to regular backpropagation to obtain $\mathbf{W}^{(x)}$, except that errors are summed at each time-step since the parameters are shared temporally. The gradient with respect to the recurrent weights for the model in Figure 2.13 becomes accordingly:

$$\frac{\partial f}{\partial \mathbf{W}^{(z)}} = \sum_{\kappa=1}^{T-1} \sum_{j=1}^{\kappa} \frac{\partial f_{\kappa}}{\partial y_{\kappa}} \frac{\partial y_{\kappa}}{\partial z_{\kappa}} \left(\prod_{k=j+1}^{\kappa} \frac{\partial z_k}{\partial z_{k-1}} \right) \frac{\partial z_j}{\partial \mathbf{W}^{(z)}}, \quad (2.99)$$

where T is the total number of time-steps and f_{κ} is the value of the cost function computed at time $\tau = \tau_{\kappa}$.

The issue with traditional RNNs comes from the product $\prod_{k=j+1}^{\kappa} \partial z_k / \partial z_{k-1}$ in Equation (2.99) as successive matrix multiplications can lead to either vanishing or exploding gradients, depending on the size of the gradient. In the former case, the network eventually stops learning; in the latter, the model becomes unstable and ends in numerical overflow. In practice, they also have problems in learning long-term dependencies (I. Goodfellow et al., 2016). Different approaches have been proposed

to modify the RNN and fix these issues, notably the *long short-term memory* (LSTM) unit (Hochreiter and Schmidhuber, 1997), which features a gating system to control a cell state (a form of “information motorway” that bypasses each unit save for some minute interactions). LSTMs are explored in Chapter 6.

2.4.2.6 Regularisation

An implicit hyperparameter of DNNs is the number of its learnable parameters, termed the capacity of the model. The larger the capacity, the better the potential of the DNN to predict with lower error. However, too many parameters can lead to *overfitting*, which occurs when the network actually stops learning and instead memorises the training data, becoming unable to generalise to new, unseen test data. For this reason, it is common during the training process to periodically assess the performance of the network on a validation set. A validation set is not used for training itself, but simply to provide insight on the pipeline’s capability for generalisation. In particular, when the validation error is much higher than the training error, it could signify that the DNN is overfitting. Conversely, too few parameters may lead to underfitting, characterised by both high training and validation errors.

To combat overfitting, apart from reducing the number of parameters, early stopping could be implemented, which consists in stopping the training process once the validation error starts to increase away from the training error. Additionally, *regularisation* procedures can be introduced, i.e. the addition of noise to the learning process. Generally speaking, some form of regularisation should always be present unless the training set contains a number of examples in the order of tens of millions (I. Goodfellow et al., 2016). This subsection summarises a few popular regularisation approaches for DNNs.

Weight Decay

Weight decay is one of the earliest forms of regularisation in machine learning, even predating ANNs. It consists in adding a term to the loss function that is proportional to each layer’s weights. A common type of weight decay is L^2 regularisation, which adds the squared sum of weights (typically, biases are ignored in weight decay). This results in the following modification to the gradient of the i -th layer (I. Goodfellow et al., 2016):

$$\nabla_{\mathbf{W}} f_i(\mathbf{W}^{(i)}, \mathbf{x}, \mathbf{y}) \leftarrow \lambda \mathbf{W}^{(i)} + \nabla_{\mathbf{W}} f_i(\mathbf{W}^{(i)}, \mathbf{x}, \mathbf{y}), \quad (2.100)$$

where λ is a hyperparameter typically chosen along a logarithmic scale, e.g. $\lambda \in$

$\{10^{-6}, 10^{-5}, \dots, 10^{-2}\}$. Despite the long-standing history of weight decay in the context of ML, for modern CNNs alternative strategies are typically employed additionally or as a complete alternative.

Dropout

Dropout (Hinton et al., 2012) consists in randomly removing units from a network by multiplying their output value by zero. Dropout attempts to emulate the ML concept of bagging (i.e. training K different models for K different training data subsets) for the case of very deep networks. For each minibatch step, a binary mask is randomly sampled to apply to the hidden units of each layer with probability p . This probability is fixed and treated as hyperparameter of each layer. Typical values are $p = 0.5$ for FC layers and $p \in [0.1, 0.2]$ for convolutional layers, placed before the activation function.

Batch Normalisation

Batchnorm (Ioffe and Szegedy, 2015) normalises a layer's inputs by calculating the mean $\mu_{\mathbb{B}}$ and variance $\sigma_{\mathbb{B}}^2$ over each mini-batch m as:

$$\mathbf{a}'^{(i)} \leftarrow \frac{\mathbf{a}^{(i)} - \mu_{\mathbb{B}_m}}{\sqrt{\sigma_{\mathbb{B}_m}^2 + \epsilon}}, \quad \mathbf{a}^{(i)} \leftarrow \mathbf{x}^{(i)} \in \mathbb{B}_m = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}, \quad (2.101)$$

where ϵ is a small numerically stabilising term. Furthermore, the outputs are modified by a learnable scale and offset:

$$\mathbf{a}''^{(i)} \leftarrow \gamma_{\mathbb{B}_m} \mathbf{a}'^{(i)} + \beta_{\mathbb{B}_m}. \quad (2.102)$$

Along the training procedure, the mean and variance of the full dataset is approximated by taking the moving average of each per-batch $\{\mu_{\mathbb{B}_m}, \sigma_{\mathbb{B}_m}^2\}$, which is then used to normalise inputs at inference time. Batchnorm was originally devised to improve DNN optimisation in general, but it introduces noise into the system that can have a regularising effect.

Image Augmentation

Image augmentation is an effective regulariser that consists in simultaneously generating more (augmented) data and indirectly teaching a DNN which are the important features to learn. In the case of CNNs, for instance, the performance of a classifier could be improved simply by randomly rotating the input image in-plane, thus improving the robustness towards rotation (e.g. a rotated satellite is still a satellite). Image augmentation is typically performed online by defining a random probability

for the application of some transformation to the input image before each training iteration.

2.5 On Datasets

Datasets are a central piece in the evaluation of an algorithm. A *dataset* is not a collection of separate, random information elements, but one of coherent ingredients that accurately represent the reality in which the method aims to be validated and tested. Better yet, an ideal dataset samples information directly from that same reality, if the given circumstances do allow it. Well-structured datasets further provide the possibility of programmatically assessing the performance of an algorithm in comparison to others under the same conditions (*benchmarking*). With the recently observed widespread adoption of machine learning techniques in both academia and industry, particularly deep learning networks which feed directly on raw data as part of the training procedure, the importance of accessing meaningful and ample datasets is ever increasingly paramount. Furthermore, the data should ideally be labelled, thus conferring on it the *ground truth* from which the algorithm can simultaneously learn to classify or regress and be evaluated.

In the context of space operations, labelled relative pose estimation datasets are scarce or non-existent. Real data from actual rendezvous missions are habitually the product of development spanning many years and extensive funding, and hence are held internally. Space agencies funded by taxpayers, on the other hand, typically do release image sets acquired by the on-board scientific payload into the public domain, but these are often sparse and unlabelled, not meant for six degrees-of-freedom (6-DOF) pose estimation. Additionally, datasets must be acquired prior to the actual mission as the whole guidance, navigation and control (GNC) subsystem is subject to extensive testing campaigns before being deemed fit for flight. As such, research efforts have turned towards the conception of *synthetic* datasets instead, i.e. images generated by computer software that aim to recreate the conditions observed in space. Generating a synthetic dataset involves a substantial initial investment in the development of a realistic *camera simulator* engine that can accurately model not only the imager's physical properties but also the imaging conditions experienced in space from the standpoint of shadows, planetary atmospheres, Sun and ambient lighting, reflections, light glare, among others. Camera simulators must be capable of faithfully replicating the positioning and relative motion of the celestial bodies in the solar system, particularly the Sun and Earth. Other assets, such as spacecraft, can be included through 3D *computer-aided design* (CAD) models which are capable of emulating complex material properties and textures.

2.5.1 Open-Source Datasets for Motion Estimation

The present section describes a number of publicly available computer vision datasets, real and synthetic, with focus on motion estimation.

2.5.1.1 Autonomous Vehicles and Mobile Robotics

The KITTI dataset (Geiger et al., 2013) remarkably features six hours' worth of real traffic data recorded aboard a car driving around Karlsruhe, Germany. The used sensor suite consists of 2 greyscale cameras, 2 red-green-blue (RGB) cameras, 1 rotating 3D laser scanner, and 1 inertial and Global Positioning System (GPS) six-axis navigation system. Each driving sequence contains the raw data, object annotations in terms of 3D bounding boxes and calibration files, placing the total dataset size at 180 GB. The KITTI dataset has been used in an extensive collection of applications such as visual odometry (VO; Boulekhour and Aouf, 2014), scene segmentation (Wieszok et al., 2017), and sensor fusion (Courtois and Aouf, 2017). With the recent growth in autonomous driving technology for road vehicles, car manufacturers themselves such as Audi (Geyer et al., 2020) and Ford (Agarwal et al., 2020) are also contributing towards open-access research by releasing their own datasets. On the other hand, TU Munich's Monocular Visual Odometry Dataset (Engel, Usenko, et al., 2016) contains 50 real-world sequences amounting to over 100 min of hand-held video captured across different environments that span from indoor spaces to wide outdoor scenes; the dataset enables the benchmarking of the tracking accuracy of monocular VO and simultaneous localisation and mapping (SLAM) methods. For further reading, Taketomi et al. (2017, §8) provide a synoptic survey of datasets used for VO/SLAM.

2.5.1.2 Space Rendezvous

The first open-source image collection for spacecraft relative pose estimation was the Spacecraft PosE Estimation Dataset (SPEED), which was used to benchmark the entries of the 2019 European Space Agency (ESA) Satellite Pose Estimation Challenge (SPEC; Kisantal et al., 2020). Overall, SPEED is a challenging dataset based on images taken of the Mango spacecraft as viewed by Tango, both of which flew in the Hyperspectral Precursor of the Application Mission (PRISMA); it is composed of both synthetic and laboratory-acquired greyscale data divided into train (SPEED/TRAIN, SPEED/REAL) and test (SPEED/TEST, SPEED/REAL-TEST) sets, numbering 12 000, 5, 2998, and 300 images, respectively. Figure 2.14 displays some sample images of the dataset, taken from the SPEED/TRAIN set. Contrary to the train sets, the ground truth pose for the test ones used for the challenge is not made

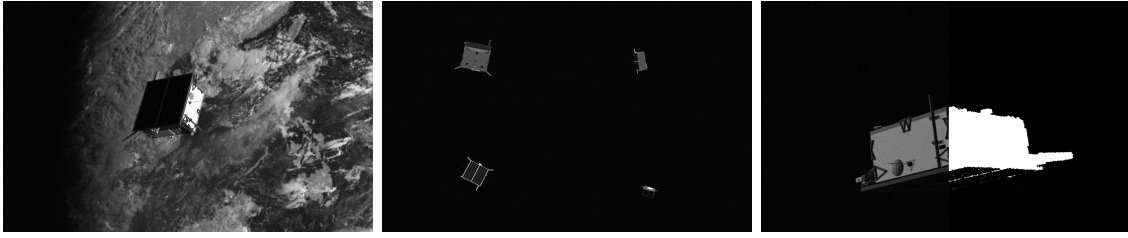


Figure 2.14: Characteristics of the Spacecraft Pose Estimation Dataset (SPEED; Kisantal et al., 2020). (Left) Earth in the background. (Centre) Ambiguous shape. (Right) Low lighting and noise, affecting segmentation.

publicly available, and the only way to acquire a performance metric is by submitting the results on the SPEC website.¹¹ At the time of this thesis’ writing, the website featured a post-mortem version allowing the submission of results despite the ending of the challenge.

Right after SPEC ended, Proença and Gao (2019), who achieved third place in the competition, released their own Unreal Rendered Spacecraft On-Orbit (URSO) dataset, featuring synthetic images of Earth-orbiting spacecraft using Unreal Engine 4. Three subsets — one with SpaceX’s Dragon and two with the Russian Soyuz — are provided, each containing 5000 high-resolution RGB images.

2.5.2 Simulation of Multimodal Trajectories for Space Rendezvous

The SPEED and URSO datasets do represent contributions in driving the task of image-based spacecraft pose estimation towards the community and collaboration facet of modern computer vision research. However, they are lacking in three distinct fronts: 1) the data consist of images of the targets generated at random poses, when spacecraft rendezvous is a continuous, time-correlated operation; 2) images are only supplied for the sunlit portion of the orbit; and 3) data are only provided for the visible modality.

In order to benchmark the algorithms developed in this thesis, a novel multimodal image-based dataset for relative navigation which fulfils all three above-mentioned points is developed. The generated images simulate a rendezvous approach with Envisat, capturing realistic variations as expected from a real space scenario, i.e. illumination, tumbling, and scale. The development of this dataset is the product of a collaboration with ESA (Dubois-Matra, 2016), who have provided access to the Astos Camera Simulator, a software capable of generating photo-realistic images based on 3D graphics. For this reason, the developed images are dubbed the ASTOS

¹¹<https://kelvins.esa.int/satellite-pose-estimation-challenge>.



Figure 2.15: Characteristics of the ASTOS dataset. (*Top Row*) A continuous sequence of red-green-blue (RGB) images in the visible wavelength of the target Envisat as imaged by a chaser's on-board camera. (*Second Row*) Earth contaminating the background. (*Middle Row*) Imaging during eclipse periods. (*Fourth Row*) Relative proximity changes. (*Bottom Row*) Imaging in long-wavelength infrared (LWIR).

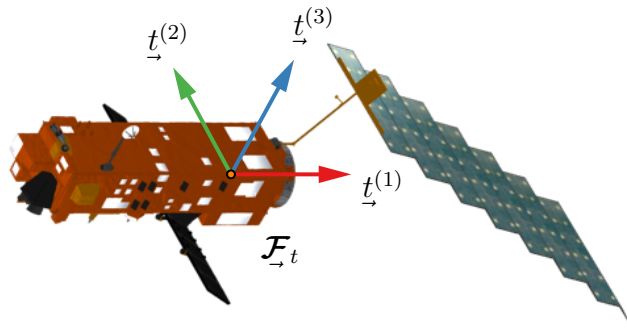


Figure 2.16: Envisat body reference frame.

dataset. Figure 2.15 illustrates some sample frames from the dataset. The following subsections detail the characteristics, tools, and methodology followed for the creation of the dataset.

2.5.2.1 Mission Scenario and Target Specifications

The ASTOS dataset simulates a rendezvous trajectory with Envisat (Louet and Bruzzi, 1999). Envisat is an Earth-observation satellite launched on 1st March 2002 by ESA with the objective of enhancing Europe’s remote sensing capabilities from space and being the successor of the European Remote Sensing 1 and 2 programmes launched in the 1990s. It became non-functional in 9th May 2012.

Envisat is characterised by its large weight (over 8000 kg) and size; it is composed of two main components: a bus measuring $2.750\text{ m} \times 1.600\text{ m} \times 10.020\text{ m}$ that contains the instruments, and a solar array measuring $4.972\text{ m} \times 0.01\text{ m} \times 14.028\text{ m}$. The bus itself is subdivided into a service model which provides the satellite with its basic functions (power, attitude and orbit control, communications, etc.), and a payload module which is dedicated to housing the payloads and payload-dedicated support systems. At the time of launch, Envisat was the largest free-flying object. Currently, it poses a collision risk in low Earth orbit (LEO) and is the potential target for the first active debris removal (ADR) mission to be carried out by ESA, e.Deorbit (Biesbroek, Innocenti, et al., 2017).

Envisat Reference Frame Definition

Figure 2.16 depicts Envisat and a superposition of its body reference frame (Dubois-Matra, 2016). It is a right-handed axis system with:

- The origin coincident with the service module’s centroid.
- The $t^{(1)}$ axis aligned along the launch vehicle axis.

Table 2.1: Envisat set of orbital elements at time $\tau = \tau_0$ for the ASTOS dataset generation.

Element	Dimensions	Symbol	Value
Eccentricity	--	e	7.6112×10^{-4}
Semimajor axis	km	a	7.1427×10^3
Inclination	deg	i	98.2156
Right ascension of the ascending node	deg	Ω	343.0760
Argument of perigee	deg	ϖ	189.5264
True anomaly	deg	θ	3.0109

- The $\underline{t}^{(2)}$ axis aligned along the synthetic aperture radar (SAR).
- The $\underline{t}^{(3)}$ axis completing the right-handed triad.

When the satellite was still operational, the $\underline{t}^{(1)}$ axis was closely aligned to the orbit normal, the negative $\underline{t}^{(2)}$ axis was closely aligned to Envisat’s velocity vector, and the negative $\underline{t}^{(3)}$ axis was closely aligned to the downward local normal (nadir).

Envisat Orbit

The satellite’s orbit is derived from the two-line element (TLE) data corresponding to its state on 30th October 2017, which corresponds to the beginning of the collaboration to create the dataset. The TLEs were obtained from the publicly accessible North American Aerospace Defense Command (NORAD) website¹² and converted to the orbital elements displayed in Table 2.1

Remark 2.8: Orbital Elements

The Cartesian position, \mathbf{r} , and velocity, \mathbf{v} , are convenient to use in computations, but do not necessarily provide an intuitive understanding of what characterises an orbit (Wertz, 2001). Six parameters are required to fully define the spacecraft’s three-dimensional position in orbit: two for the orbit size and shape, two for the orientation of the orbital plane in space, one for the orientation of the orbit within the plane, and one for the planar-constrained displacement of the spacecraft. These are represented in Figure 2.17.

¹²<http://www.celestrak.com/NORAD/elements>.

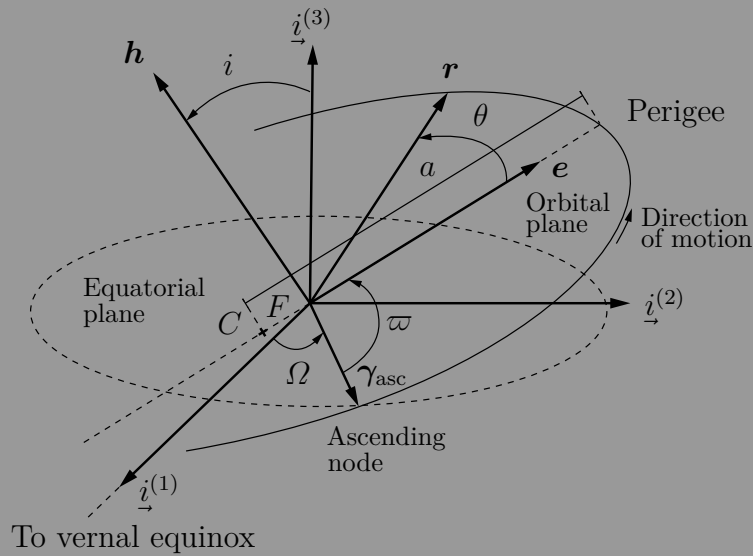


Figure 2.17: Classical orbital elements. The orbit is an ellipse with center C with Earth at the focus F . The elements $a, e, i, \Omega, \varpi, \theta$ provide the physical characterization of the orbit. Adapted from Rondao (2016).

The size and shape of a Keplerian orbit are defined by the semi-major axis, a , and the eccentricity, e . The inclination, i , is the angle between the orbital plane and a given reference plane (the equatorial plane for Earth-orbiting satellites). The line of nodes is the intersection of the orbital and equatorial planes and the ascending node designates the point on the line of nodes where the orbit crosses the equatorial plane from the south to the north. The vector γ_{asc} defines the position of the ascending node with respect to Earth's centre of mass. The second DOF to orient the orbital plane is defined by the right ascension of the ascending node, Ω , which is the angle between γ_{asc} and the vernal equinox.

The orientation of the orbit within the equatorial plane is specified through the argument of perigee, ϖ , which is the angle between γ_{asc} and the eccentricity vector (vector defining the direction of perigee), \mathbf{e} , measured in the direction of the spacecraft's motion.

Lastly, the true anomaly, θ , measures the angle between \mathbf{r} and the direction of perigee, \mathbf{e} , in the direction of motion, and thus defines the position of the spacecraft within the orbit.

Table 2.2: The three rotation scenarios of the target Envisat considered in the ASTOS dataset, defined in terms of the target body frame \mathcal{F}_t and the target local-vertical-local-horizontal (LVLH) reference frame \mathcal{F}_o (Dubois-Matra, 2016).

Scenario	Spin axis in \mathcal{F}_t	Spin axis in \mathcal{F}_o	Spin rate [deg s ⁻¹]
1	Aligned with the positive $\underline{t}^{(2)}$ axis	Aligned with the negative $\underline{o}^{(2)}$ (positive H-bar) axis	3.5
2	Along a direction contained in the $\underline{t}^{(2)}\underline{t}^{(3)}$ plane at 45 deg w.r.t. ^a the positive $\underline{t}^{(2)}$ and $\underline{t}^{(3)}$ axes	Aligned with the negative $\underline{o}^{(2)}$ (positive H-bar) axis	5
3	Aligned with the positive $\underline{t}^{(3)}$ axis	At an angle of 45 deg w.r.t. ^a the negative $\underline{o}^{(2)}$ (positive H-bar) axis and is fixed in an inertial frame	5

^a With respect to

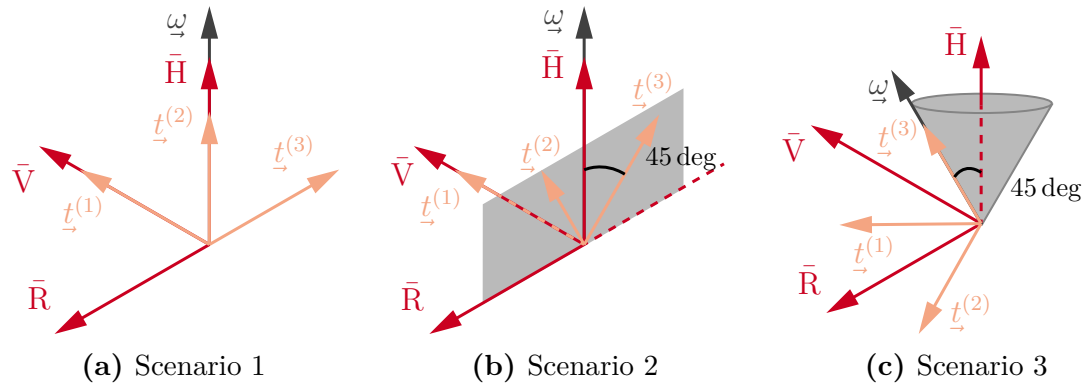


Figure 2.18: Schematic illustration of the three rotation scenarios of the target Envisat considered in the ASTOS dataset. Entities are colour-coded. (Red) The target local-vertical-local-horizontal (LVLH) reference frame \mathcal{F}_o . (Apricot) The target body-fixed reference frame \mathcal{F}_t . (Dark grey) The spin axis $\underline{\omega} = \underline{\omega}_{ot}$ of \mathcal{F}_t with respect to \mathcal{F}_o .

Envisat Rotational State Scenarios

Three different scenarios are considered for Envisat's rotational state; these are defined in Table 2.2 (Dubois-Matra, 2016). An illustration of these states is depicted in Figure 2.18. A note is made relative to Scenario 3, in which the spin axis $\underline{\omega} = \underline{\omega}_{ot}$ in \mathcal{F}_o is configured at a 45 deg angle with H-bar but is simultaneously fixed in the ECI frame \mathcal{F}_i . Since Envisat's orbit is approximately circular ($e \approx 0$, see Table 2.1), the inertial stabilisation of $\underline{\omega}$ draws out a cone when seen from the point of view of \mathcal{F}_o (see Figure 2.18c). In other words, the spin demonstrates an axial precession with a period equal to the orbital period. Using the orbital parameters from Table 2.1 and Kepler's third law (Wertz, 2001), this quantity can be calculated as:

$$P_{\text{orb}} = 2\pi \sqrt{\frac{a^3}{\mu_{\text{Earth}}}} \quad (2.103)$$

$$\approx 1.67 \text{ hours,}$$

where $\mu_{\text{Earth}} = 398\,600.5 \text{ km}^3 \text{ s}^{-2}$ is Earth's gravitational constant (Boden, 1999). This is equivalent to a precession rate of $3.59 \text{ deg min}^{-1}$.

Active Debris Removal Mission Definition

The ADR phase considered for the purpose of the study is the rendezvous and forced translation phase, defined as the segment where the chaser performs a rendezvous with the target object, evaluates its attitude dynamics and centre of mass position and if required performs a forced translation in order to reduce the relative motion between chaser and target to levels adequate prior to the capture (Dubois-Matra, 2016).

In order to characterise this phase, three different chaser guidance profiles are established in open-loop (Dubanchet, 2017):

- (1) Forced translation along an axis of the target \mathcal{F}_o frame with cross-track disturbances of 1 m amplitude. The trajectory begins at a relative distance of 100 m and ends at a distance of 50 m. The axes considered are the target V-bar and R-bar. The total trajectory length is 125 s.
- (2) Observation from a hold point on an axis of the target \mathcal{F}_o frame. The relative distance is constant and equal to 50 m. The axes considered are the target V-bar and R-bar. The trajectory last for 3 full target rotations: 309 s for R1; 216 s for R2 and R3.
- (3) Ellipse of inspection constrained to the V-bar/R-bar plane of the target \mathcal{F}_o

frame. Both semi-major axes are equal to 50 m. The total trajectory length is 200 s.

Figure 2.19 illustrates the guidance profiles in terms of the resulting trajectory shapes, and the position and velocity over time. A V-bar approach is exemplified in the case of Profiles 1 and 2.

2.5.2.2 Synthetic Image Generation

Issue 1.4 (14th August 2014) of the Astos Camera Simulator¹³ was used to generate the dataset. The 3D visible and thermal models of the targets are input as Wavefront .obj and .mtl files, along with a text file containing the 6D pose of the chaser and target at each time-step, specified either in the inertial or relative frames. The simulator is also capable of automatically propagating the objects' trajectories in space given the initial orbital elements; however, these have been generated manually for better control (see Fig. 2.19). A separate configuration file is also supplied, specifying the Julian date for the start of the simulation, the frames of references used, the camera parameters, and the graphical settings (reflections, light glare, shadows, etc.). The frames are then rendered as imaged by the synthetic cameras with the placement of Earth, Sun, and Moon defined from their true ephemeris and the input date.

Object Modelling

The orbital states of both chaser and target, the camera parameters, and a 3D CAD model of Envisat are used as inputs to the Astos Camera Simulator to generate the dataset. The original textured model was obtained from the free astronomy software Celestia¹⁴, a program which allows for the real-time 3D visualisation of space.

Visible Model The original Envisat model was heavily modified to guarantee a realistic simulation in the visible spectrum. This included re-meshing the main body of the spacecraft to emulate a “crumpled” effect for the multi-layer insulation (MLI) in order to properly emulate the diffuse reflection of light, as well as adding reflective properties to the solar panel; the differences between both models can be seen in Figure 2.20. Additionally, an image of a laboratory mock-up of Envisat from Cranfield University's Unmanned Autonomous Systems Laboratory (UASL) is also included for a qualitative comparison.

¹³<http://www.astos.de>.

¹⁴<http://celestia.space>.

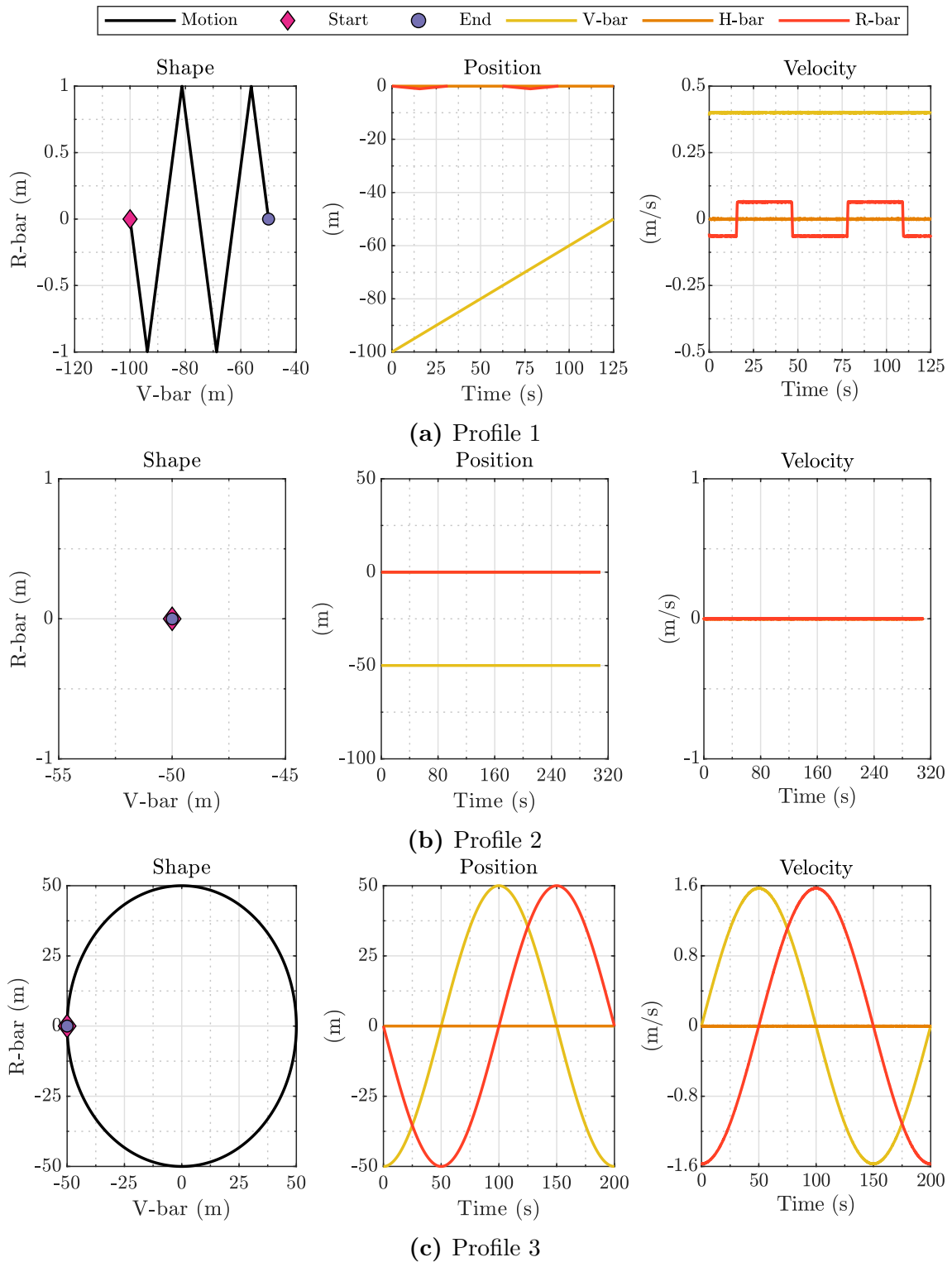


Figure 2.19: Schematic illustration of the three guidance profiles of the chaser rendezvous with the target Envisat considered in the ASTOS dataset. Quantities are resolved in the target’s local-vertical-local-horizontal (LVLH) reference frame \mathcal{F}_o . Profiles 1 and 2 are shown for the V-bar approach; an R-bar approach has also been generated.

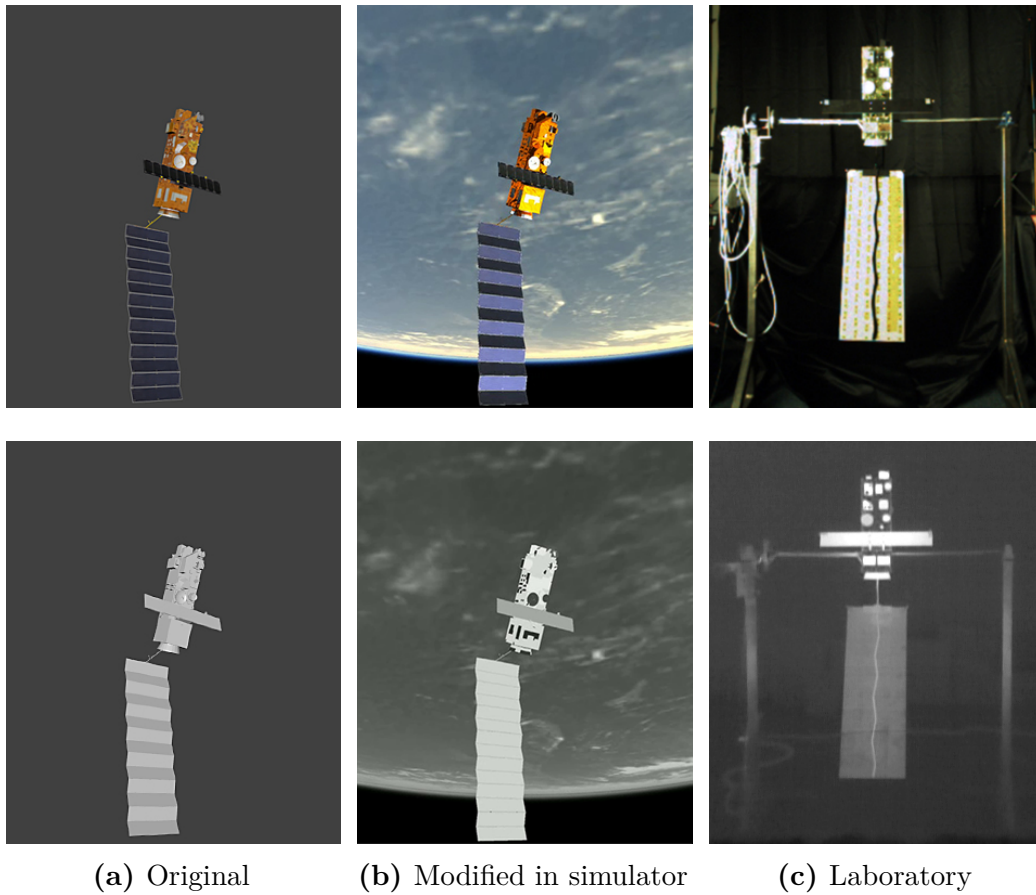


Figure 2.20: Multimodal computer-aided design (CAD) models of Envisat for use in the ASTOS dataset and qualitative comparison with laboratory data acquired from the Unmanned Autonomous Systems Laboratory (UASL). (*Top*) Visible wavelength. (*Bottom*) Long-wavelength infrared (LWIR). Laboratory thermal image reproduced from Yilmaz (2018) with permission.

Thermal Model The creation of thermal spacecraft model involved a different procedure. As part of Yilmaz’s (2018) doctoral project, a thermal testing campaign was performed at ESA using a scaled-down replica of Envisat with surface coatings of similar types to those used in the real spacecraft, from which the temperature and emissivity of each component were obtained. Two steady-state profiles were determined: one for the sunlit period of Envisat’s orbit, and another for the eclipsed period; these profiles have been shared by the author for the purpose of generating the synthetic dataset (Yilmaz, 2017).

The CAD model was then stripped of texture (see Fig. 2.20a) and the collected data was incorporated as follows. First, the in-band radiance of each component was calculated by integrating the spectral radiance, given by Planck’s law:

$$L_c(\lambda, T_c, \epsilon_c) = \epsilon_c \int_{\lambda-\delta}^{\lambda+\delta} \frac{2hc^2}{(\lambda')^5} \frac{1}{\exp\left(\frac{hc}{\lambda'k_B T_c}\right) - 1} d\lambda', \quad (2.104)$$

where λ is the sampled wavelength; T_c, ϵ_c are the component's temperature and emissivity, respectively; k_B is the Boltzmann constant; h is the Planck constant; c is the speed of light in vacuum; and δ is a small neighbourhood around λ . The LWIR band was sampled at $\lambda = (8, 11 \text{ and } 14) \mu\text{m}$. Equation (2.104) can be integrated as a series; see Widger and Woodall (1976) for details. Then, the computed radiances were normalised to $[0, 1]$ according to the modelled thermal camera's scene temperature range, yielding a 3-tuple analogous to the RGB values in the visible. Due to this normalisation step, the obtained values become insensitive to the choice of δ . It was found that for the given band and range of temperatures, the solution was stable for any $\delta < 1 \mu\text{m}$. Each 3-tuple was logged in an `.mt1` material file to accompany the `.obj` mesh file as inputs to the camera simulator.

Finally, the spectral response coefficients $\{\gamma_{\lambda_1}, \gamma_{\lambda_2}, \gamma_{\lambda_3}\}$ of the camera for each of the three sampled wavelengths are also added as inputs; the software then rendered the single-channel thermal images with intensity equal to

$$I_c = \gamma_{\lambda_1} L(\lambda_1) + \gamma_{\lambda_2} L(\lambda_2) + \gamma_{\lambda_3} L(\lambda_3). \quad (2.105)$$

The spectral response coefficients are obtained directly from the emulated thermal camera's datasheet.

The generation of synthetic thermal data according to the aforementioned process entails some approximations, which are a reflection of the limitations of the software. Concretely, a fixed thermal signature for each sequence and the assignment of a solid colour to each component, instead of gradients, are assumed. Nonetheless, this does not affect the validity of the generated dataset. The first approximation is justified by the fact that the dataset considers short duration sequences in thermal steady-state. The longest trajectory, for example, spans only approximately 5 min, which represents only about 5.1% of Envisat's complete orbital period. The second approximation is upheld based on the relatively large distances between chaser and target. Figure 2.20 also features a synthetic image rendered by the Astos Camera Simulator on the LWIR alongside a real thermal image of the mock-up as captured by a thermal infrared camera in the laboratory. Despite the different thermal signatures, it can be seen that the representation of the target in both images is comparable.

Table 2.3: Technical data – mvBlueFOX-MLC 202b



Parameter	Dimensions	Value
Resolution	px	1280 × 960
Frame rate	Hz	10
Focal length	mm	5
Sensor width	mm	4.8
Sensor height	mm	3.6

Table 2.4: Technical data – FLIR Tau2



Parameter	Dimensions	Value
Resolution	px	1280 × 1024
Frame rate	Hz	10
Focal length	mm	13
Sensor width	mm	10.875
Sensor height	mm	8.7
Spectral band	μm	8–14
Scene range	°C	−40–160

Camera Modelling

Two cameras were reproduced within the Astos Camera Simulator software: 1) one operating on the visible wavelength (0.39–0.70 μm), based on the mvBlueFOX-MLC 202b¹⁵ camera, and 2) one operating on the LWIR wavelength (8–14 μm) based on the FLIR Tau2¹⁶ camera with a scene temperature range from -40°C to 160°C .

Tables 2.3 and 2.4 summarise the properties of the emulated visible and LWIR cameras, respectively. The choices were motivated by the fact that both have a similar FOV. A frame rate of 10 Hz was fixed for both.

Dataset Key

In order to generate the complete dataset, the multiple guidance profiles and rotational scenarios are combined, along with the LVLH approach vector, illumination condition, and imaging modality. A total of 56 trajectories were synthesised. Figures (2.21) to (2.23) illustrate the key arrays for each of the trajectories.

¹⁵<https://www.matrix-vision.com/USB2.0-single-board-camera-mvbluefox-mlc.html>.

¹⁶<https://www.flir.co.uk/products/tau-2>.

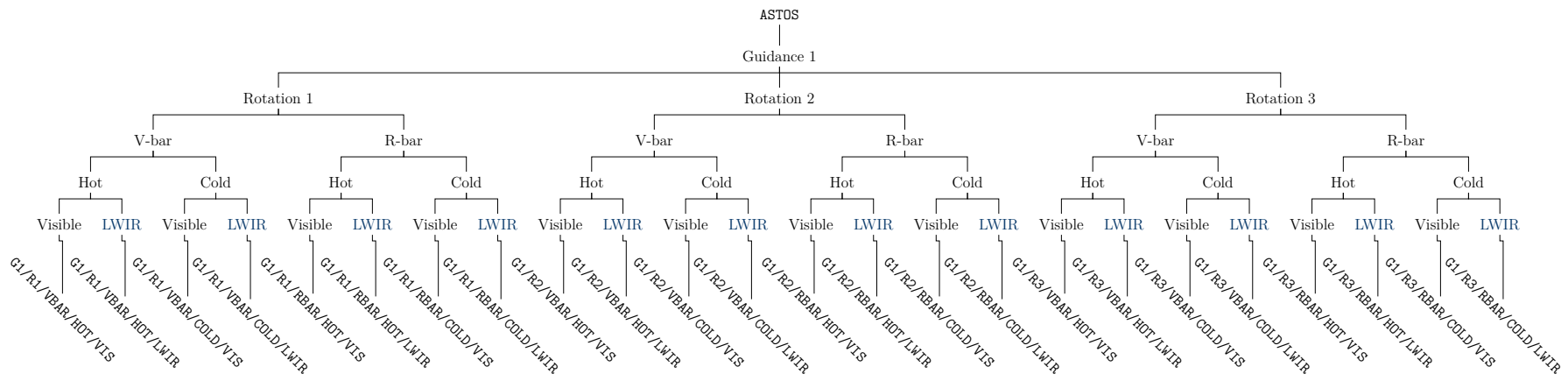


Figure 2.21: Key for the generated trajectories of the **ASTOS** dataset for chaser Guidance Profile 1. Consult Figure 2.19 for an illustration of the considered profiles. “Rotation” refers to the given target rotational state scenario (Fig. 2.18). “Hot” refers to a sunlit trajectory. “Cold” refers to an eclipsed trajectory.

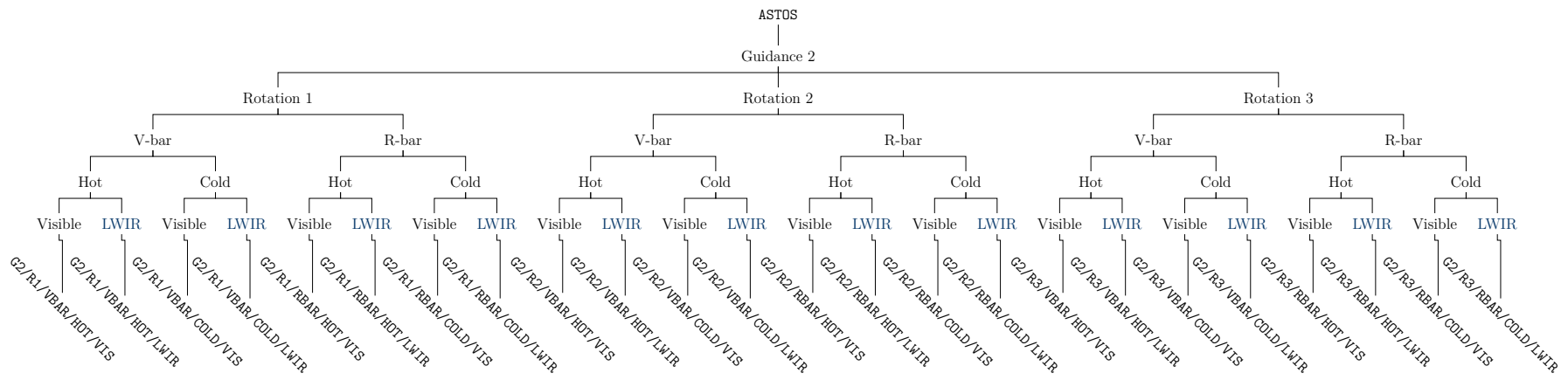


Figure 2.22: Key for the generated trajectories of the *ASTOS* dataset for chaser Guidance Profile 2. Consult Figure 2.19 for an illustration of the considered profiles. “Rotation” refers to the given target rotational state scenario (Fig. 2.18). “Hot” refers to a sunlit trajectory. “Cold” refers to an eclipsed trajectory.

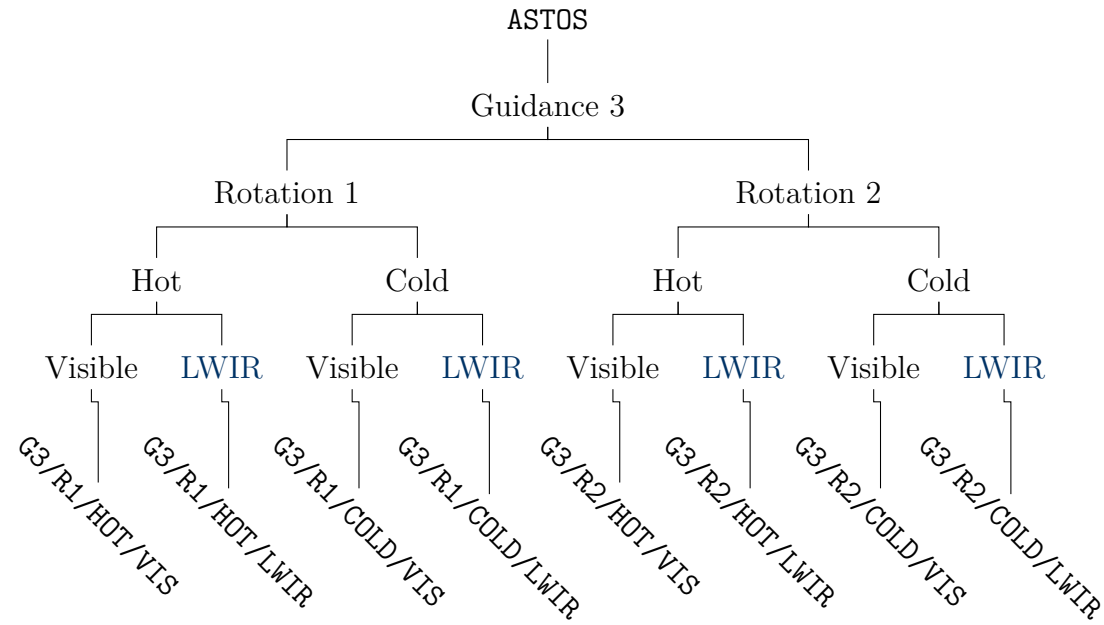


Figure 2.23: Key for the generated trajectories of the ASTOS dataset for chaser Guidance Profile 3. Consult Figure 2.19 for an illustration of the considered profiles. “Rotation” refers to the given target rotational state scenario (Fig. 2.18). “Hot” refers to a sunlit trajectory. “Cold” refers to an eclipsed trajectory.

CHAPTER 3

Benchmarking of Detectors and Descriptors for Navigation

*This chapter establishes a framework for the evaluation of low-level image processing algorithms prior to the inclusion in high-level relative navigation algorithms for space rendezvous. Twelve state-of-the-art point feature detectors and descriptors are analysed in terms of common metrics from the classical computer vision literature. A low resolution derivative of the *ASTOS* dataset that maintains the depiction of expected image transformations (illumination, rotation, and scale) while allowing a straightforward ground truth definition is introduced. Furthermore, each method's implementation is assessed on a embedded platform with reduced computing capabilities.*

3.1 Motivation

THE increasing number of space objects orbiting Earth is a jeopardising factor for current and future missions. The large proportion of bodies classified as debris, which has now been estimated at more than 83% (Andrenucci et al., 2011), not only leads to hardship in mitigation from the point of view of detection and avoidance, but also in terms of proliferation. Large, defunct spacecraft in low Earth orbit (LEO) such as Envisat represent de facto ticking time bombs waiting for an inexorable collision guaranteed to catalyse the problem. The existence of such bodies justifies the implementation of swift and efficient remissive plans of action such as active debris removal (ADR).

The e.Deorbit mission is set out to be the first ADR mission to be carried out by the European Space Agency (ESA), demonstrating the removal of a large object from its current orbit and performing a controlled re-entry into the atmosphere.

As one of the few ESA-owned debris in LEO, Envisat is a possible target for the mission (Biesbroek, Innocenti, et al., 2017). The mission is part of ESA's Clean Space Initiative, which is focused on outlining the required technology for this domain, including advanced image processing (IP) for the relative navigation aspect of the rendezvous operations. A smaller scale in-orbit demonstration mission using CubeSats to test IP algorithms, e.Inspector, has been proposed to visually inspect Envisat for the determination of its tumbling rate and axis. This data would then be used for validation purposes in e.Deorbit (Biesbroek, Wolahan, et al., 2017).

Using low-power-low-cost camera-based systems, two-dimensional features of the target image can be identified and extracted to yield a relative navigation solution (Chap. 2). As the space environment may prove hostile to solutions in the visible wavelength due to illumination, approaches to ADR in other spectra have been proposed, such as the long-wavelength infrared (LWIR), or thermal infrared (Yilmaz, Aouf, Checa, et al., 2017). Although studies comparing the general performance of IP algorithms in the visible and in LWIR are present separately in the literature, benchmarks performed in a space non-cooperative rendezvous (NCRV) context are scarce. Furthermore, to the best of the author's knowledge, no LWIR IP comparisons for NCRV were found to exist. Therefore, the purpose of this chapter is to benchmark the performance of IP techniques adjusted towards multimodal camera setups that could be inserted in ADR missions using affordable, low performance computing.

Remark 3.1: Associated Publications

This chapter is based partly on the following published work:

[C2] D. Rondao, N. Aouf, and O. Dubois-Matra (Oct. 2018). "Multispectral Image Processing for Navigation Using Low Performance Computing". In: *69th International Astronautical Congress (IAC) 2018*. Bremen, Germany: IAF. URL: <https://dspace.lib.cranfield.ac.uk/handle/1826/13558>

[J1] D. Rondao, N. Aouf, M. A. Richardson, and O. Dubois-Matra (July 2020). "Benchmarking of local feature detectors and descriptors for multispectral relative navigation in space". In: *Acta Astronautica* 172, pp. 100–122. DOI: 10.1016/j.actaastro.2020.03.049

3.2 Related Work

The evaluation of feature detectors goes back to before the turn of the century, when interest points were reduced to any point in an image for which the signal changed two-dimensionally, encompassing the traditional “L-corners”, “T-junctions”, and “Y-junctions”; a small image patch (the template) around the detected corner would then be extracted and matched for in the target image using correlation (Schmid et al., 2000). By then, however, there was not yet a clear consensus on how a proper evaluation framework should be set up. In fact, some authors resorted to subjective visual inspection methods to evaluate the quality of detection (e.g. López et al., 1999).

A few years later, with the advent of algorithms capable of detecting invariant features, such as Scale Invariant Feature Transform (SIFT; Lowe, 2004), criteria such as repeatability and matching scores became commonplace in evaluative frameworks. These algorithms would automatically extract a support region around the feature and encode it into a numerical descriptor, allowing it to be matched without searching the whole image. Arguably, the most well-known examples in the computer vision literature are the studies by Mikolajczyk, Tuytelaars, et al. (2005) on detectors and Mikolajczyk and Schmid (2005) on descriptors. This change in paradigm potentiated new developments in visual simultaneous localisation and mapping (VSLAM); hence, the contemporary studies included benchmarks regarding transformations that one would expect to experience in that context, such as scale, rotation, illumination, among others (Gil et al., 2009).

With the onset of binary descriptors, the focus of study began to include the computational advantage these and others presented in the face of the more traditional, already established, algorithms. One such notable study is the one by Miksik and Mikolajczyk (2012), which highlights the speed of Features from Accelerated Segment Test (FAST; Rosten and Drummond, 2006) for detection, and of Binary Robust Invariant Scalable Keypoints (BRISK; Leutenegger et al., 2011) and Oriented FAST and Rotated BRIEF (ORB; Rublee et al., 2011) for detection and description, in the face of the classical difference of Gaussians (DoG)/SIFT and Fast-Hessian/Speeded-Up Robust Features (SURF; Bay et al., 2006). However, like their preceding studies, the authors evaluate the methods on fixed, sparse image sequences, each one benchmarking a different transform (e.g. the VGG Oxford dataset¹), rather than application-specific data. There have been some publications focused on the latter, such as visual tracking for unmanned aerial vehicles (UAVs; Cowan et al., 2016) and grid map matching (Blanco et al., 2010); regarding space NCRV, the

¹<http://www.robots.ox.ac.uk/~vgg/research/affine/>.

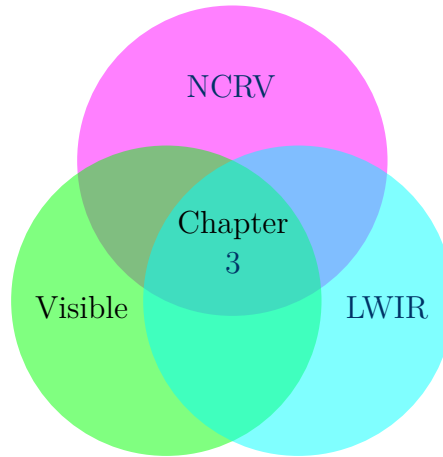


Figure 3.1: The domain of the present chapter. While literature exists on one or the intersection of two domains represented by each circle, this study introduces a connection between all three.

only reference in the literature found during the present survey was the study by Takeishi et al. (2015) on the benchmarking of the aforementioned IP algorithms for automatic landmark tracking on the Itokawa asteroid in the context of the Hayabusa mission. In it, the authors analyse their performance on a tumbling target navigation dataset in the visible wavelength, where they found that the algorithms suffer from low recalls in terms of corresponding interest regions when the angle of the asteroid shifts more than 20 deg and that the matching precision scores decline sharply after 10 deg.

Studies in the LWIR are certainly fewer in number, but they have been the object of recent study. Ricaurte et al. (2014) evaluate the behaviour of classic descriptors in a cross-modality outdoor dataset, finding that many of the algorithms are actually more robust to changes in rotation and scaling in the LWIR than in the visible. Johansson et al. (2016), and more recently Mouats et al. (2018), highlight the importance of experimenting with different combinations of detectors and descriptors in the LWIR, as these often outperformed the native setups. Yilmaz (2018) investigates the performance of feature detectors on thermal images of specific spacecraft materials, but lacks an analysis in the context of video sequences and NCRV.

In contrast, the analysis presented in this chapter intends to fill the gap in the literature by benchmarking local point feature detectors and descriptors for space NCRV, and ADR in particular, in both the visible and LWIR wavelengths (Figure 3.1). More specifically, the performance of the Harris, GFTT, DoG, Fast-Hessian, FAST, and CenSurE detectors and of the SIFT, SURF, LIOP, ORB, BRISK, and FREAK descriptors is assessed under the same lens as the seminal work done in the pure computer vision domain such as Mikolajczyk and Schmid's (2005). The analysis is

conducted not just in terms of the algorithm’s errors, distinctiveness, and robustness to changes in viewing conditions, but also in terms of their computational efficiency; the latter is carried out via processor-in-the-loop (PIL) simulations with an embedded board that parallels the expected power for an on-board computer.² The study is tailored to the domain of this thesis as the methods are evaluated on a branch of the ASTOS dataset. This smaller series, ASTOS-B, features a lower image resolution for on-board processing and a different rendezvous approach. The latter preserves the characteristics of NCRV but permits the computation of the two-dimensional feature morphing that occurs from frame to frame, making the computation of the ground truth possible.

3.3 Methodology

Each ADR application, or more generally rendezvous (RV) mission, using imaging systems must consider performance figures to assess the viability of the IP algorithms used. Feature detectors search an image for locations that are probable to match well in other images, and feature descriptors convert each region around the detected keypoint locations into a condensed vector that can be matched against other descriptors (Szeliski, 2011). First, the theoretical background for these algorithms is provided in Sections 3.3.1 and 3.3.2 for detectors and descriptors, respectively. Then, the figures of merit used in the assessment framework are defined in Section 3.3.3.

3.3.1 Feature Detectors

The analysed detectors can be classified into two groups. The first group consists of corner detectors, i.e. algorithms that extract points defined as the intersection of two edges. Conversely, the second group considers blob detectors, which extract points taking into account a supporting neighbouring region. This class of algorithms attempts to tackle many of the drawbacks of simple corner detectors, such as invariance to scale changes. The Laplacian of Gaussian (LoG) operator is often utilised to this end as the resulting function is sensitive to corners and edges (Lindeberg, 1994). However, the LoG involves the computation of second-order derivatives which are both sensitive to noise and computationally expensive.

²For the purposes of this chapter, these are defined as the major microprocessors used, or to be used, in most European space applications. See https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Computers_and_Data_Handling/Microprocessors for additional details.

3.3.1.1 Harris Corner Detector

Harris and Stephens (1988) assembled their historically influential computer vision algorithm from the mathematical formalisation of Moravec's (1980) work through the minimisation of the auto-correlation function that compares an image patch against itself shifted for small increments:

$$E(\mathbf{u}) = \sum_i w(\mathbf{x}^{(i)}) [\bar{I}(\mathbf{x}^{(i)} + \mathbf{u}) - \bar{I}(\mathbf{x}^{(i)})]^2, \quad (3.1)$$

where \bar{I} is the intensity of the greyscale image, $\mathbf{x} = [x \ y]^\top$ is the pixel position vector in \bar{I} , $\mathbf{u} = [u \ v]^\top$ is the displacement vector, and $w(\mathbf{x})$ is a weighting function. For small variations in position $\mathbf{u} = \Delta\mathbf{u}$, it is shown that Equation (3.1) can be written using a Taylor series approximation as

$$E(\Delta\mathbf{u}) \approx \Delta\mathbf{u}^\top \mathbf{A} \Delta\mathbf{u}, \quad (3.2)$$

where \mathbf{A} is the auto-correlation matrix defined as:

$$\mathbf{A} = w(\mathbf{x}) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \quad (3.3)$$

with “*” representing the convolution operator and $I_x := \partial\bar{I}/\partial x$, $I_y := \partial\bar{I}/\partial y$ evaluated at \mathbf{x} . The matrix \mathbf{A} contains the information on how stable the auto-correlation function is at a given point. Consider the eigenvalues of \mathbf{A} , the pair (λ_1, λ_2) . If both eigenvalues are small, that translates into an approximately constant intensity profile within a window. A small and a large eigenvalue are equivalent to a unidirectional texture pattern, i.e. the surface of $E(\Delta\mathbf{u})$ is flat along that direction. If the two eigenvalues are sufficiently large, it corresponds to a minimum in $E(\Delta\mathbf{u})$ and to a corner or other pattern that can be tracked reliably. Harris and Stephens (1988) propose a corner response function given by:

$$R = \det(\mathbf{A}) - k \text{Tr}(\mathbf{A})^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2, \quad (3.4)$$

where k is an empirical constant. The region is then considered a corner based whether the size of the response R is greater than a given threshold.

3.3.1.2 Good Features To Track

J. Shi and Tomasi (1994) attempt to further improve Harris and Stephens's (1988) work by proposing a different measure to determine what are Good Features To

Track (GFTT). Since the larger uncertainty component in the location of a matching patch is in the direction corresponding to the smallest eigenvalue, the proposed corner response function is merely dependent on it:

$$R = \min(\lambda_1, \lambda_2). \quad (3.5)$$

3.3.1.3 Difference of Gaussians

Although invariant to rotation, corner detectors such as Harris and GFTT employ a fixed window size which makes interest point detection sensitive to scale changes. In his Scale Invariant Feature Transform (SIFT) algorithm, Lowe (2004) makes use of scale-space filtering to tackle this issue. A difference of Gaussians (DoG) is used to approximate the LoG; it is obtained by computing the difference between two Gaussian blurs of the same image with different standard deviations separated by a constant factor, i.e. σ and $k\sigma$. Successive blurrings are performed until the last layer is transformed with a value of twice the initial σ . Once a complete octave is processed, this layer is down-sampled by a factor of 2, marking the start of the following octave. Once all the DoG are found, the resulting structure is searched for extrema in space (\mathbf{x}) and scale (σ): each sample point is compared to its eight neighbours in the current image and nine neighbours in the scale (Figure 3.2). It is selected as a potential feature if it is either larger or smaller than all of them.

As a further refinement, each potential feature is subjected to a rejection process based on a contrast threshold value. Additionally, in order to reject edges, a process similar to the Harris corner detector is employed by computing the 2×2 Hessian matrix $\text{Hess}_{\mathbf{x},\sigma}(\mathbf{D})$ of the difference image \mathbf{D} at the location and scale of the interest point

$$\text{Hess}_{\mathbf{x},\sigma}(\mathbf{D}) = \begin{bmatrix} D_{xx}(\mathbf{x}, \sigma) & D_{xy}(\mathbf{x}, \sigma) \\ D_{yx}(\mathbf{x}, \sigma) & D_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \quad (3.6)$$

and submitting its ratio of principal curvatures to an edge threshold. The quantities $D_{xx}(\mathbf{x}, \sigma) := \partial^2 \mathbf{D}(\mathbf{x}, \sigma) / \partial^2 x$, etc. are estimated by taking differences of neighbouring sample points.

3.3.1.4 Fast-Hessian

The Fast-Hessian detector was introduced as part of the Speeded-Up Robust Features (SURF) algorithm (Bay et al., 2006), which aimed to provide a computationally faster version of SIFT. The Fast-Hessian detector makes use of a further approximation of the LoG by using box filters, which can be evaluated swiftly independently of size

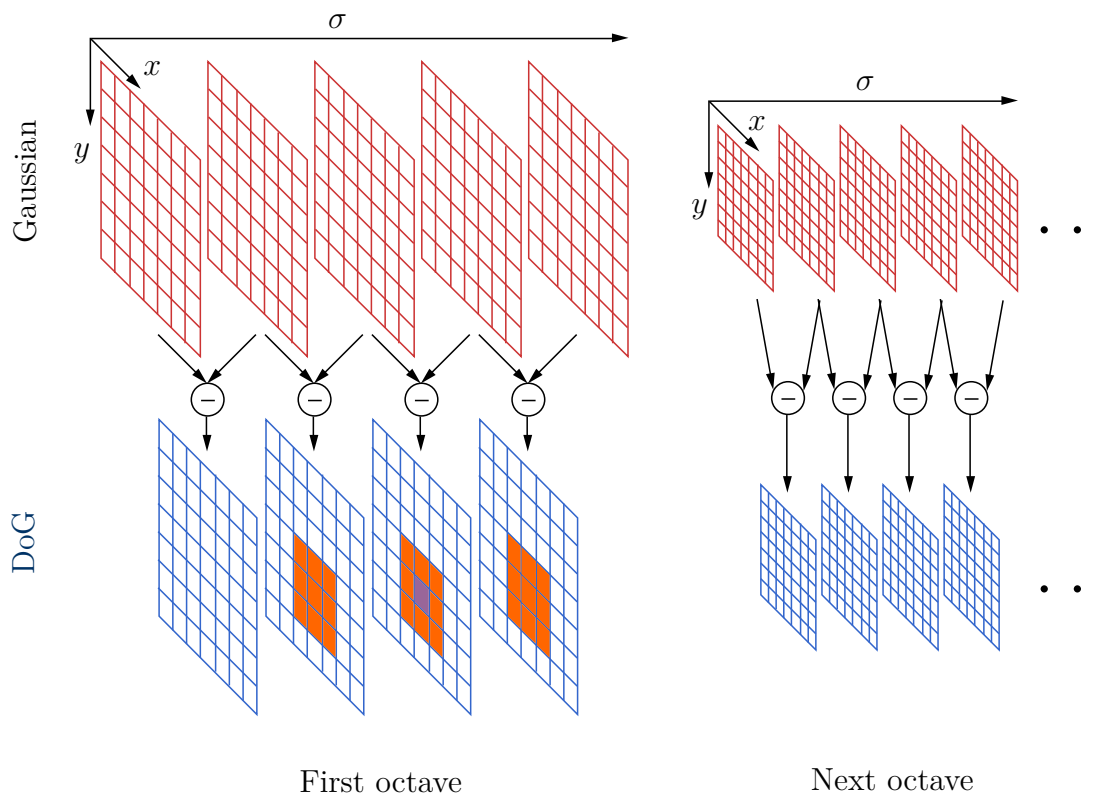


Figure 3.2: The difference of Gaussians (DoG) pyramid structure. Adjacent Gaussian images are subtracted to produce the DoG images, each octave is characterised by downsampling the previous one by a factor of 2. Features are selected in the DoG images by comparing a candidate point (*purple*) to its neighbours (*orange*) in scale and space.

using integral images. The box filters are used to compute approximations to the derivatives D_{xx} , etc. For instance, 9×9 box filters are approximations for Gaussian second order derivatives with $\sigma = 1.2$. These approximations are consequently used to produce an estimation of the determinant of the Hessian, which is used as a threshold for candidate features.

3.3.1.5 Features from Accelerated Segment Test

The Features from Accelerated Segment Test (FAST) algorithm (Rosten and Drummond, 2006) was developed with the purpose of creating a high-speed feature detector for real-time applications, such as VSLAM. FAST first selects a pixel $\mathbf{x}^{(i)}$ in the image as an interest point candidate. A circle of 16 pixels around $\mathbf{x}^{(i)}$ and a threshold t are defined. If there exists a set of n contiguous pixels in the circle which are all brighter than $\bar{I}(\mathbf{x}^{(i)}) + t$ or all darker than $\bar{I}(\mathbf{x}^{(i)}) - t$, then $\mathbf{x}^{(i)}$ is classified as a corner. The detection process is robustified through an offline machine learning stage, where a decision tree is built from alternative training images that is used in

deciding which pixels should be assessed first on the test images in order to exclude a large number of non-corners, hence improving detection speed. The algorithm also makes use of non-maximal suppression to avoid detecting multiple features adjacent to one another. Bearing a greater resemblance to corner detectors rather than blob detectors, **FAST** is not natively scale- or rotation-invariant.

3.3.1.6 Centre Surround Extrema

For **SIFT** and **SURF**, responses are not computed at all pixels for larger scales. At each successive octave, the sub-sampling is increased, so the accuracy of features at larger scales is sacrificed. One solution to tackle this problem in scale-space filtering is to approximate the **LoG** using bi-level centre-surround filters, as proposed for the Centre Surround Extrema (**CenSurE**) algorithm (Agrawal et al., 2008). This allows for the achievement of full spatial resolution at every scale.

Bi-level filters multiply the image intensity value by either -1 or 1 . The circular bi-level filter is shown to be the most faithful to the **LoG**, but the hardest to compute. Other filter shapes can be computed briskly with integral images, with decreasing cost from octagon to hexagon to box filter. After computing the filter responses, candidate features are subjected to a non-maximal suppression over the scale space in a $3 \times 3 \times 3$ neighbourhood. Lastly, the Harris measure from Equation (3.3) at the particular scale is used to filter out edge-like responses.

3.3.2 Feature Descriptors

The research pursued throughout this chapter features three floating point type, or distribution-based, descriptors and three binary type descriptors. Distribution-based descriptors are called as such since they encode (in a floating point vector) how certain elements of the support region to the feature point are distributed around it. The second type of considered local feature descriptor differs from the previous one in the sense that, instead of using a floating point vector representation, each descriptor consists of a binary string. For each feature point, a binary descriptor typically samples sets of pixel pairs $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}_i, i \leq n$ from the support patch, and performs a simple intensity comparison, where the result is 1 if $\bar{\mathbf{I}}(\mathbf{x}^{(1)}) < \bar{\mathbf{I}}(\mathbf{x}^{(2)})$, and 0 otherwise, generating an n -dimensional bit string. Using binary descriptors is advantageous as feature matching can be performed with resort to the Hamming distance, which provides better runtime performance with respect to the Euclidean distance test used with floating point descriptors.

Table 3.1: Characteristics of feature descriptors (adapted from Miksik and Mikolajczyk, 2012).

Descriptor	Data Type	# Elements	Size [bytes]	Matching Type
SIFT	Floating point	128	512	Euclidean norm
SURF	Floating point	64	256	Euclidean norm
LIOP	Floating point	144	576	Euclidean norm
ORB	Binary	256	32	Hamming norm
BRISK	Binary	512	64	Hamming norm
FREAK	Binary	512	64	Hamming norm

Remark 3.2: Hamming Distance

The Hamming distance between two strings of equal length is defined as the minimum number of substitutions required to convert one into the other. It is an operation with an efficient implementation, consisting only of applying the exclusive-OR (**XOR**) logical operator followed by a bit count.

Consider, for example, two binary vectors: $\mathbf{s}^{(1)} = [1\ 0\ 1\ 1\ 1\ 0\ 1]^T$ and $\mathbf{s}^{(2)} = [1\ 0\ 0\ 1\ 0\ 0\ 1]^T$. The **XOR** operation yields:

$$\begin{array}{r}
 1\ 0\ 1\ 1\ 1\ 0\ 1\ \mathbf{s}^{(1)} \\
 1\ 0\ 0\ 1\ 0\ 0\ 1\ \mathbf{s}^{(2)} \\
 \hline
 0\ 0\ 1\ 0\ 1\ 0\ 0\ \mathbf{XOR}
 \end{array}$$

As such, one has:

$$\mathbf{XOR}(\mathbf{s}^{(1)}, \mathbf{s}^{(2)}) = 2.$$

Table 3.1 highlights the differences between the descriptor types. Note that many of these algorithms were designed for detection as well as description. Indeed, DoG and Fast-Hessian are part of SIFT and SURF, respectively, and ORB and BRISK both use a FAST-based method for feature detection in their original implementations.

3.3.2.1 Scale Invariant Feature Transform

For the SIFT algorithm (Lowe, 2004), each keypoint is conferred an orientation by sampling the gradient magnitude and direction in a neighbourhood around it with

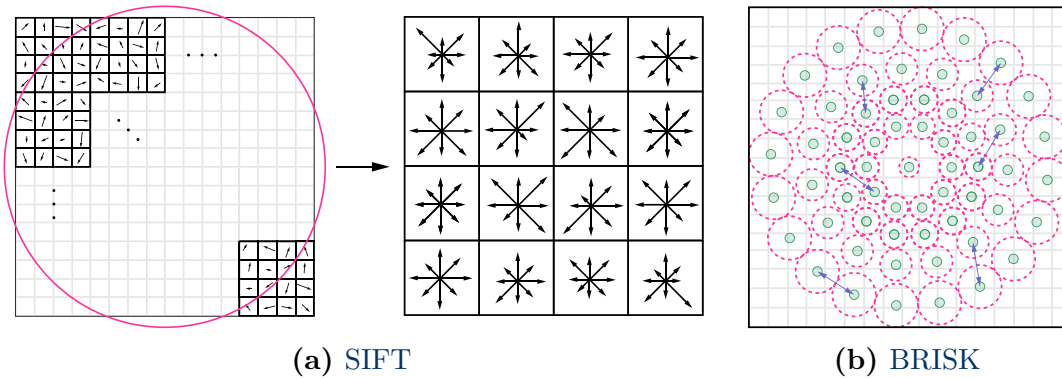


Figure 3.3: Distribution-based description and binary description. *(a, Left)* In Scale Invariant Feature Transform (SIFT), the gradient magnitude and orientation at each subregion are weighted by a Gaussian window (*pink*). *(a, Right)* The result is then accumulated into a histogram. *(b)* For Binary Robust Invariant Scalable Keypoints (BRISK), sampling locations (*green*) and Gaussian kernels to smooth intensity values (*pink*) for $n = 60$ points and some pairwise comparisons (*purple*) between them are shown.

a size dependant on the scale. The creation of the descriptor itself starts with the computation of the gradient magnitudes and orientations of a 16×16 sample array around the location of the detected interest point. The orientations are computed with respect to the keypoint’s own orientation in order to achieve rotation invariance. To avoid abrupt changes in the descriptor, the computed quantities are weighted by a Gaussian window. Then, the samples of each 4×4 subregion are aggregated into an orientation histogram, each orientation weighted by the corresponding magnitude. The descriptor is finally formed from a vector that holds the magnitudes of all the orientation histogram entries. Each histogram has 8 bins, giving the descriptor vector a size of 128 elements (Fig. 3.3a).

3.3.2.2 Speeded-Up Robust Features

For SURF (Bay et al., 2006), the orientation of each extracted region is assigned by computing instead the Haar wavelet responses (Stanković and Falkowski, 2003) in a circular neighbourhood of radius equal to six times the scale, which are then weighted with a Gaussian window centred at the feature point. The first step in building the descriptor itself is defining a square region of size twenty times the scale centred around the interest point and oriented along the previously defined direction. This area is divided into smaller 4×4 subregions, and for each of them the Haar wavelet responses are again computed, in the horizontal and vertical directions with respect to the orientation, and weighed with a Gaussian function. The sum of the wavelet responses and of their absolute values are stored in a four-dimensional

descriptor vector for each subregion, making up for a total of 64 elements. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation and is therefore conserved to allow for faster matching and an increase in performance.

3.3.2.3 Local Intensity Order Pattern

Local Intensity Order Pattern (**LIOP**; Z. Wang et al., 2011) is an algorithm for feature description designed to grant not only invariance to rotation and scale but also to complex illumination changes. As indicated by its name, it is based on order patterns, i.e. the order acquired by sorting the pixels of selected image patches by increasing intensity. It operates on the principle that this relative order remains unaltered in the case of monotonic intensity changes. First, the image is smoothed by a Gaussian filter as the relative order is sensitive to noise. Then, the size of each feature is normalised to a fixed diameter. The descriptor is constructed in an orientation-independent fashion, making it inherently invariant to rotation; therefore, the local patch is not rotated according to the local orientation as in **SIFT**. Afterwards, the overall intensity order is used to divide the local patch into subregions labelled ordinal bins. A **LIOP** of each point is defined based on the relationships among the intensities of its neighbouring sample points inside each bin. Lastly, the descriptor for the patch is constructed by concatenating the **LIOPs** of each bin together.

3.3.2.4 Oriented **FAST** and Rotated **BRIEF**

Oriented **FAST** and Rotated **BRIEF** (**ORB**) is a method supporting both feature detection and description (Rublee et al., 2011). It applies a pyramidal representation of **FAST** for multi-scale feature detection combined with a Harris corner filter for edge rejection. An orientation is assigned to the feature through the intensity centroid method, i.e. the assumption that a corner's intensity is offset from its centre, where the direction of the vector from the interest point to this centroid yields the orientation. The feature description procedure is based upon the Binary Robust Independent Elementary Features (**BRIEF**) mechanism (Calonder et al., 2010), i.e. the pixel pairs are sampled from an isotropic Gaussian distribution. The original **BRIEF** algorithm is not rotation-invariant though, so **ORB** first steers the computed descriptor according to the feature orientation. However, this causes a loss of variance in each descriptor string, which is undesirable as high variance makes a feature more discriminative since it responds distinctively to inputs. In order to recover from the loss of performance of steered **BRIEF**, a greedy search algorithm is employed to look through all possible binary tests to find sets that both have high variance and are

uncorrelated, resulting in a description processed coined “rBRIEF”.

3.3.2.5 Binary Robust Invariant Scalable Keypoints

As ORB, Binary Robust Invariant Scalable Keypoints (BRISK; Leutenegger et al., 2011) also employs a scale-space modification of FAST for feature detection. Likewise, the description process yields a binary string and is based on pixel intensity comparison tests. The key concept of the descriptor is the sampling pattern used: n locations equally spaced on circles concentric with the interest point (Fig. 3.3b). Two subsets are defined in accordance with two scale-proportional thresholds: one of short-distance pairings and another of long-distance pairings. The gradients of the long-distance pairs are used to compute the overall characteristic pattern direction of the feature. After that, the pattern is rotated accordingly and the binary descriptor string is assembled by performing all the short-distance intensity comparisons of pixel pairs. When sampling the image intensities for each pair, Gaussian smoothing is applied with a standard deviation proportional to their distance.

There are three main distinctions between BRISK and ORB. Firstly, BRISK’s uniform sampling pattern prevents accidental distortion of brightness comparison between pairs after Gaussian smoothing. Secondly, in BRISK a single point takes part in more comparisons, limiting the complexity the intensity values look-up process. Lastly, the comparisons are restricted spatially such that the brightness variations are only required to be locally consistent.

3.3.2.6 Fast Retina Keypoint

Fast Retina Keypoint (FREAK; Alahi et al., 2012) is a binary feature description algorithm which takes inspiration in the design of the human retina. The method adopts the retinal sampling grid as the sampling pattern for the pixel intensity comparisons: it is a circular geometry where the density of points drops exponentially from the centre outwards, mimicking the spatial distribution of ganglion cells in the eye. These are segmented into four different areas; this is believed to result in a body resource optimization, where a higher resolution is captured in the fovea (inner-most circle), while lower acuity images are formed in the perifovea (outer-most circle). To match this biological model, the algorithm uses different kernel sizes for the Gaussian smoothing of every sample point in each receptive field, where these overlap for added redundancy leading to increased discriminative power. To determine which pairs of pixels to compare, the authors defend that a coarse-to-fine pair selection yield the largest variance and uncorrelation between pairs, i.e. the first selected pairs compare sampling points in the outer circles and the last pairs compare points in

the inner circles. This is interestingly consistent with modern understanding of the retina, where the perifoveal fields are first used to estimate the location of a point of interest and the validation is then performed with the densely distributed foveal receptive fields. Effectively, to describe a (even static) scene, the eye moves around with discontinuous individual movements called saccades. As such, **FREAK** emulates this process by parsing the computed descriptor in a way that the first 16 bytes represent coarse information, which is applied as a triage in the matching process. This way, a cascade of comparisons is performed, accelerating the procedure even further. For rotation-invariance, the orientation of the feature is estimated using local gradients similarly to **BRISK**.

3.3.3 Performance Metrics

In order to evaluate the algorithms, the concept of correspondence is first defined: two regions, A and B , each from a different image, are said to be correspondences if the second region, when mapped to the first image, has an overlap with the first region higher than a defined threshold (Fig. 3.4). Formally, the following condition must hold:

$$1 - \frac{R_{M_A} \cap R_{(\mathbf{H}^\top M_B \mathbf{H})}}{R_{M_A} \cup R_{(\mathbf{H}^\top M_B \mathbf{H})}} < \varepsilon_0, \quad (3.7)$$

where R_M represents the elliptic region defined by $\mathbf{x}^\top \mathbf{M} \mathbf{x} = 1$, with \mathbf{M} being the 2×2 symmetric matrix of ellipse coefficients, and ε_0 is the overlap error threshold. This mapping, the ground truth, can be given by a 3×3 homography matrix \mathbf{H} , assuming a pinhole camera model and that the two related images represent same planar surface in space.

Consequentially, the repeatability score for a given pair of images is calculated as the ratio between the number of correspondences and the number of total features presented in the reference image:

$$\text{repeatability} := \frac{C^+}{C}. \quad (3.8)$$

A second type of testing performed is based on the matching score. This test verifies how well the regions can be algorithmically matched, thus assessing the distinctiveness of the detected regions. To this end, a descriptor for the regions is computed and the total matches M^* provided by it are checked to see if they agree with the correspondences obtained with \mathbf{H} . If a matched pair is also a correspondence, then it is deemed a correct match M^+ , contributing to the matching score as

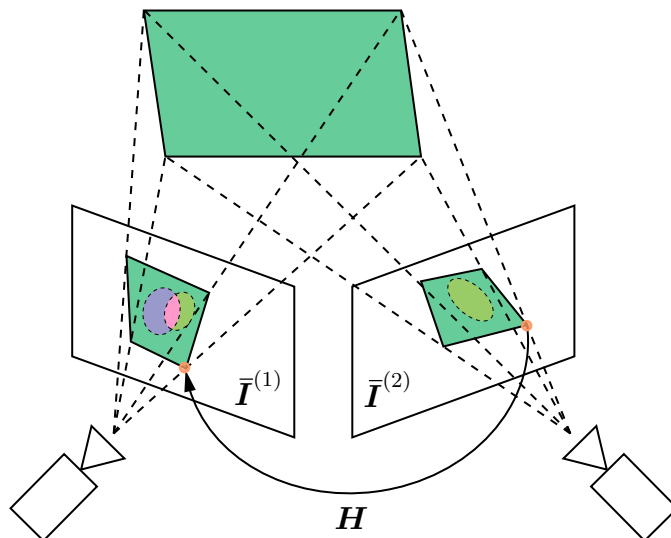


Figure 3.4: The homography ground truth \mathbf{H} maps the light green feature from image $\bar{I}^{(2)}$ to image $\bar{I}^{(1)}$. The overlap area with the original purple feature from $\bar{I}^{(1)}$ is shown in pink. If the amount of overlap is above a certain defined threshold, then the two features correspond.

$$\text{matching score} := \frac{C^+ \cap M^*}{C} = \frac{M^+}{C}. \quad (3.9)$$

Succinctly, features are desired to be repeatable, i.e. the same features should be observed regardless of how the target is manipulated, but they should also be distinctive enough so that they can be matched regardless of those transforms.

To evaluate the performance of feature descriptors, the figures of recall and precision are used. Recall is defined as the ratio of correct matches to the number of correspondences between a pair of frames:

$$\text{recall} := \frac{M^+}{C^+}. \quad (3.10)$$

On the other hand, precision is the ratio of correct matches to the total number of matches:

$$\text{precision} := \frac{M^+}{M^*}. \quad (3.11)$$

This performance metric is occasionally represented as its complement, i.e. $1 - \text{precision}$, the ratio of false matches to the total matches. For the ideal case, the recall and the precision would both be close to 1, meaning that the descriptor would return a great number of matches, all labelled correctly. A descriptor with high recall and low precision would translate into a great number of matches but many of them are false positives. Lastly, a descriptor with low recall and high precision

would mean a small number of returned matches, but most of them are correct.

Note that the definition of a match is dependent on the chosen strategy. Mikolajczyk and Schmid (2005) define three different ones. The first one is termed threshold-based matching, where two regions are matched if the distance between their descriptors is below a certain threshold μ . The second one is the nearest-neighbour (NN) based matching: regions A and B are matched if the descriptor $\mathbf{d}^{(b)}$ is the nearest neighbour to $\mathbf{d}^{(a)}$ and

$$\text{NN} = \|\mathbf{d}^{(b)} - \mathbf{d}^{(a)}\| < \mu. \quad (3.12)$$

For the scope of this chapter, the third and last concept of nearest-neighbour distance ratio (NNDR) is used: two regions are a match if the ratio of the distance to the first and to the second nearest neighbouring descriptors is below a certain threshold μ :

$$\text{NNDR} = \frac{\|\mathbf{d}^{(b)} - \mathbf{d}^{(a)}\|}{\|\mathbf{d}^{(c)} - \mathbf{d}^{(a)}\|} < \mu, \quad (3.13)$$

where $\mathbf{d}^{(b)}$, $\mathbf{d}^{(c)}$ are the first and second nearest neighbours to $\mathbf{d}^{(a)}$, respectively. While for threshold-based matching a descriptor can have several matches — and several of them might be correct — for the NN and NNDR-based techniques, a descriptor only has one match. The former strategy can be attractive for real-time applications due to low computational effort. However, setting the threshold value μ proves to be a difficult task, as a fixed value may bias the results towards a given region of interest, whereas for the other strategies, μ is relative to each pair. Results demonstrated by Mikolajczyk and Schmid (2005) and Mouats et al. (2018) show that the NN strategy results in high precision, as all matches below μ are rejected, diminishing the number of false matches; using NNDR improves the precision even further.

The performance of different descriptors is often compared by generating for each one sets of recall and 1-precision values with varying values of μ . The plotted points result in a receiver operating characteristics (ROC) curve (Szeliski, 2011, see Fig. 3.5).

Remark 3.3: Receiver operating characteristics curve

A ROC curve plots the recall, or true positive rate, versus the complement of the precision, or false positive rate, for a given classification task. For a perfect classifier, the recall is equal to 1 and the complement of the precision to 0. The area under curve (AUC) is a scalar figure of merit derived from the ROC

to evaluate a classifier’s performance: the larger the area under a descriptor’s ROC curve, the better its performance in terms of assigning matches, providing an intuitive way to benchmark descriptors.

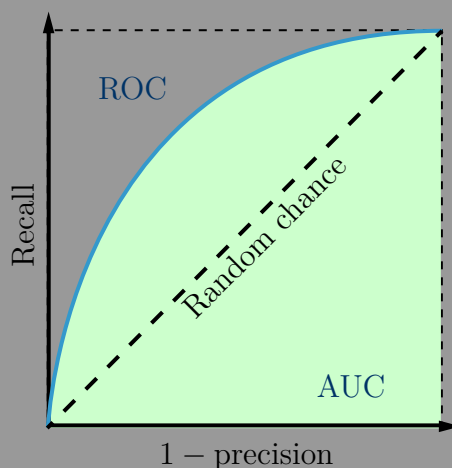


Figure 3.5: The receiver operating characteristics (ROC) curve and its related rates. Adapted from (Szeliski, 2011).

The average computation times per extracted and described feature are benchmarked, respectively, for each detection and description algorithm. This assumes a proportionality between the required time and the computation burden, which can then be of interest to make an informed choice on the algorithm for a given application.

3.4 Experiments

In this section, the experimental setup arranged to evaluate the performance of the IP algorithms is described. The generated datasets are delineated, and details of the implementation of the algorithms are outlined. The outline of the performed experiments is summarised in Table 3.2.

Table 3.2: Summary of experiments in Chapter 3.

Section	Description	Dataset
Section 3.4.5.1	Tuning of the parameters used in the benchmarked algorithms	ASTOS-B
Section 3.4.5.2	Benchmarking of feature detectors	ASTOS-B
Section 3.4.5.3	Benchmarking of feature descriptors	ASTOS-B
Section 3.4.5.4	Timings of computational execution times	ASTOS-B

3.4.1 Dataset

A second multimodal dataset was designed specifically for the scope of the proposed IP analysis framework. The dataset, **ASTOS-B**, was generated according to the same methodology as the **ASTOS** dataset (see Chap. 2) but features a different, specific trajectory to incorporate the necessary ground truth for the evaluation of 2D detectors and descriptors.

ASTOS-B features images generated with two simulated cameras with parameters specified in Table 3.3. One camera operates on the visible wavelength and the other operates on the **LWIR** wavelength, with the difference that both cameras now share the same field of view (**FOV**). This is to ensure that the scene is imaged similarly in terms of perspective projection for both modalities. Additionally, both cameras had their resolution scaled down to $320 \text{ px} \times 256 \text{ px}$ and acquisition rate set to 1 Hz to run the image processing functions on a low performance hardware board. The generated images simulate a rendezvous approach with Envisat, capturing realistic variations in illumination, rotation, and scale. Two mission scenarios are considered:

- (1) A “Hot Case”, where the spacecraft is in a sunlit section of their orbit; and
- (2) A “Cold Case”, where they are in eclipse, under no direct illumination from the Sun.

This yields a total of four different imaging sequences for the benchmarking of the IP algorithms, with 200 frames per sequence. The dataset hierarchical key is portrayed in Figure 3.6.

3.4.2 Ground Truth

To evaluate the proposed performance metrics (see § 3.3.3), the ground truth relating the changing of the scene between frames must be established. Generally, calibrating

Table 3.3: Simulated camera properties for the **ASTOS-B** dataset. The same parameters are used for acquisition in the visible and long-wavelength infrared (**LWIR**) modalities.

Parameter	Unit	Value
Resolution	$\text{px} \times \text{px}$	320×256
Focal length	mm	5
FOV	$\text{deg} \times \text{deg}$	51×40
Measurement rate	Hz	1

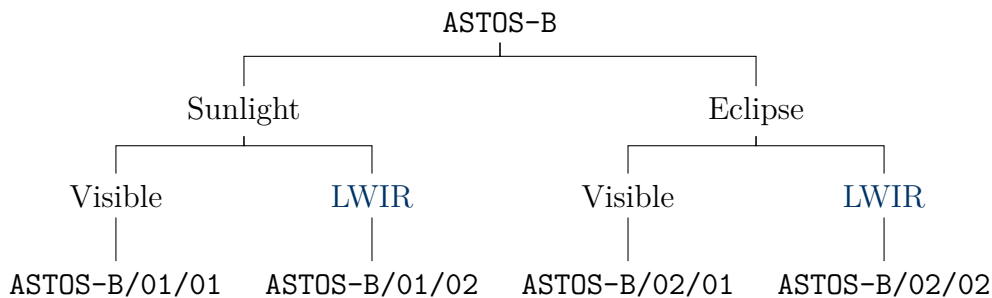


Figure 3.6: Key for the benchmarked trajectories of the ASTOS-B dataset.

a three-dimensional scene, or the motion of a 3D object as in the present case, would require the back-projection of the detected features into rays, computing their intersection with a 3D mesh of the target, and transferring the feature and its support region to a differently rotated and translated scene, similarly to the work of Takeishi et al. (2015), resulting in a highly complex and possibly time-consuming framework.

However, this can be greatly simplified by simulating a planar scene. This is achieved by modelling a rendezvous approach such that the same facet of the target is constantly visible. In this case, the ground truth can be computed just from the dataset itself, without resorting to the computer-aided design (CAD) model of the target, via a 3×3 homography matrix \mathbf{H} (Hartley and Zisserman, 2004). 2D points $\mathbf{x}^{(i)}$ in one frame are related to those $\mathbf{x}'^{(i)}$ in another frame as:

$$\mathbf{x}'^{(i)} = \mathbf{H}\mathbf{x}^{(i)}, \quad (3.14)$$

where the points $\mathbf{x}^{(i)} = [x \ y \ 1]^\top$, $\mathbf{x}'^{(i)} = [x' \ y' \ 1]^\top$ are expressed in homogeneous coordinates.

The homography matrices can be computed directly from feature point correspondences between each frame (see, for example, Hartley and Zisserman, 2004, Chapter 4); however, a different approach is taken to avoid biasing the algorithms to be tested, similarly to Mouats et al.’s (2018) work. First, putative feature matches between the two frames are obtained using a detector and descriptor not included in the benchmarks. This work uses Accelerated KAZE (AKAZE; Alcantarilla et al., 2013)³ features for this purpose. Then, an initial homography $\hat{\mathbf{H}}$ is estimated from these matches using Random Sample Consensus (RANSAC; Fischler and Bolles, 1981) to reject outliers. Finally, $\hat{\mathbf{H}}$ is used to initialise a forward additive enhanced correlation coefficient (ECC) algorithm (Evangelidis and Psarakis, 2008) to compute a refined \mathbf{H} . Figure 3.7 illustrates the ground truth computation for a pair of frames

³Here, “kaze” is not an acronym, but the romanisation of the Japanese word “風”, meaning “wind”, an allusion to the algorithm’s speed.

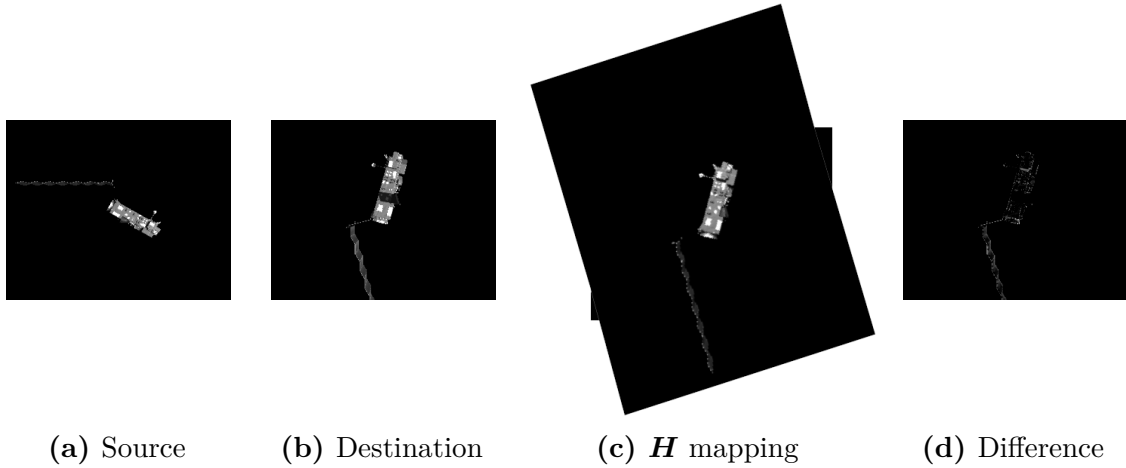


Figure 3.7: Homography computation example for a pair of wide baseline frames. The source image is mapped to the destination image via \mathbf{H} . The quality of the transform is related to the residual difference image between the true and mapped destination images.

in the dataset.

3.4.2.1 Planarity Assumption

The facet of the spacecraft’s main body that is constantly observed by the chaser is approximately flat (cf. Fig. 3.7) and hence well modelled by a plane $\Pi_{\mathbf{H}}$ parallel to $\underline{t}^{(2)}\text{-}\underline{t}^{(3)}$ in the target frame \mathcal{F}_t (see the frames of reference defined in Fig. 3.8). This represents the dominant plane based on which the planar homography in Equation (3.14) is computed.

The solar panel is not contained in this plane, meaning that Equation (3.14) would normally not model the ground truth adequately. However, for this particular motion a valid planar assumption is upheld as follows. Since the motion of the chaser is always parallel to $\Pi_{\mathbf{H}}$ (see § 3.4.3), the only apparent transformation experienced by the solar panel not explained by \mathbf{H} is due to perspective projection (i.e. the dimension along $\underline{t}^{(1)}$ appears longer the closer the target is to the camera). In the computation of the homography between consecutive frames, due to the reduced motion the changes in the solar panel caused by perspective projection are not observed, thus producing a stable \mathbf{H} . When computing it for larger baselines, the number of frames is limited so as to maximise the length of the sequence for benchmarking while minimising the perspective projection deformations and hence keeping the stability of \mathbf{H} . In this way, features detected on both the main body and the solar panel can be accurately benchmarked without violating the planarity assumption and the validity of the ground truth.

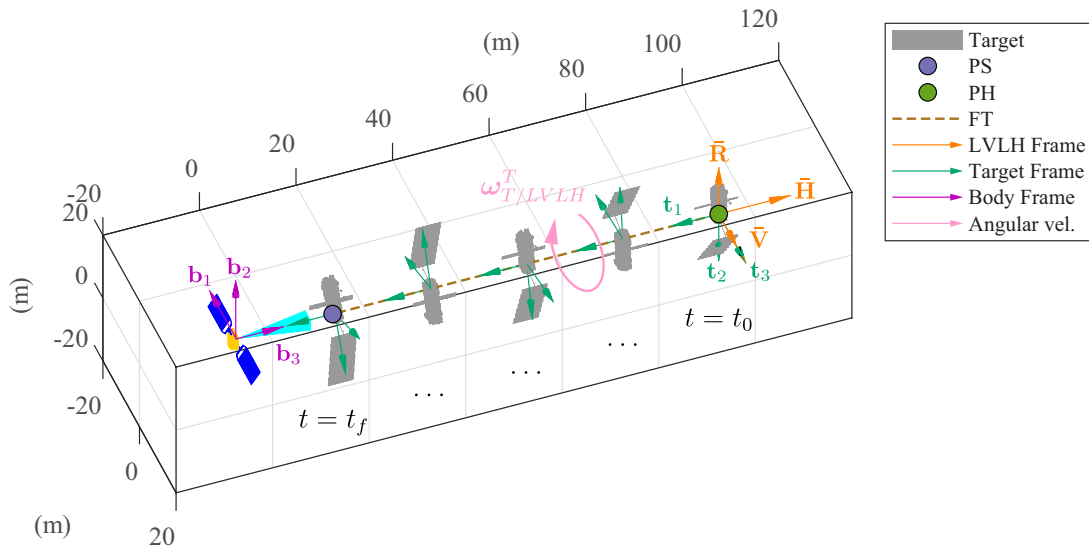


Figure 3.8: Scenario specifications for the ASTOS-B dataset trajectory generation, centred in the chaser’s body frame \mathcal{F}_b . The relative configurations of the target frame \mathcal{F}_t and of the local-vertical-local-horizontal (LVLH) frame \mathcal{F}_o are shown.

3.4.3 Orbital Dynamics

A chaser spacecraft is assumed to approach the target with the translational profile relative to the local-vertical-local-horizontal (LVLH) reference frame illustrated in Figure 3.8. The spin axis of the target in the target frame, \mathcal{F}_t , is aligned with the positive $t_1^{(1)}$ axis, and the spin axis in the LVLH frame, \mathcal{F}_o , is aligned with the positive H-bar axis; the rotation rate is 3.5 deg s^{-1} . The chaser (\mathcal{F}_b frame) assumes a constant orientation with regards to \mathcal{F}_o . The sequence begins with the chaser in a hold point (“PH”) 100 m away from the target. The rendezvous sequence is performed through a forced translation H-bar approach (“FT”) with the target until a stop point (“PS”) is reached at 20 m distance, after which the sequence ends.

The real orbit of Envisat is emulated using two-line element (TLE) data analogously to the ASTOS dataset, see Chapter 2, Section 2.5.2 for details.

3.4.4 Implementation

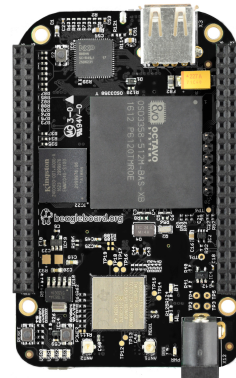
The orbital states of both chaser and target, the camera parameters, and a 3D CAD model of Envisat are used as inputs to the Astos Camera Simulator to generate the dataset; see Chapter 2 for details.

The performance analysis framework was coded in the C++ programming language. The OpenCV library (version 3) was used for computer vision and image processing related functions. The implementations of every detector and descriptor

Table 3.4: The BeagleBone Black (BBB) wireless single board computer.

Parameter	Specification
System on a Chip	AM3358/9
CPU	Cortex-A8 1 GHz
Digital Signal Processor	N/A
On-board storage	8 bit eMMC (running Ubuntu 16.04), microSD card 3.3 V supported
Memory	512 MB DDR3
Size	86.40 mm×53.3 mm
Power ratings	210–460 mA at 5 V

(a) Hardware properties



(b) Image of the board

used are publicly available from OpenCV, except for LIOP, where the author's original open-source code⁴ was used. The implementation of DoG+SIFT is based on the code of Rob Hess⁵. Fast-Hessian+SURF, Harris, and GFTT are direct adaptations of the original papers (Bay et al., 2006; Harris and Stephens, 1988; J. Shi and Tomasi, 1994). FAST, FREAK, BRISK, and ORB are ports of the authors' own implementations. Lastly, for CenSurE, the OpenCV implementation is termed STAR and it is an altered version of the original algorithm (Agrawal et al., 2008) for added computational stability and speed.

To verify the computing performance of the IP methods, these were implemented and tested on a BeagleBone Black (BBB) single-board computer with a 1 GHz ARM Cortex-A8 processor and 512 MB DDR3 RAM (Table 3.4). The board was listed by Dubois-Matra (2016) as one of the ESA-approved microprocessors.

3.4.5 Results

In this section, the results of the adopted framework to determine the performance of the algorithms on the multimodal dataset are delineated. The pipeline is based on the works of Mikolajczyk and Schmid (2005), Mikolajczyk, Tuytelaars, et al. (2005), and Mouats et al. (2018) with some modifications given the nature of the dataset.

The first one refers to the considered dataset itself. Unlike common studies which consider scenes where the detected features are distributed over the whole image, for the images in the present dataset the background is featureless and the target may occupy a relatively small area. This imposes a limitation on the number of features that can be extracted from each frame. Since it is desirable to have a

⁴<https://github.com/foelin/IntensityOrderFeature>.

⁵<http://robwhess.github.io/opensift/>.

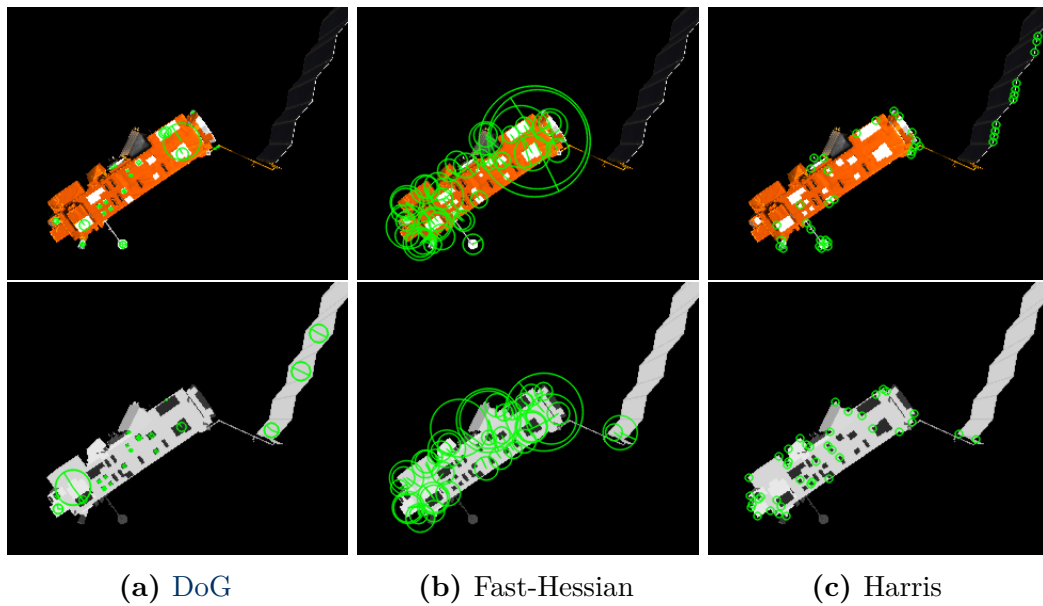


Figure 3.9: Examples of detected features (*green*) on hot case frames from the dataset. (*Top Row*) visible wavelength. (*Bottom Row*) long-wavelength infrared (LWIR).

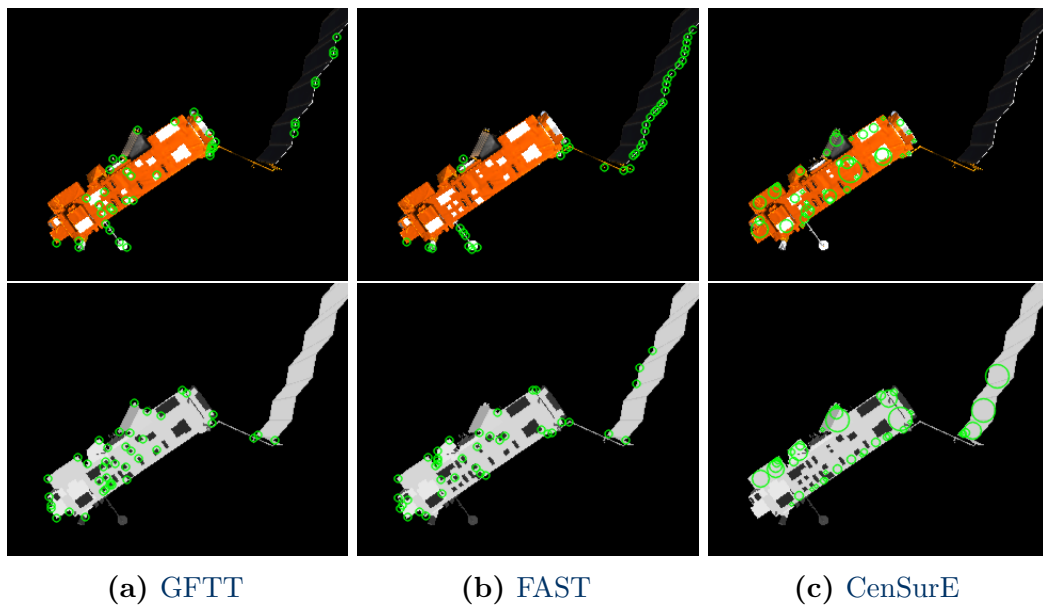


Figure 3.10: Examples of detected features (*green*) on hot case frames from the dataset. (*Top Row*) visible wavelength. (*Bottom Row*) long-wavelength infrared (LWIR).

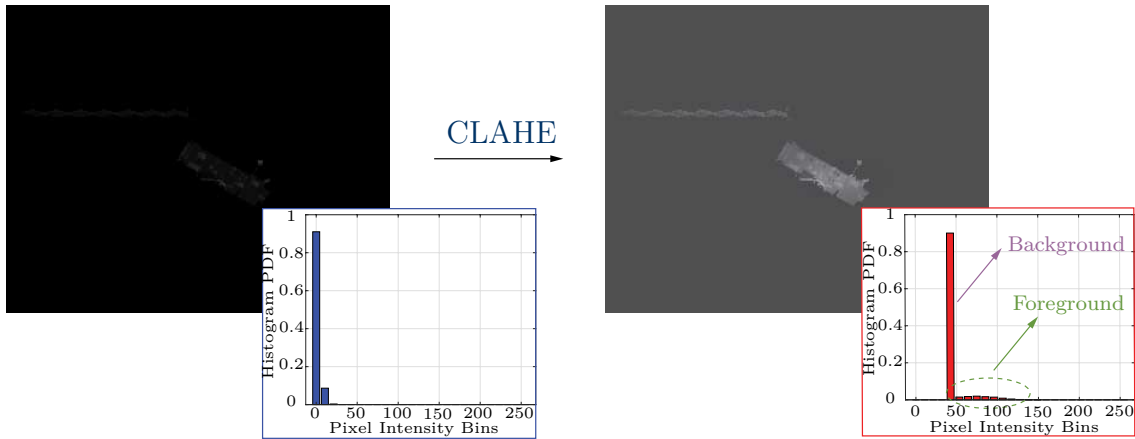


Figure 3.11: Effect of contrast limited adaptive histogram equalisation (**CLAHE**) on the visible cold case. (*Left*) The original image, where the foreground is virtually indistinguishable from the background. (*Right*) The image after application of **CLAHE**, where the intensity of the foreground pixels becomes more spread out over the range of possible values, resulting in the enhancement of the target.

number of detections that is constant across frames and sequences for a balanced basis of comparison, this implies adequately tuning the sensitivity thresholds of each algorithm instead of relying on the default values. Samples of generated images and of detected features are illustrated in Figures 3.9 and 3.10; the number of plotted features is limited to 40 for clarity.

The second modification, also related to the dataset, considers the particular case of the eclipse in the visible band. For this sequence, the target is barely visible, as the only source of illumination is light reflected by the Earth’s atmosphere. This is illustrated in Figure 3.11 (left), where it can be seen on the histogram of image pixel intensities that the values are concentrated to the left of the spectrum, next to the largest bar representing the background. This has a limiting effect on the number of features that can be detected, which is a problem since it is intended to compare the **IP** algorithms under similar conditions. To enhance the visualisation of the target in these conditions, adaptive histogram equalisation is employed: the image is automatically divided into different sections (the default in OpenCV is a tile size of 8×8) and a histogram is computed for each one. The pixel intensities in each histogram are then equalised, improving the contrast and the edges (and hence, corners). To prevent overshooting that could amplify noise, the output contrast is limited, in what is called contrast limited adaptive histogram equalisation (**CLAHE**). After equalisation, bilinear interpolation is used to cull artefacts on tile borders. The result is displayed in Figure 3.11 (right).

The last aspect concerns the implementation of the algorithms. Apart from

differing internal mechanics, the computational code of each algorithm has been developed by different authors. As such, the parameters used to tune each one are not uniform. Consider, for example, the non-maximum suppression functionality: the process of removing multiple interest points that were detected in adjacent locations, leaving only the most distinctive ones. For Harris, GFTT and CenSurE, it is possible to set the suppression window size as an input parameter; for FAST it is only possible to toggle the functionality on or off; whereas for DoG and Fast-Hessian it is not controllable at all. In general terms, the smaller the suppression window, the more features are obtained, but the less distinctive they will be. These differences in interface make it difficult, to guarantee that each processed sequence will have the same ratio of feature number to feature distinctiveness. During the testing campaign upon which this chapter is written, it has been observed that turning off non-maximum suppression on a feature detector, when given the option, leads to a sharp drop in performance when compared to the others. Therefore, it is ensured that non-maximum suppression is activated for a fair benchmark. Another aspect to consider is the implementation performance of each algorithm.

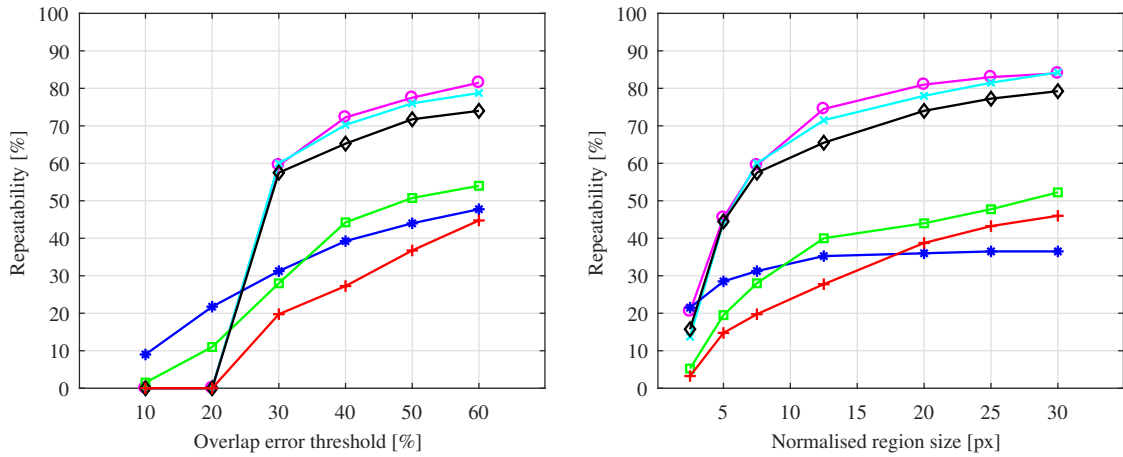
3.4.5.1 Tuning of Benchmark Parameters

Firstly, the different parameters that have a potential influence on the algorithms' benchmarking setup is analysed. A pair of common frames from each sequence, corresponding to an original image and a transformed one, is selected. The resulting data are averaged over all sequences for each feature detector and plotted in Figure 3.12.

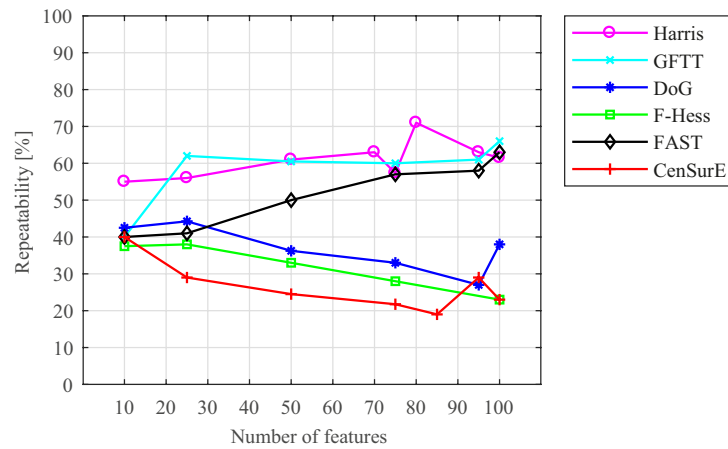
Accuracy of the Detectors Figure 3.12a illustrates the repeatability as a function of the overlap error threshold for the two considered bands. As the overlap error, i.e. the requirement to qualify two regions as corresponding, is relaxed the repeatability score goes up. For strict thresholds (10–20 %), the three corner detectors demonstrate a null repeatability, but become the highest-ranking ones as the threshold is relaxed. This shows that these detectors are less accurate than the others for this type of scenario. CenSurE exhibits a similar behaviour but does not change its order relative to the others, scoring below the remaining two blob detectors. An overlap error threshold of 30 % is selected to ensure non-zero repeatability scores.

Normalised Region Size Secondly, the effect of the choice on the normalised region size is studied; the results are displayed in Figure 3.12b. This test was conducted with a fixed overlap error threshold of 30 %. The relative ordering of the feature detectors remains the same, save for DoG: for the minimum considered radius,

3. BENCH'ING OF DETECTORS AND DESCRIPTORS FOR NAV'N



(a) Repeatability vs overlap error threshold (b) Repeatability vs normalised region size



(c) Repeatability vs region density

Figure 3.12: Repeatability scores as a function of different benchmark parameters.

it ranks first in repeatability score, but as soon as this parameter is increased, it is surpassed by the corner detectors and begins to saturate beyond 12.5 px. Choosing a normalised region size of 7.5 px will limit the bias in further evaluations.

Region Density For this test, the effect of increasing the number of features on the repeatability of the detectors is considered. This is achievable by altering the tuning parameters for each algorithm, allowing them to be compared when they output a similar amount of interest points. This is plotted in Figure 3.12c, where the overlap error threshold and the normalised region size were set to 30 % and 7.5 px, respectively. It can be seen that the corner detectors (i.e. Harris, GFTT, and FAST) tend to improve their repeatability scores when the number of features is increased, whereas the opposite is observed for the blob detectors (i.e. DoG, Fast-Hessian, and CenSurE). Note that the scores of DoG and CenSurE slightly increase towards the maximum considered number of detections, which indicates that these algorithms

could possibly be less robust to noise: the quality threshold of the extracted features must be lowered to increase detections, which can lead to more false positives.

3.4.5.2 Benchmarking of Feature Detectors

For this test, the repeatability and number of correspondences obtained by each detector for each full sequence is analysed. In addition, the matching scores and number of matches is computed. This is done using the **LIOP** descriptor. This descriptor was chosen as it is independent from all the detectors considered. Since the goal is to study the performance of the different feature extraction processes, this avoids any bias towards a specific detector, allowing for the examination of the features' distinctiveness regardless of the chosen descriptor. For added comparison, the performance using the original descriptors for **DoG** and **Fast-Hessian** (**SIFT** and **SURF**, respectively) is also showcased to benchmark the full original algorithms and provide a baseline.

An overlap error threshold of 30 %, a normalised region size of 7.5 px, and a fixed number of 75 extracted features for each detector are considered. The benchmarks are plotted in Figures 3.13 to 3.20. As in Mouats et al. (2018), two plots are provided for each sequence:

- (1) The first benchmarks consecutive image transforms, which is commonly done in structure from motion (**SFM**) and **VSLAM** algorithms; in this case, a value pertaining to frame $\bar{\mathbf{I}}^{(\kappa)}$ in the plot is referent to the transformation between frames $\bar{\mathbf{I}}^{(\kappa)}$ and $\bar{\mathbf{I}}^{(\kappa+1)}$; while
- (2) The second plot demonstrates the behaviour of the detectors when faced with large image transformations, which is usually the case encountered when applying model-based navigation strategies; the number "0" is used to represent the reference image (a frame from the middle of each sequence is chosen), whereas positive numbers represent transformations in posterior frames with respect to that reference and negative numbers represent those prior to it.

Visible Modality Hot Case Figures 3.13 and 3.14 showcase the performance of the detection algorithms for the **ASTOS-B/01/01** approach sequence. **Harris**, **GFTT**, and **FAST** achieve the highest repeatability scores. However, in terms of matching scores, they are comparable to **Fast-Hessian** and **CenSurE**, where the former actually outperforms the rest towards the end of the sequence, showing a bias in favour of shorter target ranges. Conversely, the correct matches when using **GFTT** and **FAST** actually decrease as the chaser nears the target, meaning that the high number of

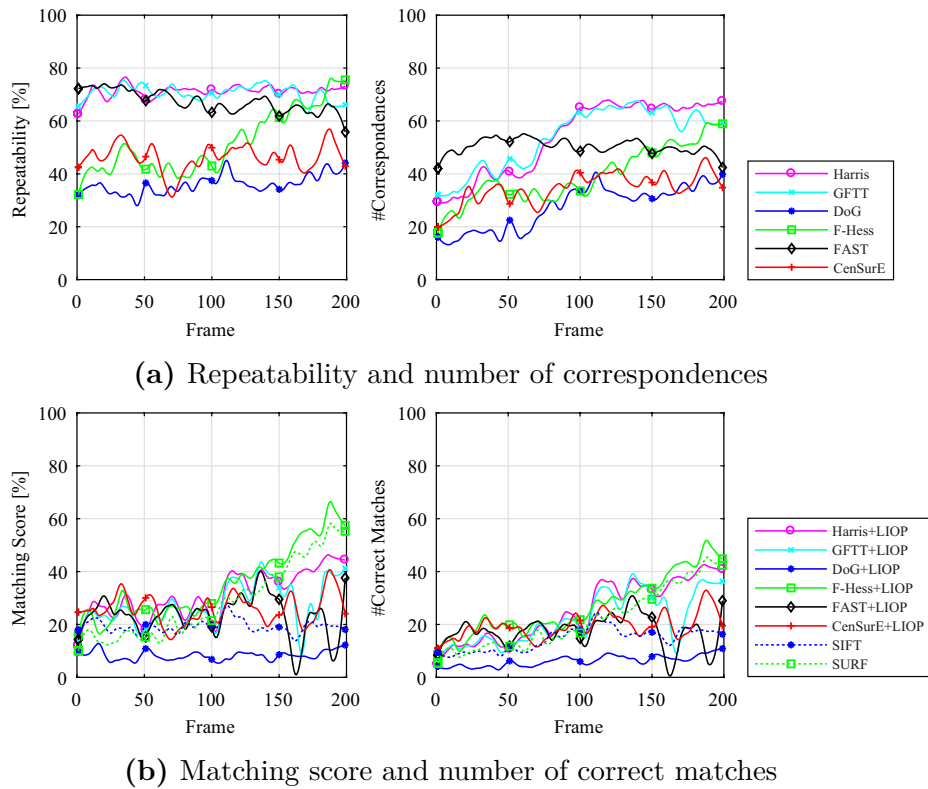


Figure 3.13: Performance for ASTOS-B/01/01 rendezvous sequence: successive transformations, visible band, hot case. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

obtained correspondences likely stems from accidental overlap. This could represent a problem when using these detectors with visible imagery at close proximity. CenSurE is the most consistent algorithm throughout. Note from Figure 3.13b that Fast-Hessian shows a better performance when coupled with LIOP than when combined with its native descriptor. From Figure 3.14 it can be seen that the detection algorithms are in general less resilient to large image transformations. In spite of a high repeatability for variations relatively close to the baseline, the number of correct matches of the three corner detectors drops rapidly; for sufficiently large transformations, they produce no correspondences at all. Interestingly, DoG when used with its native SIFT is shown to be the most robust in terms of matching score for large variations, when it performed the worst for small variations.

Visible Modality Cold Case Figures 3.15 and 3.16 represent the results obtained for the ASTOS-B/02/01 trajectory. Generally, the repeatability scores are quite similar to the hot case both in trend and magnitude. In opposition, the matching scores are now seen to decrease with time; the exception is Fast-Hessian combined with

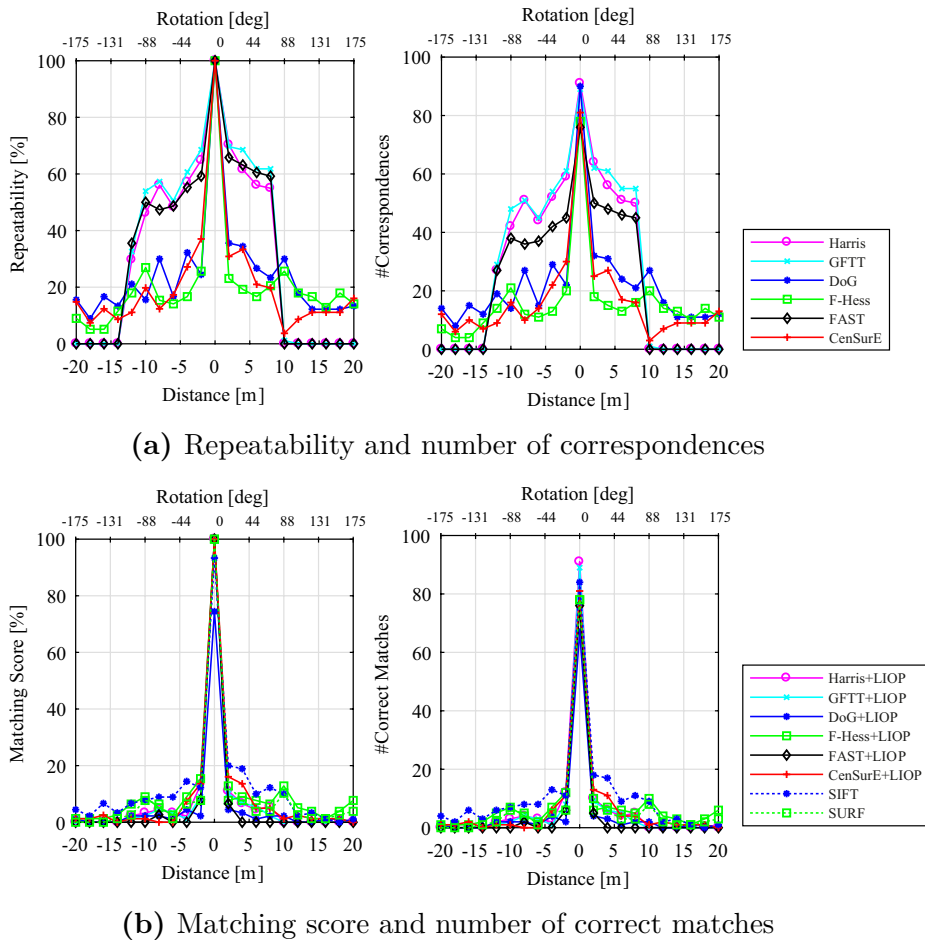


Figure 3.14: Performance for ASTOS-B/01/01 rendezvous sequence: large transformations, visible band, hot case. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

LIOP, which remains consistent, and the same detector use with SURF, which actually increases performance with time. The decreasing number of matches when the correspondences are increasing indicates that as the sequence progresses the features are becoming less distinctive to be correctly matched. In terms of large variations, the matching score decreases more sharply than in the hot case; this could be explained by a decreased consistency in the target pixels' intensity values due to CLAHE between the reference and query frames.

Thermal Infrared Modality Hot Case Figures 3.17 and 3.18 show the results attained for the ASTOS-B/01/02 rendezvous sequence. The algorithms suggest robustness in this modality with high repeatability scores overall (notably in the case of the blob detectors: DoG, Fast-Hessian, and CenSurE) and matching scores increasing with time. Note that FAST shows significant declines in the matching

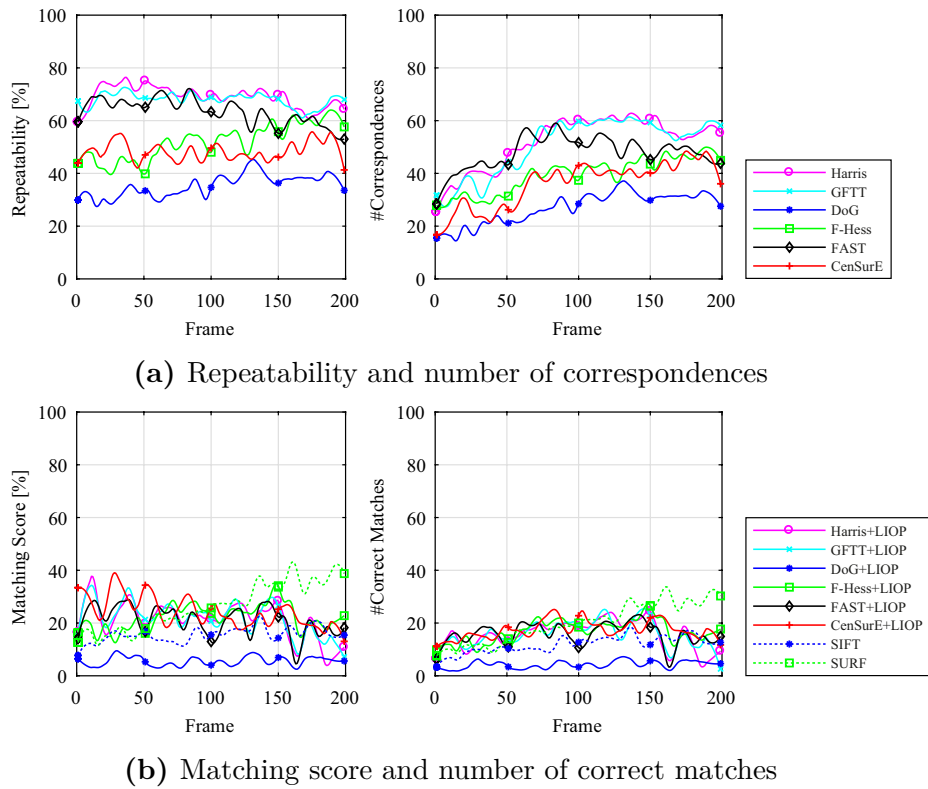


Figure 3.15: Performance for ASTOS-B/02/01 rendezvous sequence: successive transformations, visible band, cold case. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

score despite its high repeatability, illustrating lower feature distinctiveness when compared with the other corner detectors. Fast-Hessian again scores one of the highest benchmarks in general. From Figure 3.18 the behaviour of the detectors is less consistent: FAST and DoG outperform the other algorithms in medium transformations (up to ± 10 m and ± 90 deg baselines) with respect to matching score, whereas Fast-Hessian provides the best performance for larger variations.

Thermal Infrared Modality Cold Case Figures 3.19 and 3.20 illustrate the detector performance for the ASTOS-B/02/02 sequence. For both consecutive and large transformations, the number of correct matches is generally lower than for the hot case in the same band; in spite of that, the repeatability scores are similar, which suggests that the cold case generates less distinctive features. This is more noticeable in the case of FAST, whereas Fast-Hessian and CenSurE are more impervious to the changes in temperature. It is however important to note that Harris and GFTT recover greatly towards the end of the sequence in terms of correct matches for successive transformations, outperforming the remaining detectors (Figure 3.19b).

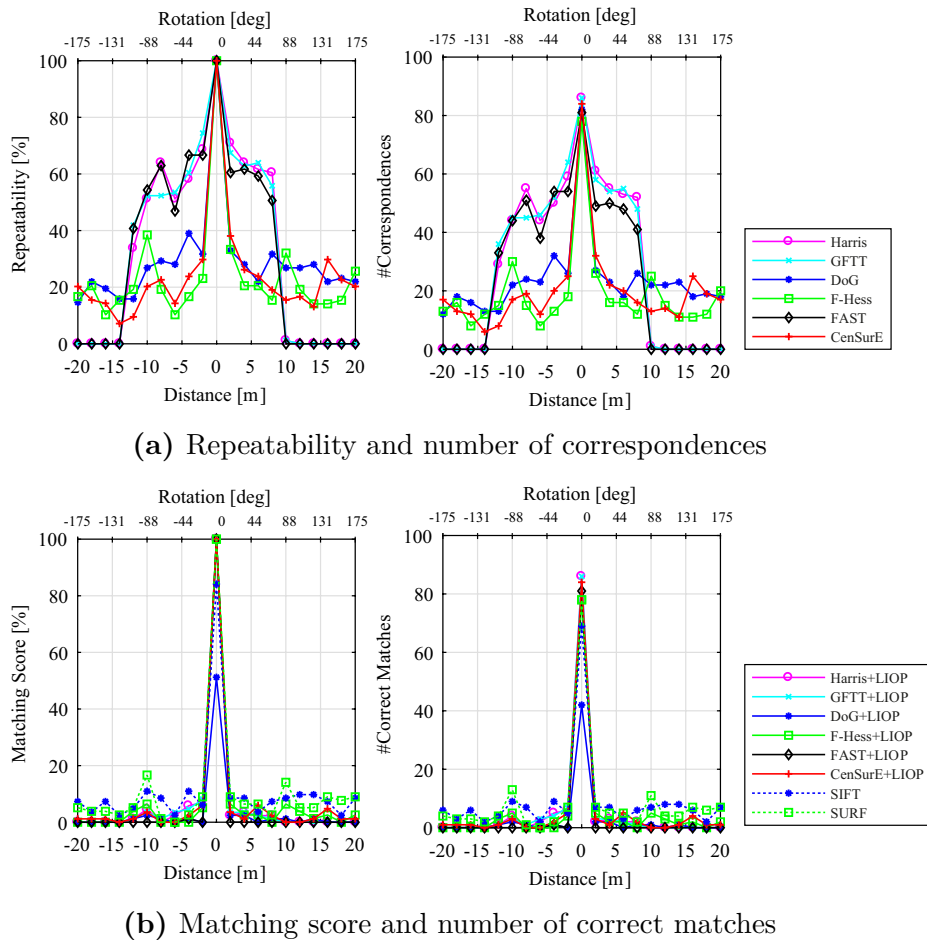


Figure 3.16: Performance for ASTOS-B/02/01 rendezvous sequence: large transformations, visible band, cold case. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

Discussion

Despite being imaged in two different modalities, the simulated sequences feature a common relative motion. Therefore, some similarities in the results are expected. The repeatability trend for the successive transformations, in particular, is similar for all four sequences: corner detectors tend to be the most robust and for blob detectors the score tends to increase with the inverse of the distance to the target. For large transformations, the repeatability of corner detectors drops to zero after a certain point, whereas blob detectors are resilient. The same cannot be said about the matching scores, however: despite scoring generally lower than the repeatability, they vary in trend and relative ranking between sequences. This highlights the importance in using descriptors to compute matches instead of relying on the geometry overlap only, and implies different degrees of distinctiveness in extracted features depending on the detector, wavelength, and illumination condition considered.

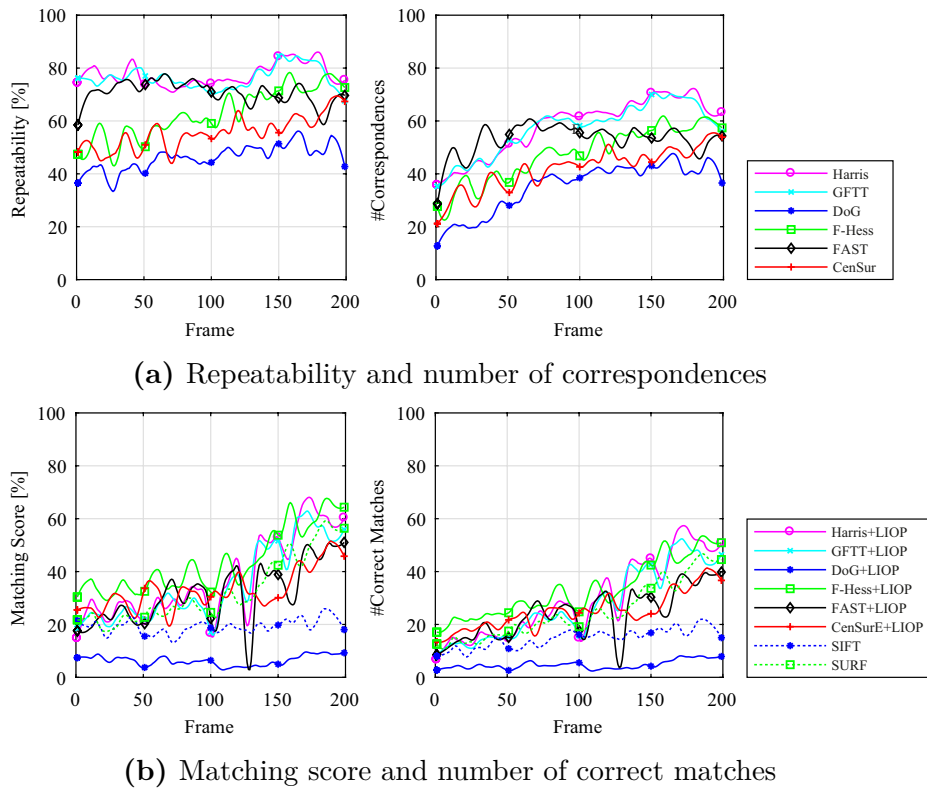


Figure 3.17: Performance for ASTOS-B/01/02 rendezvous sequence: successive transformations, thermal infrared band, hot case. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

Despite their high repeatability, corner detectors are often equalled or even surpassed by the blob detectors in terms of matching score. Despite high repeatability, FAST is one of the least distinctive algorithms across all tests. Fast-Hessian performs well in terms of matching scores in most cases despite average repeatability; the exception is the visible cold case, where there is a generalised loss of performance, but it still maintains a good ranking in relative terms. This suggests an extraction of quite distinctive features, which confirms what was stated in the LWIR analysis of Mouats et al. (2018) and extends the conclusions to the visible spectrum. This is an important finding as it is desirable to have a detector that works well in both spectra. DoG shows low scores for successive transformations regardless of the wavelength and illumination, but seems to perform worse on the LWIR cold case. On the other hand, its performance is comparable to the other blob detectors when dealing with large transformations. It performs better with SIFT than with LIOP in every situation, whereas Fast-Hessian usually performs better with LIOP than SURF, the exception for the latter being the visible cold case. This reiterates the importance of testing detectors and descriptors separately to avoid any cause of bias.

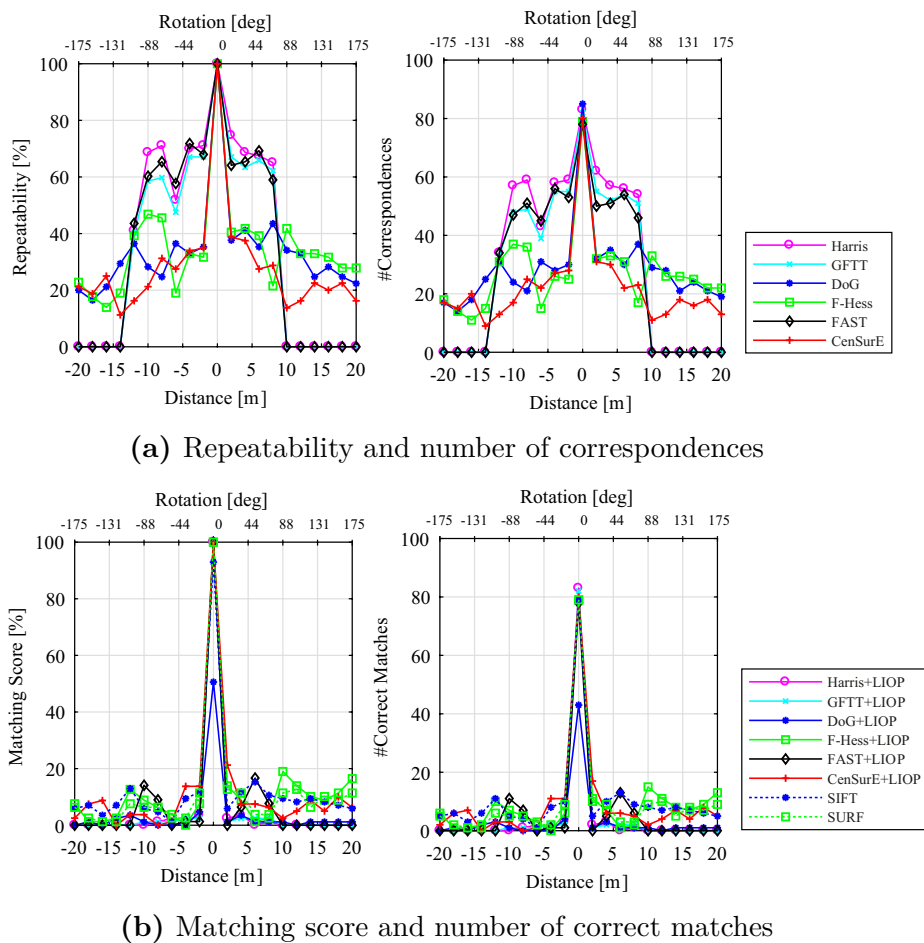


Figure 3.18: Performance for ASTOS-B/01/02 rendezvous sequence: large transformations, thermal infrared band, hot case. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor

In the benchmarking of successive transforms, corner detectors are shown to lose in performance when the target is closer on the visible. This could signify that they are more sensitive to noise inherent to the multi-layer insulation (MLI), for example, as their matching scores are better on the textureless LWIR. In the latter case, the actual corners are better defined and impervious to illumination changes. On the same note, performance is generally better for the LWIR sequences: for the hot cases, CenSurE and Fast-Hessian, in particular, are comparable to the visible case, but the former performs better than its visible counterpart in the end of the sequence where the latter does so in the beginning of it. In the visible eclipse sequence, the efficiency of the algorithms is greatly diminished. This finding suggests that an artificial solution such as CLAHE to tackle the cold case is not a feasible solution. It does allow for the detection of more features, but these are not distinctive enough to guarantee an acceptable matching score. The use of a thermal infrared camera is a

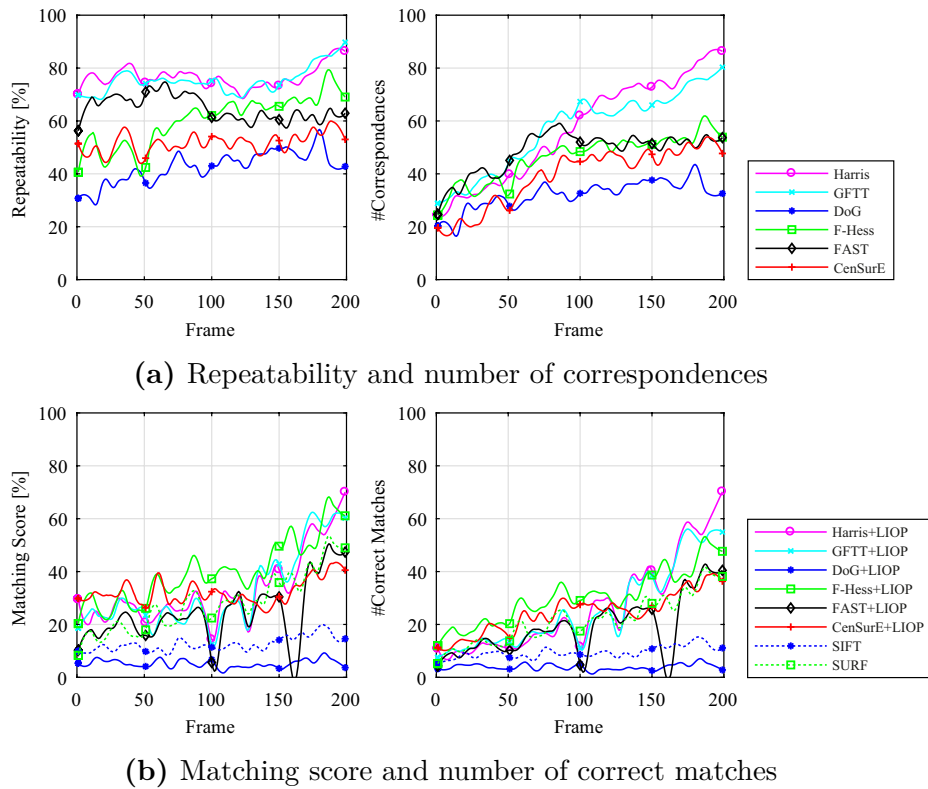
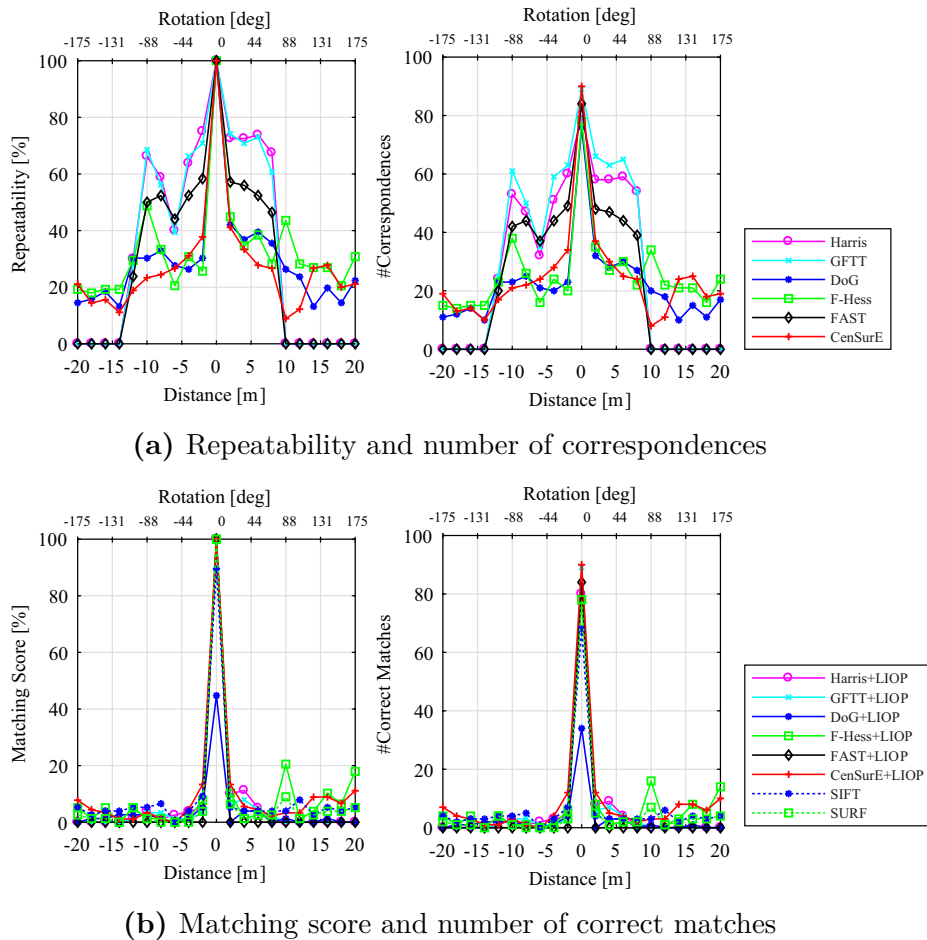


Figure 3.19: Performance for ASTOS-B/02/02 rendezvous sequence: successive transformations, thermal infrared band, cold case. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

better approach in this case according to the results.

Regarding the benchmarking of large transformations, the results show a quasi-symmetrical pattern around the baseline. The matching scores are generally biased towards the right, meaning that larger scales (shorter distances between chaser and target) are favourable. This is a judicious hypothesis since, due to the low resolution of the dataset, bigger distances quickly translate into less details. On the other hand, there is a bias towards the left in repeatability, which is explained by the fact that smaller scales with a constant region size lead to more overlaps. The lack of scale invariance in the corner detectors is evident from the abrupt decline of the associated number of matches when varying the distance to the target. In general, the performance of the detectors is quite low for large baselines as opposed to successive transformations, which can make their use difficult in model-based pose estimation pipelines.



(a) Repeatability and number of correspondences

(b) Matching score and number of correct matches

Figure 3.20: Performance for ASTOS-B/02/02 rendezvous sequence: large transformations, thermal infrared band, cold case. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

3.4.5.3 Benchmarking of Feature Descriptors

For this test, the performance of the descriptors is assessed. To this end, a comparison is done using the same feature detector for all the descriptors in order to reduce the influence of the former on the results. Similar settings as in the previous experiments were used, i.e. an error threshold of 30% and a fixed number of 75 extracted features. The regions are not normalised in the computation of the descriptors.

The efficiency of the algorithms is evaluated by computing their ROC, or recall/1-precision, curves. For each of the four sequences, and similarly to Mouats et al. (2018), two sets of results are shown: the first representing a descriptor benchmark for short (successive) and large image transformations using the DoG detector; and the second repeats the same experiment using Fast-Hessian. This allows insight into if and how different detector-descriptor combinations affect the outcomes. These are plotted in Figures (3.21) to (3.24).

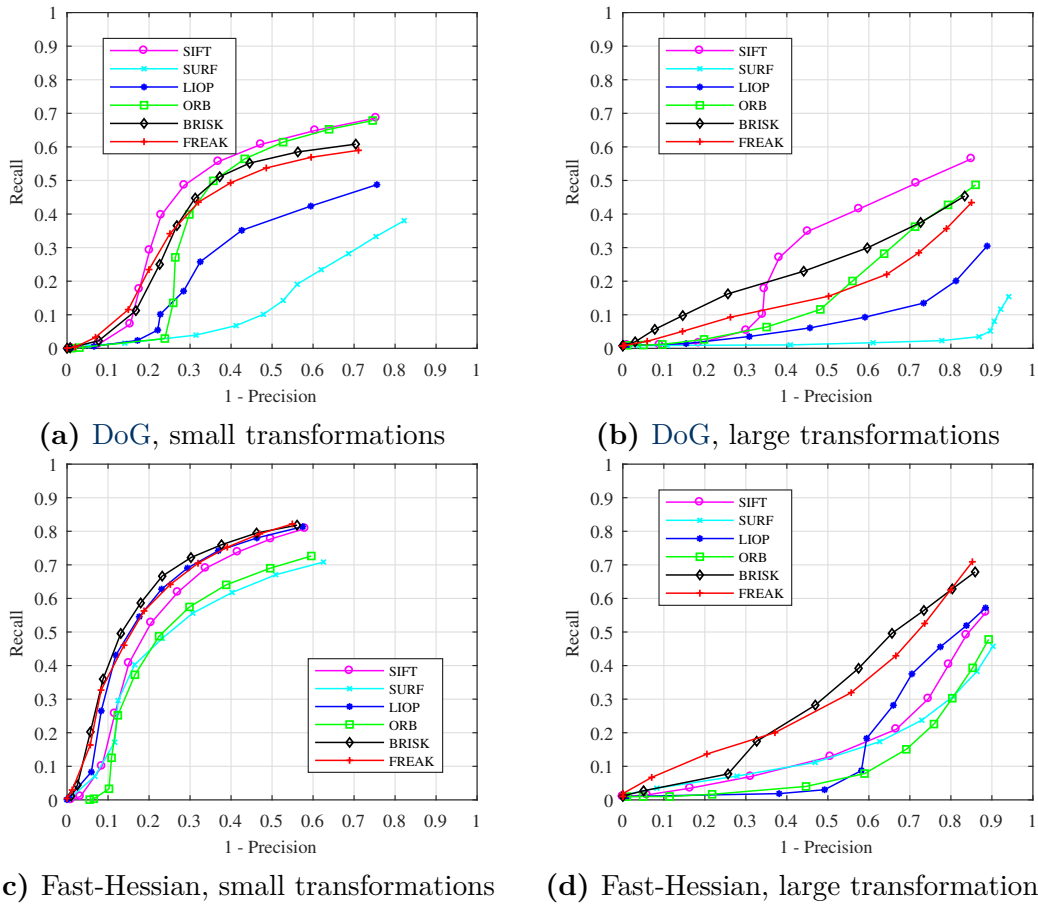


Figure 3.21: Descriptor ROC curves for ASTOS-B/01/01 rendezvous sequence: visible band, hot case. (*Top Row*) With DoG as feature detector. (*Bottom Row*) With Fast-Hessian as feature detector. (*Left Column*) For small image transformations. (*Right Column*) For large image transformations.

However, a different procedure is adopted regarding which frames from the dataset are used. Mouats et al. (2018) consider only the matching between features from two frames (one pair with a short baseline, another pair with a large one) for this test. This is because the authors benchmark image transform variations in an isolated way, i.e. one test for rotation variation, one for scale change, and so on. For the analysis presented within this chapter, the ASTOS-B sequences have in common a fixed trajectory where more than one transform is present. Since the aim is to assess the performance for the whole rendezvous manoeuvre, the ROC curves are computed using the average values for every pair of frames; in particular, for the large transformations set, the reference used is a frame located at the middle point of the sequence, i.e. when the target is 60 m away from the chaser, and the test includes variations in the range of ± 20 m/ ± 175 deg relative to the reference.

As mentioned in Section 3.3, a NNDR-based matching strategy is considered.

Visible Modality Hot Case Figure 3.21 illustrates the attained ROC curves for the visible modality during the sunlight period. It can be seen that the performance of the descriptors depends on the feature detection algorithm used: Fast-Hessian features are shown to yield better precision. It can also be seen that the performance of the algorithms is degraded for large transformations comparatively to sequential ones.

It is interesting to note that SIFT performs better with Fast-Hessian features (Figure 3.21c) than with DoG features (Figure 3.21a) in the case of short transformations. However, the opposite is true for large transformations (Figures 3.21b and 3.21d). Indeed, when DoG features are used, SIFT performs best, followed by ORB and BRISK. For small transformations, the performance of the three descriptors are comparable, whereas for large transformations, BRISK obtains the best results if $1 - \text{precision} < 0.35$ but SIFT dominates for values above that.

For Fast-Hessian features, BRISK, FREAK, and LIOP give the best results in the case of small variations; in the case of large variations the performance of the latter one degrades considerably, which seems to agree with the observations of Mouats et al. (2018) regarding the monotonic intensity changes of LIOP’s rotation invariant sampling not holding for large angles. Overall the results obtained for SURF are sub-par, showing that combining a feature detector with a non-native descriptor can yield better results.

Visible Modality Cold Case Figure 3.22 shows the descriptors’ performance for the visible in eclipse. The algorithms are affected by the low illumination case more than the sunlit scenario for this spectrum. The precision can be shown to be relatively lower, particularly for larger variations, which means the descriptors incur more frequently in false matches. The relative ranking of the algorithms is similar to the previous case, save for small variations computed on Fast-Hessian features, where SIFT shows the best performance (close to FREAK and BRISK) and LIOP performs the worst. This is in agreement with the plot of Figure 3.15, where there is a drop in the matching score of Fast-Hessian + LIOP, but it still remains higher than that of SIFT.

Thermal Infrared Modality Hot Case Here, the descriptors are compared for the case of the thermal infrared imaging of the sequence during sunlight conditions; the results are shown in Figure 3.23. The performance computed on DoG features follows the same trend as for the visible case, albeit with a yielded precision lower than the eclipse case.

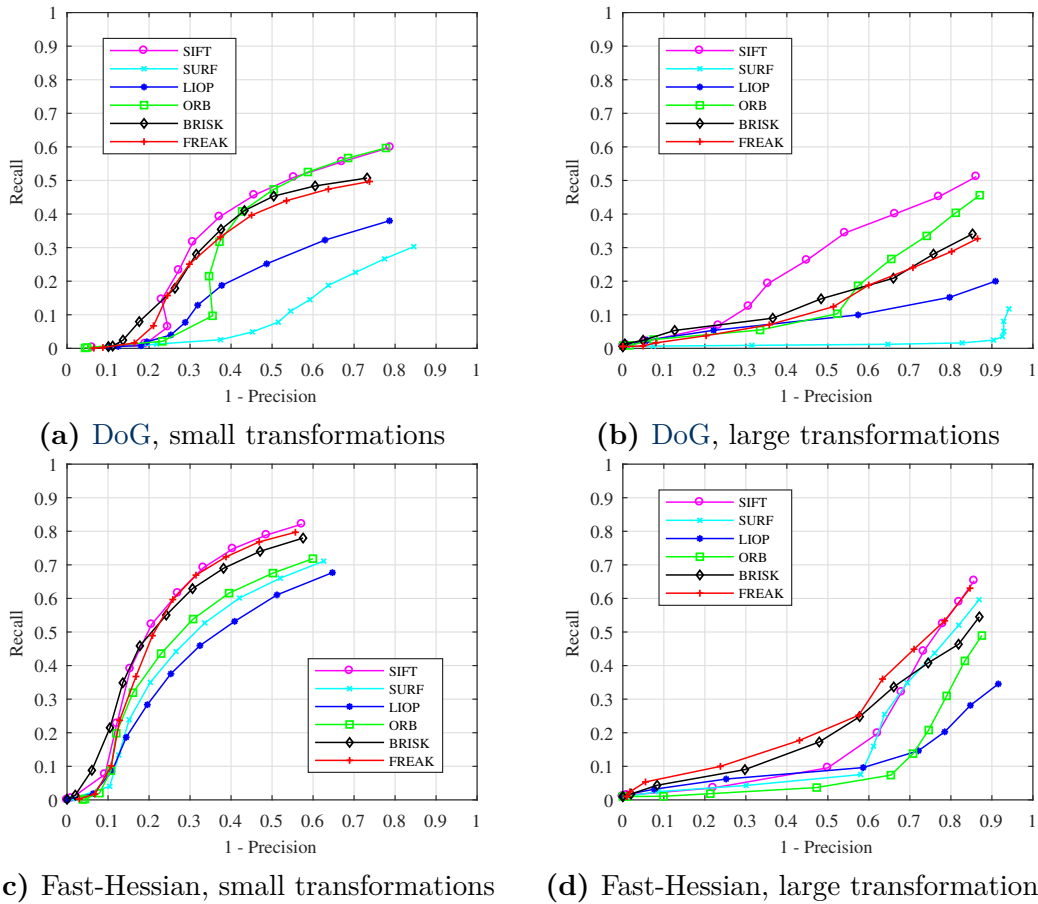


Figure 3.22: Descriptor ROC curves for ASTOS-B/02/01 rendezvous sequence: visible band, cold case. (*Top Row*) With DoG as feature detector. (*Bottom Row*) With Fast-Hessian as feature detector. (*Left Column*) For small image transformations. (*Right Column*) For large image transformations.

On the other hand, when using Fast-Hessian features the descriptors perform better than both visible cases. For short transform variations, **FREAK** obtains the higher score, but as in the analogous visible case, it behaves quite similarly to **BRISK**, **SIFT**, and **LIOP**. With respect to larger transformations, **FREAK** performs best by a large margin. The other algorithms are also less affected by these variations than in the visible case. This means that, for the same relative motion, the descriptors are more affected by the dynamic effects present in the visible modality — such as textural noise, glare, shadows — than by a textureless scene.

Thermal Infrared Modality Cold Case Lastly, Figure 3.24 illustrates the benchmarking of the descriptors in the eclipse case for the thermal infrared sequence. As in the visible case, the algorithms are more affected by these transformations than in the hot case.

When DoG features are used, the descriptors perform worse than in the visible

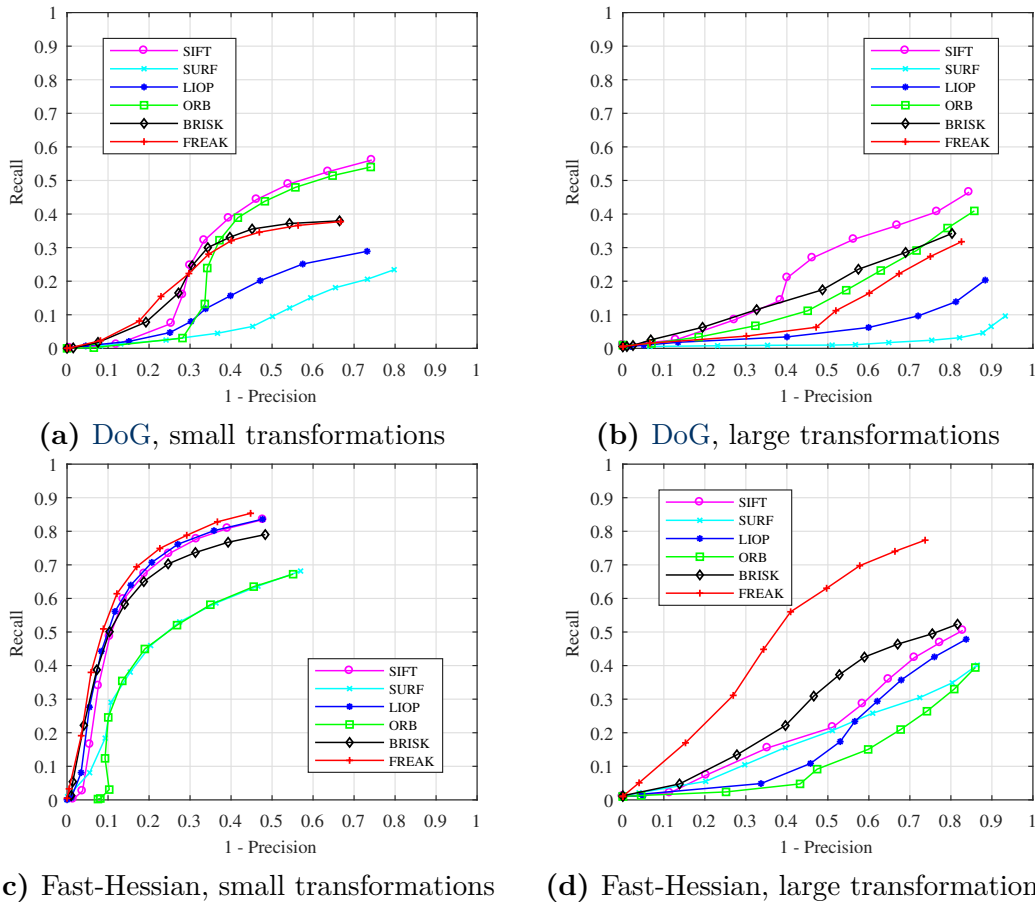


Figure 3.23: Descriptor ROC curves for ASTOS-B/01/02 rendezvous sequence: thermal infrared band, hot case. (*Top Row*) With DoG as feature detector. (*Bottom Row*) With Fast-Hessian as feature detector. (*Left Column*) For small image transformations. (*Right Column*) For large image transformations.

cold case. The precision attained by the algorithms is quite low, which is in line with the observations from Section 3.4.5.2 regarding the low number of correct matches for this detector in the thermal infrared modality.

Conversely, descriptors computed on Fast-Hessian features in this scenario are actually comparable to the performance attained for the visible hot case; for small transformations, LIOP achieves the best performance, however it is again degraded in the case of larger transform variations.

Discussion

The presented results suggest that the performance of the descriptors is dependent on the feature they are applied on, regardless of descriptor type. Fast-Hessian performs better in general both in terms of recall and precision scores, regardless of the modality, although the gap is narrower in the benchmarking of large transformations. As theorised by Mouats et al. (2018), a possible explanation for this could be the fact

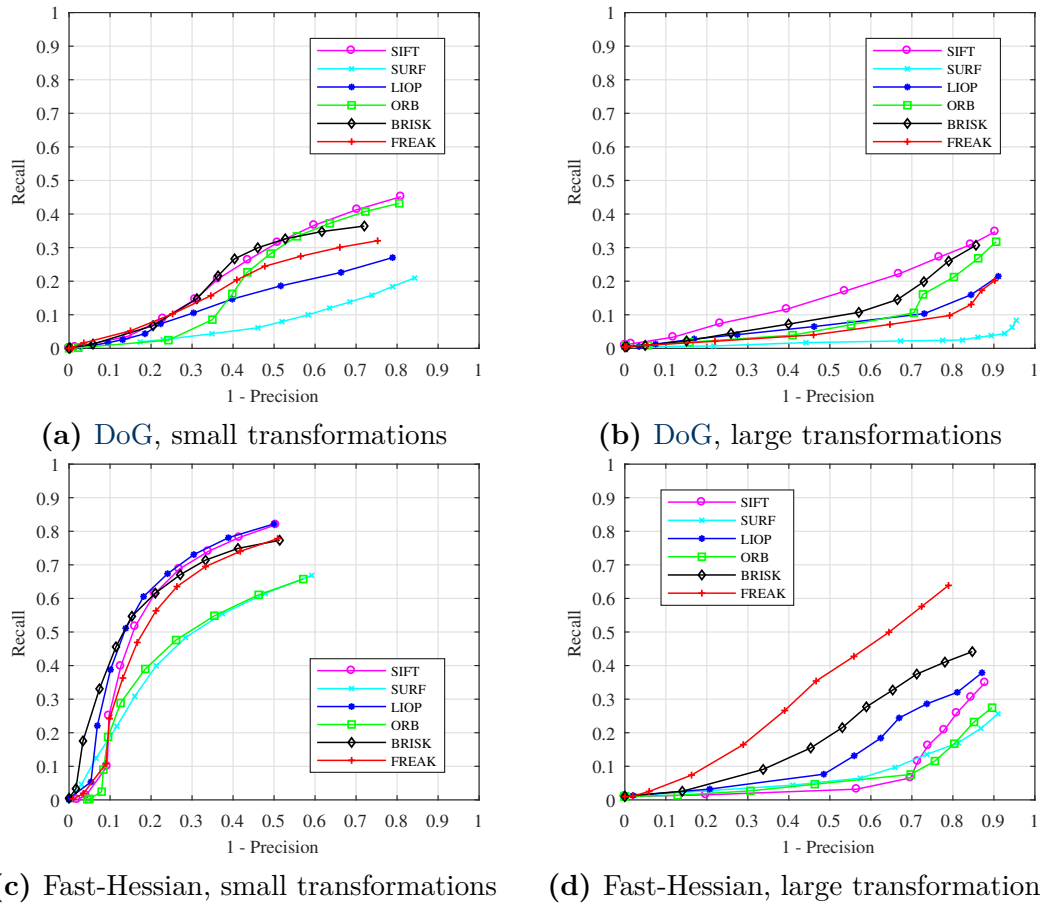


Figure 3.24: Descriptor ROC curves for ASTOS-B/02/02 rendezvous sequence: thermal infrared band, cold case. (*Top Row*) With DoG as feature detector. (*Bottom Row*) With Fast-Hessian as feature detector. (*Left Column*) For small image transformations. (*Right Column*) For large image transformations.

that Fast-Hessian usually extracts larger blobs than DoG, so a larger support area is considered in the computation of the descriptor, capturing in principle a larger signal variation. In can be seen by inspecting Figure 3.9 that this is also the case for the analysed dataset.

Overall, SIFT as a whole obtained very good scores. However, its performance is degraded substantially in the case of large transformations (particularly on Fast-Hessian features).

LIOP was shown to perform better when computed on Fast-Hessian features, both on the visible, and as reported by Mouats et al. (2018) on the LWIR. It can be ranked amongst the best descriptors when used with this type of feature for successive transformations. The exception is the visible cold case, where it is ranked last. Furthermore, when considering large transforms, its performance declines, which is in line with the analysis made for the detectors in Section 3.4.5.2

Table 3.5: Average detection times per feature.

Detector	Time (ms)	Speed-up
FAST	0.03	814
CenSurE	1.32	16
Harris	1.42	15
GFTT	1.49	14
F-Hess	2.63	8
DoG	21.22	1

Table 3.6: Average description times per feature.

Descriptor	Time (ms)	Speed-up
ORB	0.16	103
BRISK	0.21	81
SURF	0.77	22
SIFT	9.48	2
LIOP	14.54	1
FREAK	16.73	1

Overall, BRISK and FREAK are ranked among the best descriptors for all cases.

3.4.5.4 Computation Times

In this subsection, the IP algorithms are benchmarked in terms of their computational performance. These tests are ran on the single board computer setup, allowing for the examination of their real-time capacity on a low performance embedded system. The recorded benchmarks account only for the core tasks of detection or description. All values are averaged between the four sequences for each algorithm.

Table 3.5 portrays the average extraction time per feature for each detector. This type of analysis is useful in shifting awareness towards the computation time, which can be limiting depending on the application, and is particularly important for those involving low performance computing. DoG scores the slowest detection time, at 21.2 ms per feature. To better compare their performance, in addition to the absolute computation times, the relative speed-up factors with respect to the heaviest algorithm are also displayed. FAST is the quickest algorithm to run, being almost three orders of magnitude swifter than DoG. As expected, CenSurE is faster than Fast-Hessian, which is in turn faster than DoG. Surprisingly, GFTT is recorded having a higher execution time than Harris.

Figure 3.25 displays the average computation times of the detectors per frame. DoG is the clear outlier, being the only detector that does not fit in the computational

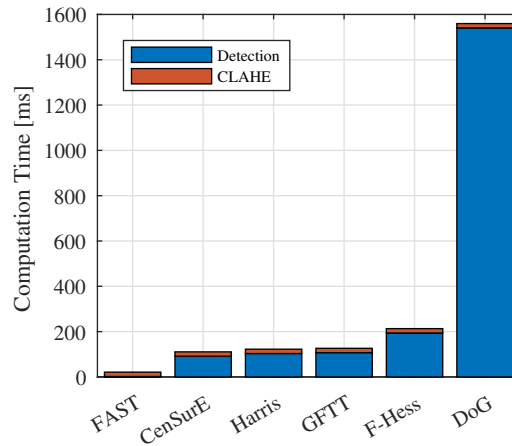


Figure 3.25: Comparison of average feature detection times per frame.

budget of 1 Hz. In addition, the average execution time per frame for **CLAHE** in the case of the visible cold case sequence are also shown (in red). This function does not depend on the detector and the mean execution time was 19.22 ms, accounting for less than 2% of the allocated budget. Note, however, that the average detection time per frame of **FAST** was 1.9 ms, which is faster than the preprocessing step by a factor of 10.

Analogously, Table 3.6 shows the benchmarked computation times for the descriptors averaged per feature. While the list is topped by two of the binary descriptors, **FREAK** is actually the slowest algorithm, costing 16.7 ms per feature on average. The high computation time is unusual for a binary descriptor and contradicts the findings in the literature. **LIOP** is similar in performance, while **SIFT** is two times faster. Surprisingly, the performance of **SURF** is in the same order of magnitude as **ORB** and **BRISK**.

Figure 3.26 illustrates the average computation times of the descriptors per frame. The matching times are represented in purple. As expected, the matching times for the binary descriptors are the fastest, scoring an average of 2.5 ms per frame (75 features). **ORB** features are the fastest to be matched at an average of 1.9 ms per frame. The distribution-based descriptors are on average one order of magnitude slower in terms of matching speed, at 24 ms; **SURF** features are the fastest of the kind, scoring 14.5 ms on average.

FREAK Results

Given that the previous experiments recorded abnormally high execution times for the **FREAK** descriptor, the benchmarks are repeated, this time on a desktop workstation with an Intel(R) Core(TM) i7-6700 processor (x64) at 3.40 GHz. Figure 3.27 compares the speed-up times for the six descriptors and two processors relative to the heaviest

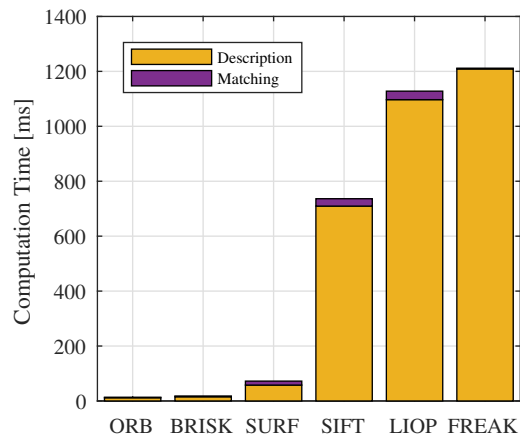


Figure 3.26: Comparison of average feature description and matching times per frame.

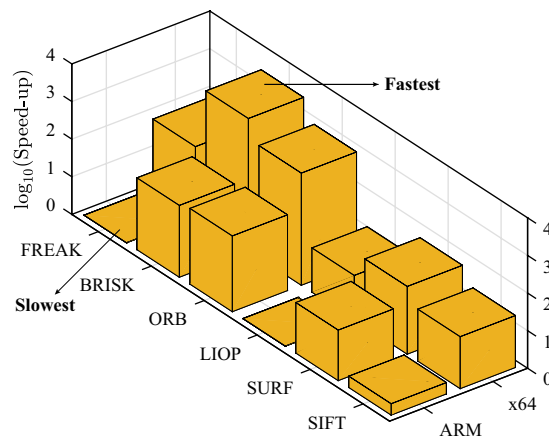


Figure 3.27: Comparison of descriptor speed-up factors in different processors.

test run. It can be seen that the relative ranking of the algorithm changes for the x64 processor, where **FREAK** totals as the third fastest descriptor. It is almost two orders of magnitude faster than **LIOP**, whereas the execution times are identical on the ARM processor. The relative ranking of the distribution-based descriptors is the same on both processors, and they maintain approximately the same proportions in terms of runtime. However, **BRISK** is the fastest running descriptor for the x64 processor, being 243% faster than **ORB**; for the ARM processor it was 21% slower. This seems to suggest implementation issues in the case of the binary descriptors, i.e. the algorithms are optimised differently depending on the architecture.

3.5 Conclusions and Future Work

In this chapter, several state-of-the-art feature detectors and descriptors have been benchmarked in the context of an *ADR* application. To this end, a custom synthetic dataset featuring a rendezvous with the defunct spacecraft *Envisat* was created. This dataset, derived from the work presented in Chapter 2, encompasses two different trajectories, one during a sunlight period and one during eclipse, imaged in two different modalities, the visible and the *LWIR*, yielding four different scenarios. The performance of the *IP* algorithms has then been benchmarked for these scenarios, providing a multimodal evaluation of the low-level processes in computer vision required for a further integration in a vision-based navigation system.

The presented benchmarks have shown that features in the *LWIR* domain are generally more repeatable than in the visible. In terms of matching score, the difference between the two modalities is smaller when the target appears small in the *FOV* of the camera, and greater at shorter distances in favour of the *LWIR*, meaning that the shadows and noise in the visible become more noticeable and the algorithms become more sensitive, which could be a limitation of this modality for short ranges. Conversely, this could mean an advantage of using *LWIR* imaging as a way to bypass the difficulties of optical navigation relative to a complex spacecraft bearing non-imaging-friendly components such as *MLI*.

In terms of the analysis of feature descriptors, it was found that the performance depends on the type of feature used: when *DoG* features were used, the performance is better on the visible, but the performance became better on the *LWIR* with Fast-Hessian features. The latter were shown to be larger in radius than the former, hence capturing a larger support region.

The results also shown the advantage of thermal imaging in eclipse sequences. Using visible imaging, all detectors have shown a decline in matching score, and the benchmarking of the descriptors resulted in a lower number of matches and elevated false positives. Regardless of the sequence, the *IP* algorithms have performed substantially worse when testing large baseline transformations, which could hinder the development of model-based visual navigation pipelines when only feature points are used.

With respect to computation times, it was found that, for a fixed number of 75 features per frame, only one of the detectors (*DoG*) and two of the descriptors (*LIOP*, *FREAK*) exceed the computation budget of 1000 ms. *FAST* has shown the largest speed-up factor (814) with respect to the traditional *DoG*, and in general the corner detectors were faster to compute than the blob detectors. As expected, the binary descriptors (*ORB*, *BRISK*) demonstrated lower running times with respect to

SURF, SIFT, LIOP; the exception was FREAK, although its large processing time was subsequently shown to be related to its current OpenCV implementation in the ARM architecture.

The benchmarks have additionally provided an interesting insight into the state-of-the-art baseline algorithms such as SIFT and SURF. The latter, for instance, provided higher scores with Fast-Hessian features than with its native detector, DoG. In general, the results have motivated combining different detectors and descriptors to boost performance. Overall, a combination of Fast-Hessian with FREAK is capable of providing adequate performance for a vision-based navigation in the context of ADR. However, it is currently compromised by its current implementation in the low-performance ARM processor. Fast-Hessian + BRISK offers similar performance and is computationally efficient, as it was shown to run inside the boundaries of the considered low acquisition frame-rate, taking up slightly over 20% of the computational budget, leaving the remaining 80% open for the relative pose estimation tasks. Furthermore, the benchmark of Fast-Hessian + BRISK is comparable in both spectra, meaning it could potentially be used for a multimodal navigation algorithm, analysing a frame of each modality per cycle, and it would still perform the detection and description tasks in less than half of the budget with lower memory usage.

Given the conducted analysis, it should be noted that other detector/descriptor combinations that comply with the hardware requirements are possible. Recommendations for future work would include additional experimentation with algorithms besides the ones tested herein, e.g. ORB with its native descriptor. An additional direction to follow would involve an investigation of the improvement of the performance of IP algorithms for model-based navigation. Lastly, a further future research avenue could consist in analysing the embedded board performance when running the full navigation algorithms, developed in Chapters 4 and 5, that make use of feature detectors and descriptors.

3. BENCH'ING OF DETECTORS AND DESCRIPTORS FOR NAV'N

CHAPTER 4

Markerless Multi-View Monocular Pose Estimation

In this chapter, a method of estimating the pose of a non-cooperative target for spacecraft rendezvous applications employing exclusively a monocular camera and a three-dimensional model of the target is proposed. This model is processed to build an offline database of pre-rendered keyframes with known poses. Then, an online stage solves the model-to-image registration problem by matching two-dimensional point and edge features, detected by the camera, to the database. The combination of these two types of features is analysed in terms of robustness for large keyframe displacements and computational efficiency.

4.1 Motivation

ACTIVE navigation sensors, such as lidar, have traditionally been the “go-to” tool for proximity operations in space (J. Christian et al., 2011) by having the advantage of being invariant to illumination changes and supplying range information. The latter cannot be intrinsically recovered by a camera (unless paired with a second one), and the basis of relative motion estimation for camera-based systems is rather the extraction of two-dimensional features from the captured image of the scene. As monocular systems slowly but surely gain the necessary traction to replace lidar due to their attractive budgeting properties, wide-scale image processing (IP) techniques are being explored to complement them.

Such techniques for pose estimation can be categorised into two main classes: model-free and model-based. Model-free methods do not require previous knowledge of the scene or target at hand, working by jointly estimating the camera’s motion

through the unknown environment and a mapping of it, in a process termed visual simultaneous localisation and mapping (VSLAM; Karlsson et al., 2005). For some platforms, however, the overall uncertainty of the estimation could inordinately grow in the event of rapidly-changing imaging conditions. In these cases, it can be advantageous to exploit existing a priori information about the three-dimensional structure of the scene. If a model of it exists, the camera motion can be evaluated by matching the 2D features from the image with the 3D reference points. Then, the camera can be localised with respect to the scene in terms of the position and attitude that yield the best alignment between these matches, effectively solving the model-to-image registration problem. This notion is naturally extensible to spacecraft relative pose estimation: the scene becomes reduced to a single target body (or potentially multiple, in the case of formation flying missions), with which the frame of reference translates and rotates accordingly.

Furthermore, the nature of this model is directly related to the level of cooperation of the mission: a target that is fitted with specialised artificial markers for model-based solutions falls under cooperative rendezvous. These markers are placed at known points on the surface of the target and facilitate detection by the camera system through properties such as light emission (Junkins et al., 1999) or known geometric patterns (G. Zhang et al., 2016). Conversely, non-cooperative rendezvous (NCRV), by definition, does not rely on marker-based systems. In this case, a model of the target can be provided in the form of a computer-aided design (CAD) that provides textural and structural information with relatively high fidelity, which is a reasonable assumption to make for manufactured objects such as satellites.

This chapter aims to investigate the feasibility of a model-based approach for non-cooperative spacecraft relative pose estimation. The rationale is that, by shifting most of the computational burden to an offline training phase, the advantages of model-based estimation can be taken advantage of while bypassing the need for complex rendering hardware. The limitations of point matching for wide baselines are combatted with the introduction of edges. The iteratively reweighed least squares (IRLS) formulation is utilised to efficiently combine both types of features based on the estimated quality of the matching. Furthermore, it is shown how the covariance of the IRLS solution can be used to reset the pose estimate in case of convergence towards a local minimum. The proposed framework is tested on Simplesat, a modified spacecraft model based on the design of Envisat where the size of each module has been made more balanced and dynamic lighting has been removed. The goal is to allow for a preliminary benchmark of the developed methods in terms of the tumbling motion alone.

Remark 4.1: Associated Publications

This chapter is based partly on the following published work:

- [C1] D. Rondao and N. Aouf (Jan. 2018). “Multi-View Monocular Pose Estimation for Spacecraft Relative Navigation”. In: *2018 AIAA Guidance, Navigation, and Control Conference*. Kissimmee, FL: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2018-2100

4.2 Related Work

Recent work has shown [VSLAM](#) to be executable in real-time with good performance using keypoint detectors and tracking features from frame to frame (Mur-Artal et al., 2015) or even the raw image pixel intensities themselves (Engel, Schöps, et al., 2014). [VSLAM](#) has also been demonstrated to be applicable to relative navigation with an artificial target under certain conditions; however, developments for monocular systems lack validation either on space environments (Augenstein and Rock, 2009) or on tumbling cases (Dor and Tsiotras, 2018). Furthermore, a considerable disadvantage is that the scale of the estimated trajectory cannot be recovered: [the methodology is reliant on the triangulation of features matched at different time-steps, but the geometrical relationship between them obeys the epipolar constraint, of which the essential matrix is a homogeneous quantity and hence only has five degree-of-freedom \(DOF\) rather than six \(Hartley and Zisserman, 2004\).](#)

Contrarily, model-based approaches assume that information about the three-dimensional structure of the target is known a priori. In this case, IP algorithms are employed to solve the model-to-image registration problem, i.e. the coupled pose and correspondence problems, the latter which consists in establishing matches between the target’s 3D structural information and the 2D features obtained by the camera (i.e. perspective- n -point [PnP], thus avoiding the aforementioned issue of scale ambiguity), and which is often overlooked in pure guidance, navigation and control (GNC) literature. In the circumstances where the target is artificial, such as in active debris removal (ADR), on-orbit servicing, or docking, it is justifiable to assume that its structure, or at least part of it, is known.

Within this approach, two paths can be followed for non-cooperative systems (Lepetit and Fua, 2005): tracking by recursion or tracking by detection. For the former, the system is initialised with a pose estimate and propagated by tracking features from one frame to the next. A prominent technique in this category is edge-based tracking (Drummond and Cipolla, 2002) of industrial CAD models. It

is assumed that the camera motion between frames is limited such that sampled 3D control points from these models are reprojected onto the image plane using an expected pose accompanied by a one-dimensional scan to locate the corresponding edge on the feature space. Additional control points can be rendered as the found edge is subsequently tracked to the next frame. The motion estimation problem is cast in terms of a Lie group formalism to recover the pose based on the minimisation of the distance between corresponding features. This method has been successfully applied to spacecraft pose estimation first by Kelsey et al. (2006) using a wireframe model of the target. It would later form the basis for the Goddard Natural Feature Image Recognition (GNFIR) algorithm, which was tested in-orbit during the SM4 mission to the Hubble Space Telescope (HST; Naasz et al., 2010). Neither application tackled uncontrolled tumbling of the target.

Shortly after, Comport et al. (2006) introduced the virtual visual servoing (VVS) pipeline, which re-purposed the problem in terms of a visual servoing control law, in which the goal is to move a camera to observe a given object at a given position in the image by minimising the error between the desired state of the image features and the current state. This allowed the extension of Drummond and Cipolla’s (2002) method towards the tracking of model straight lines, circles, cylinders, and spheres through the definition of specific “interaction matrices” for each geometric primitive. Later on, Petit et al. (2013, 2014) upgraded the VVS pipeline to include information from colour and point features for tracking. It also used hardware acceleration based on graphics processing units (GPU) for the real-time rendering of the target spacecraft’s model, allowing it to tackle the tumbling problem. On the other hand, this represents a significant drawback as it makes an implementation on current flight-ready hardware unlikely. Nevertheless, advances in model-based methods for the past decade have continued to focus on feature tracking either by disregarding the initialisation stage (Zou et al., 2016) or by assuming that the chaser images the scene from a constant viewpoint (Cai et al., 2015; Gansmann et al., 2017; Oumer, 2014), limiting their use for rendezvous when prior knowledge of the target’s attitude is not known.

Tracking by detection consists in matching 2D image features to a database of training features that have been pre-computed offline. In the case of three-dimensional targets, this database is often obtained from a set of rendered viewpoints of the object, i.e. keyframes (Vacchetti et al., 2003, 2004). Despite the popularity of tracking methods for space applications, there have been proposals to apply detection methods to the problem: a thorough search of the relevant literature yielded Cropp’s (2001) doctoral thesis as the first of this kind for the recovery of the full 6-DOF

pose of a spacecraft, where, based on Dhome et al.'s (1989) and Lowe's (1991) work, pre-generated 3D edge features of a model of the target were matched to detected 2D image edges to retrieve the pose. Textureless features such as edges (Abderrahim et al., 2005) or ellipses (C. Liu and Hu, 2014) gained popularity due to their robustness, but the matching process is often complex and relies on multiple hypotheses and long convergence times. This has recently shifted the focus towards point features, which can be efficiently matched using descriptors. The challenge in this case is in achieving robustness between test and train images (J.-F. Shi, Ulrich, and Ruel, 2016), as these are often dissimilar in terms of baseline and illumination conditions; this has been alternatively tackled by in situ keypoint triangulation to shift the problem towards 3D descriptors (Post and J. Li, 2018).

The method introduced in this chapter takes inspiration from the work of Vacchetti et al. (2003, 2004), who proposed the first tracking by detection system using more than two keyframes, but with some important developments:

- (1) Feature point matching is performed by resorting to invariant detectors and descriptors (Chap. 3), as opposed to template matching supported by planar homographies; and
- (2) The solution is stabilised by relying only on the reprojected contour edges of the model, rather than the full depth map.

The importance of these points for space systems is reflected on the fact that they eliminate the need for specialised hardware for the rendering of complex models such as in (Petit et al., 2013) or (Zou et al., 2016).

4.3 Methodology

The goal of the proposed method is to robustly estimate the camera pose relative to a tumbling target solely from reference keyframes. The keyframes are generated offline based on a CAD model and depict the target under different viewpoints and consist of a textural pass and a depth map pass. The former is used to extract 2D features and the latter to annotate them with 3D information. All the necessary renderings are confined to this stage, dramatically reducing the computational requirements for the online stage, where the structural information is conveyed by 2D-2D matching of detected features to the current keyframe. An IRLS scheme is used to combine information from the matching of the two used feature types and to recover the pose. The next keyframe is picked based on this estimate. Figure 4.1 depicts the high level structure of the method. Section 4.3.1 summarises the used feature

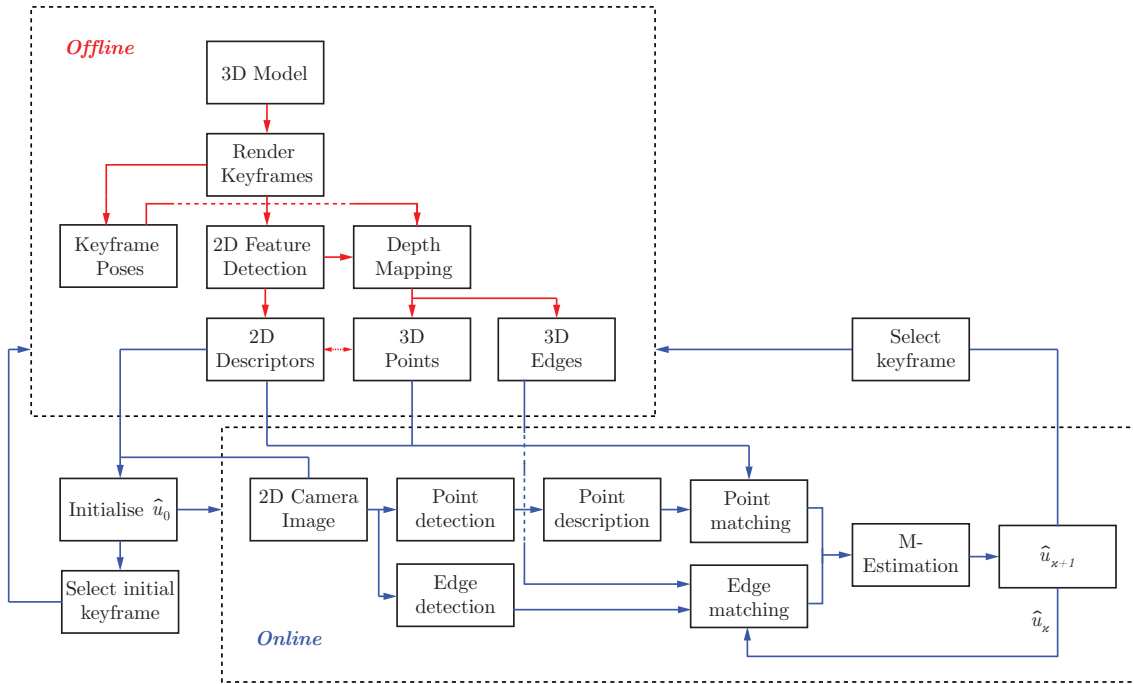


Figure 4.1: High level view of the proposed method. A three-dimensional model is used to learn the structure of the target and to generate keyframes annotated with two-dimensional features in an offline stage (*red*). During the online stage, detected features on the target are matched to the current keyframe (*blue*). The relative camera pose is recovered from the 2D-3D association.

detection and matching strategies. Section 4.3.2 reviews the IRLS method for pose estimation. Lastly, Section 4.3.1 describes the complete algorithm.

4.3.1 Local Feature Detection and Matching

The model-to-image registration problem can traditionally be solved exclusively using 2D-3D point correspondences. In this case, the problem is specifically termed PnP . Notwithstanding the many different approaches in the literature towards its solution (e.g. Lepetit, Moreno-Noguer, et al., 2008, Kneip et al., 2014, or Ferraz et al., 2014), point-based features are not free from drawbacks, and a relative motion solution would benefit from combining different types of features. For the purpose of this work, point and edge features are combined to estimate the pose.

4.3.1.1 Feature Point Detection in Scale-Space and Binary Description

The advent of the early feature point detectors (Harris and Stephens, 1988) did not bring about any clear-cut method to match them. Patches of pixels centred on the corner (templates) would be extracted and matched with others based on the minimisation of their sum of squared differences (template matching; Szeliski, 2011,

Chap. 8). This meant that corner matching was limited to differences in translation, admitting only minute perspective distortion between templates. Approaches such as the one by Vacchetti et al. (2003) have tackled this issue by registering the local surface normal of each template and then warping it according to a plane-induced homography recovered from the previously estimated pose.

The present method proposes instead to take advantage of the state-of-the-art keypoint detectors and descriptors benchmarked in Chapter 3 which carry an innate invariance (up to a limit) towards wide baseline matching. Based on the positive matching score results obtained for the visible wavelength and its ability to extract distinctive features, interest points are extracted using the Fast-Hessian detector (Bay et al., 2006), which approximates the Laplacian of Gaussian (LoG) operator with box filters, granting it a speed factor of $8 \times$ when compared to difference of Gaussians (DoG; Lowe, 2004).

In the same manner, keypoint information is encoded into binary strings using the Fast Retina Keypoint (FREAK) descriptor (Alahi et al., 2012), which was shown to perform best for visible sequences in combination with Fast-Hessian. Furthermore, the matching performed through Hamming distance minimisation makes this operation fundamentally faster than most of its Euclidean distance-based counterparts.

4.3.1.2 Edge Detection with False Positive Control

The results from Chapter 3 suggest that state-of-the-art keypoint detectors and descriptors suffer from some performance degradation during rendezvous sequences when considerable transformations are experienced. To further robustify the approach, the inclusion of edge features is proposed. Effectively, the former are not always impervious to illumination changes, and matching failure due to partial occlusion is a possibility. In contrast, the latter are typically less distinctive but carry an extra degree of robustness by showing stability towards such conditions (Lepetit and Fua, 2005).

Standard IP edge detection techniques that rely almost exclusively on the gradient computation of the intensity image are greatly affected by noise and rich, dense textures. Approaches such as Canny's (1987) introduce non-maximum suppression and hysteresis thresholds to combat this at the expense of losing detail. Instead, one step farther is taken and line segment features are computed for test images with the Edge Drawing Lines (EDLines) detector (Akinlar and Topal, 2011). The algorithm is divided into three main steps. First, the edge drawing method is applied: the greyscale input frame is filtered to remove noise and the gradient magnitude and orientation are computed at each pixel; peaks in this gradient map are marked

as anchors due to their high probability of being edge elements; anchors are then connected by drawing edges between them. Secondly, line segments are extracted from the generated anchor chains using a least squares line fitting method. Lastly, the segments are subject to a validation process: for each line segment, the gradient orientation is computed for each pixel along it to assess the number of aligned pixels. The number of false alarms for the segment is then evaluated as

$$\text{NFA}(n, k) = W^2 H^2 \sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i}, \quad (4.1)$$

where n is the length of the segment, k is the number of aligned pixels, $p = 0.125$ is the accuracy of the line direction (the alignment computation is discretised into 8 bins), W, H are the width and height of the image, respectively, and the brackets signify the binomial coefficient. The line segment is accepted as valid if $\text{NFA} \leq 1$.

EDLines is tested in its inception manuscript against classic line feature detection algorithms such as Canny combined with the Hough transform (Duda and Hart, 1972), where more accurate, well-localised edges with considerably less false detections are obtained while simultaneously reducing the computational effort.

Initial experimentation with state-of-the-art line segment feature descriptors (e.g. line band descriptor; L. Zhang and Koch, 2013) for a matching strategy analogous to point features did not produce satisfactory results, even for cases of very small baselines. As such an alternative strategy was adopted, similar in spirit to Drummond and Cipolla’s (2002), where the keyframe model edge features are projected onto the image using the current pose estimate, and then matched to the detected **EDLines** features by searching along the projected local normal vector, $\mathbf{n}^{(i)}$. As this approach can be prone to ambiguous matches, only the model edges corresponding to the target’s contour (i.e. the “limb”) are kept in the database. This also provides a natural way of working with complex spacecraft models without the need of explicitly modifying the **CAD**.

4.3.2 Iteratively Reweighed Least Squares Minimisation

Given a frame of reference \mathcal{F}_t attached to the target and the camera frame of reference \mathcal{F}_c connected to the chaser, the goal is to find the pose matrix $\mathbf{T}_{ct} := \mathbf{T}$ that maps \mathcal{F}_t to \mathcal{F}_c . The Gold Standard algorithm (Hartley and Zisserman, 2004) to recover an estimate $\hat{\mathbf{T}}$ from 2D-3D correspondences can be shown to be optimal in the maximum likelihood sense, but it cannot inherently handle the presence of spurious matches, i.e. outliers. Rather, the standard formulation is augmented with a robust loss function ρ as:

$$\hat{\mathbf{T}} = \arg \min_{\mathbf{T}} \sum_{i=1}^n \rho \left(\frac{r_i(\mathbf{T})}{\sigma_i} \right), \quad (4.2)$$

where $r_i(\mathbf{T}) := r_i$ and σ_i^2 are the residual and variance of the i th match, respectively, and n is the total number of matches. A robust $\rho(x)$ is one with a bounded influence function $\psi(x) := d\rho(x)/dx$, in which case it is termed an M-estimator. Analogously, $\hat{\mathbf{T}}$ becomes an M-estimate of \mathbf{T} . The scale σ_i can be estimated from the data points by assuming homoscedasticity (i.e. assumption of equal variances) and computing the median absolute deviation (Stewart, 1999):

$$\hat{\sigma} = \frac{1}{\Phi^{-1}(0.75)} \sqrt{\text{median}_{i \in n} r_i^2} \quad (4.3)$$

where Φ^{-1} is the inverse of the cumulative normal distribution.

The iterative solution to Equation (4.2) can take multiple forms, as noted in Chapter 2 (§ 2.4.1.1). Due to favourable results obtained by the state-of-the-art (e.g. Petit et al., 2013; Zou et al., 2016), the proposed pipeline employs the IRLS formulation, which can be solved with any of the classical least squares techniques, with the additional step of calculating weighing factors $w_i = w(x_i) := \psi(x_i)/x_i$ using the pose estimate $\hat{\mathbf{T}}^{(\kappa)}$ at time $\tau = \tau_\kappa$ in the computation of the new $\hat{\mathbf{T}}^{(\kappa+1)}$. The Levenberg-Marquardt (LM) method is adopted for the IRLS minimisation, for which the solution at each step is computed as (Eqs. [2.27], [2.28], and [2.70]):

$$\boldsymbol{\xi} = -(\mathbf{J}^\top \mathbf{W} \mathbf{J} + \mu \mathbf{I})^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{r}, \quad (4.4)$$

$$\hat{\mathbf{T}}^{(\kappa+1)} = \hat{\mathbf{T}}^{(\kappa)} \boxplus \boldsymbol{\xi}, \quad (4.5)$$

where $\boldsymbol{\xi} := [\boldsymbol{\rho}^\top \ \boldsymbol{\phi}^\top]^\top \in \mathbb{R}^6$ is an element of $\mathfrak{se}(3)$, \mathbf{J} is the Jacobian matrix, $\mathbf{W} := \text{diag}(w_1/\hat{\sigma}, \dots, w_n/\hat{\sigma})$ is the scale-normalised weights matrix, $\mathbf{r} := [r_1 \ \dots \ r_n]^\top$ is the vector of residuals, μ is the LM weight factor, and \mathbf{I} is an identity matrix of appropriate dimensions. The “box-plus” operator, \boxplus , denotes composition of an element of $\mathfrak{se}(3)$ with one of $\text{SE}(3)$ such that the result also belongs to $\text{SE}(3)$, thus guaranteeing that the estimated $\hat{\mathbf{T}}^{(\kappa+1)}$ at each new time-step is a valid pose.

4.3.2.1 Point-based Features

By computing the corresponding feature descriptors of each detected point feature $\mathbf{z}^{(i)}$, these can be matched to other features in one of the views from the database, for which the coordinates of the corresponding points P_i in \mathcal{F}_t , $\mathbf{p}^{(i)}$, have been registered offline. Using the obtained set of 2D-3D correspondences, the following function is

minimised:

$$\Delta_p = \frac{1}{n_p} \sum_{i=1}^{n_p} \rho_p \left(\frac{r_i^{(p)}}{\hat{\sigma}_p} \right), \quad (4.6)$$

where ρ_p is the Tukey M-estimator associated to the point features, $r_i^{(p)}$ is the residual for the i th point match, and n_p is the number of point matches.

The definition of the residual function in the Gold Standard algorithm stems directly from the manifold theory introduced in Chapter 2, Section 2.4.1.1. Since the structural points $\mathbf{p}^{(i)}$ are obtained via ground truth depth maps for keyframes with perfectly known \mathbf{T} , they are considered to be measured with maximum accuracy, and the error is thus concentrated in the measured image points $\mathbf{z}^{(i)}$. In other words, the measurement space \mathcal{X} (see Fig. 2.6, Chap. 2) is a manifold embedded in \mathbb{R}^{2n_p} (i.e. the measurement \mathbf{x} is construed by stacking the x and y components of all $\mathbf{z}^{(i)}$) and the parameter space \mathcal{U} is 6-dimensional (i.e. the dimensions of $\mathfrak{se}(3)$). Furthermore, as each measured image point is obtained algorithmically with the same feature detector, each $\mathbf{z}^{(i)}$ is modelled as a random variable sampled from an isotropic (Gaussian) distribution. The maximum likelihood estimate (MLE) of the pose is in this manner obtained by minimising the Mahalanobis distance (see Eq. [2.59], Chap. 2) which is reduced to the standard 2D geometric, or reprojection, error:

$$\begin{aligned} \mathbf{r}_i^{(p)} &= d_\pi(\mathbf{z}^{(i)}, \mathbf{p}^{(i)}) \\ &:= \pi\left(\mathbf{K}\hat{\mathbf{T}} \oplus \mathbf{p}^{(i)}\right) - \mathbf{z}^{(i)}. \end{aligned} \quad (4.7)$$

with $\mathbf{r}_i^{(p)} := \mathbf{r}_{2i-1:2i}^{(p)}$, defined as the 2×1 i th sub-block of the residual vector, decomposed into both image plane components. Each $2n_p \times 6$ block, $\mathbf{J}_i^{(p)} = \mathbf{J}_{2i-1:2i}^{(p)}$, of the Jacobian matrix, $\mathbf{J}^{(p)}$, corresponding to the residual of each match is known from the VVS theory (Comport et al., 2006; Petit et al., 2014):

$$\mathbf{J}_i^{(p)} = \begin{bmatrix} \frac{f_1}{p_3^{(i)}} & 0 & -f_1 \frac{p_1^{(i)}}{p_3^{(i)2}} & -f_1 \frac{p_1^{(i)} p_2^{(i)}}{p_3^{(i)2}} & f_1 \left(1 + \frac{p_1^{(i)2}}{p_3^{(i)2}}\right) & -f_1 \frac{p_2^{(i)}}{p_3^{(i)}} \\ 0 & \frac{f_2}{p_3^{(i)}} & -f_2 \frac{p_2^{(i)}}{p_3^{(i)2}} & -f_2 \left(1 + \frac{p_2^{(i)2}}{p_3^{(i)2}}\right) & f_2 \frac{p_1^{(i)} p_2^{(i)}}{p_3^{(i)2}} & f_2 \frac{p_1^{(i)}}{p_3^{(i)}} \end{bmatrix}, \quad (4.8)$$

evaluated at $\mathbf{T} = \hat{\mathbf{T}}$, where $\mathbf{p}^{(i)} = [p_1^{(i)} \ p_2^{(i)} \ p_3^{(i)}]^\top = \mathbf{T} \oplus \mathbf{p}^{(i)}$ is taken to be the mapped $\mathbf{p}^{(i)}$ to \mathcal{F}_c , and f_1, f_2 are the focal length normalised by the sensor's horizontal and vertical dimensions, respectively. The derivation of Equation (4.8) is presented in Chapter 5.

4.3.2.2 Edge-based Features

Registered model edges are sampled to a discrete number of 3D points, which are then reprojected onto the image plane, yielding the following edge-based minimisation function:

$$\Delta_e = \frac{1}{n_e} \sum_{i=1}^{n_e} \rho_e \left(\frac{r_i^{(e)}}{\hat{\sigma}_e} \right), \quad (4.9)$$

where ρ_e is the Tukey M-estimator associated to the edge features, $r_i^{(e)}$ is the residual for the i th edge match, and n_e is the number of sample edge point matches. The residual function incorporates the normal distance between the detected and the reprojected edge points:

$$\begin{aligned} r_i^{(e)} &= d_{\perp}(\mathbf{z}^{(i)}, \mathbf{p}^{(i)}) \\ &= \mathbf{n}^{(i)\top} d_{\pi}(\mathbf{z}^{(i)}, \mathbf{p}^{(i)}), \end{aligned} \quad (4.10)$$

where $\mathbf{n}^{(i)}$ is the normal of the i th projected sampled edge point. The computation of the edge Jacobian matrix is approximated by the product of Equation (4.8) with the normal:

$$\mathbf{J}_i^{(e)} \approx \mathbf{n}^{(i)\top} \mathbf{J}_i^{(p)}. \quad (4.11)$$

4.3.2.3 Combining Point and Edge Features

As stated by Petit et al. (2014), the IRLS framework provides a straightforward mechanism to couple different types of features for the estimation of the relative pose. The function to minimise becomes:

$$\Delta = \alpha_p \Delta_p + \alpha_e \Delta_e, \quad (4.12)$$

where α_p, α_e are weighing factors that measure the contribution of each feature type. To compute these weights, the method of Zou et al. (2016) is followed, which states that a larger number of features and a smaller residual vector should contribute more towards the estimated solution. This philosophy leads to the combination of the strong points of each feature type. Thus, the weight is increased proportionally to the number of matches but decreased exponentially when the residual increases:

$$\alpha_p = \frac{n_p}{\sqrt{\Delta_p}} \exp(-\Delta_p), \quad (4.13)$$

$$\alpha_e = \frac{n_e}{\sqrt{\Delta_e}} \exp(-\Delta_e), \quad (4.14)$$

followed by a normalisation:

$$\alpha_p \leftarrow \frac{\alpha_p}{\alpha_p + \alpha_e}, \quad (4.15)$$

$$\alpha_e \leftarrow 1 - \alpha_p. \quad (4.16)$$

Lastly, the factors $\lambda_p = \alpha_p/n_p$, $\lambda_e = \alpha_e/n_e$ are defined to build a global residual vector and Jacobian matrix, respectively, via weighed stacking of the ones from each feature type:

$$\mathbf{r}^\top = \begin{bmatrix} \sqrt{\lambda_p} \mathbf{r}^{(p)\top} & \sqrt{\lambda_e} \mathbf{r}^{(e)\top} \end{bmatrix}, \quad (4.17)$$

$$\mathbf{J}^\top = \begin{bmatrix} \sqrt{\lambda_p} \mathbf{J}^{(p)\top} & \sqrt{\lambda_e} \mathbf{J}^{(e)\top} \end{bmatrix}. \quad (4.18)$$

A global weighing matrix is formed by arranging the respective matrices from each feature type as:

$$\mathbf{W} = \text{blockdiag}(\mathbf{W}^{(p)}, \mathbf{W}^{(e)}). \quad (4.19)$$

Using the computed \mathbf{r} , \mathbf{J} , and \mathbf{W} , Equations (4.4) and (4.5) can be used to iteratively solve for the pose.

4.3.3 Biphasic Approach to Pose Estimation

The procedure for the adopted pose estimation architecture is now presented. This architecture can be branched into two main aspects: the first one is the creation of an offline database using the CAD model of the target, whereas the second one is an online stage which compares information between the buffer images and this database to yield the solution.

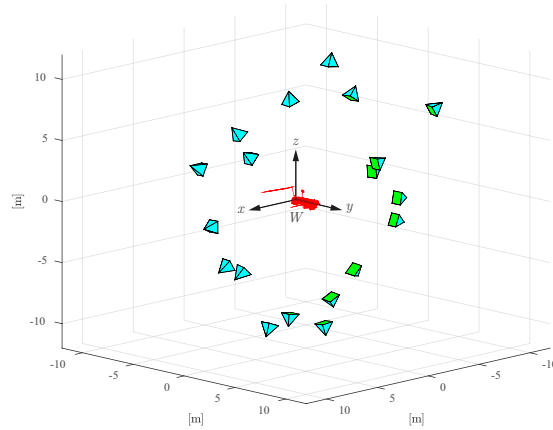


Figure 4.2: Visual representation of the employed multi-view sampling for training. Different viewpoints are obtained by changing the camera’s (in blue) elevation, azimuth, and distance with respect to the target (red) in the \mathcal{F}_t frame. The camera’s front face is highlighted (green).

4.3.3.1 Offline Database Creation

Multi-View Sampling

The architecture of the offline database takes inspiration from the multi-view object detection literature (Liebelt et al., 2008; Thomas et al., 2006), consisting in the rendering of different views of the target model. To accomplish this, a \mathcal{F}_t -centred ring is defined with a minimum radius such that it encases the complete model. Then, a virtual camera is positioned at a certain point on the ring pointing at the origin of \mathcal{F}_t and an image of the target is rendered from that viewpoint. By changing the position of the virtual camera along the ring, as well as the radius itself, a collection of m keyframes $\mathbb{K} = \{\mathcal{K}^{(1)}, \dots, \mathcal{K}^{(m)}\}$ can be generated such that the target is covered from multiple perspectives. Figure 4.2 is an illustrated example of such a sampling strategy where the relative motion is expected to be mostly coincident with the $\underline{t}^{(1)} - \underline{t}^{(3)}$ plane.

Each keyframe $\mathcal{K}^{(i)} = \{\mathbf{I}^{(i)}, \mathbf{K}^{(i)}, \mathbf{T}^{(i)}, \mathbb{Z}_{\text{train}}^{(i)}, \mathbb{D}_{\text{train}}^{(i)}, \mathbb{P}_{\text{p}}^{(i)}, \mathbb{P}_{\text{e}}^{(i)}\}$ is in itself a set of data structures: the rendered image $\mathbf{I}^{(i)}$; the intrinsic and extrinsic camera matrices $\mathbf{K}^{(i)}, \mathbf{T}^{(i)}$, respectively, which are derived automatically using the CAD software; the set of detected Fast-Hessian training keypoints $\mathbb{Z}_{\text{train}}^{(i)} = \{\mathbf{z}^{(\text{train},1)}, \dots, \mathbf{z}^{(\text{train},n_p)}\}$ and corresponding FREAK descriptors $\mathbb{D}_{\text{train}}^{(i)} = \{\mathbf{d}^{(\text{train},1)}, \dots, \mathbf{d}^{(\text{train},n_p)}\}$; and lastly the 3D points and edges in the \mathcal{F}_t frame $\mathbb{P}_{\text{p}}^{(i)} = \{\mathbf{p}^{(\text{p},1)}, \dots, \mathbf{p}^{(\text{p},n_p)}\}$, $\mathbb{P}_{\text{e}}^{(i)} = \{\mathbf{p}^{(\text{e},1)}, \dots, \mathbf{p}^{(\text{e},n_e)}\}$, respectively.

Data training

There are several ways to extract the $\mathbb{P}_p^{(i)}$ and $\mathbb{P}_e^{(i)}$ required to train the keyframes from the database. A simple method consists in backprojecting each point and edge extrema onto the surface of the CAD model: this is achievable by first computing the ray that passes through the detected features by inverting the reprojection equation, π , as in Equation (2.8). Then, since each face in the CAD mesh can be decomposed into triangles, the corresponding 3D model features can be found using a ray-triangle intersection algorithm (e.g. Möller and Trumbore, 1997).

However, this method is not free from drawbacks. Despite the 3D registration being meant to occur offline, the most simple ray-triangle intersection algorithms can prove computationally costly as they require every mesh triangle to be tested. This is particularly impactful when dealing with complex CAD models. Another drawback is that the edge registration might fail for some cases, as these features are located on the boundary of the model’s 2D projection.

As the proposed method does not seek to become encumbered by restrictions such as the need to work with simplified CAD models, the alternative approach of depth mapping is explored, i.e. the generation of training images containing encoded information relating to the distance of the scene objects with respect to the camera viewpoint. For each $\mathcal{K}^{(i)}$ in \mathbb{K} , a corresponding depth map $\mathbf{M}^{(i)} \in \mathbb{M}$, $i \in \{1, \dots, m\}$ is generated in parallel using the same CAD software by exporting the z -buffer output of the scene. Since the depth data is encoded in the image’s intensity values, this means that for a 2D feature detected at image plane coordinate $\mathbf{z}^{(i)}$, the scale of the corresponding 3D point with respect to the origin of \mathcal{F}_c is found by accessing the same coordinates on the depth map. An image from the database and the corresponding generated depth map are represented in Figure 4.3. The figure also shows the corresponding 2D contour edge features obtained by applying Canny’s (1987) method to $\mathbf{M}^{(i)}$, which is an added benefit of the latter, as the lack of texture yields a noiseless feature output, minimising their 3D registration and in turn increasing the accuracy of the online matching process.

4.3.3.2 Online Pose Estimation

Nominal Estimation

The online pose estimation loop consists of the following steps:

- (1) Point features $\mathbb{Z}_{\text{query}}^{(p,\kappa)}$ and edge features $\mathbb{Z}_{\text{query}}^{(e,\kappa)}$ are detected in the current camera image at $\tau = \tau_\kappa$;
- (2) The features are matched to features from the selected model keyframe $\mathcal{K}^{(\kappa)}$;

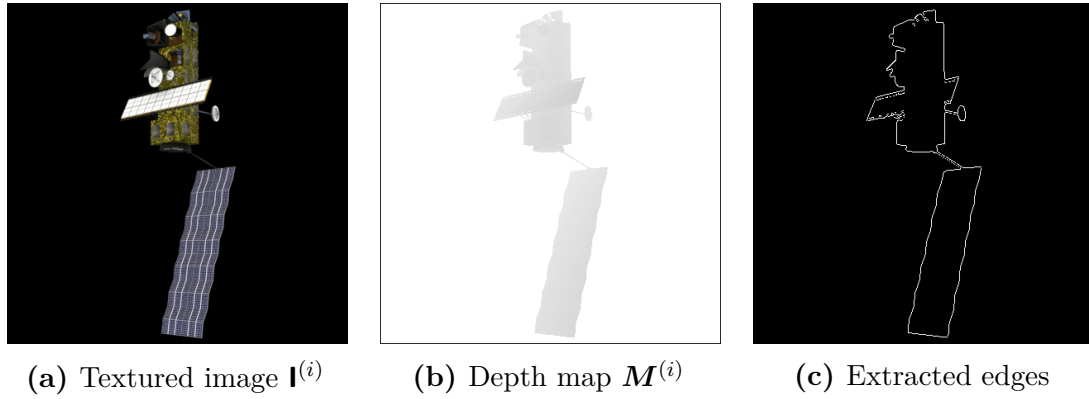


Figure 4.3: Offline training. For each textured training image, an equivalent depth map is rendered, here normalised to an 8-bit depth image for visualisation where darker tones indicate a nearer surface, allowing for the 3D registration of detected features. The true contours are easily obtained from the latter.

- (2-a) In the case of feature points, correspondences $\mathbb{D}_{\text{query}}^{(\kappa)} \leftrightarrow \mathbb{D}_{\text{train}}^{(\kappa)}$ are obtained directly via descriptor matching based on Hamming distance minimisation, followed by a nearest-neighbour distance ratio (NNDR) test¹ on the two closest descriptor matches, i.e. the matching of the descriptors $\mathbf{d}^{(\text{query},i)}$ and $\mathbf{d}^{(\text{train},j)}$ is accepted if the following condition is met:

$$\frac{d_{\text{Ham}}(\mathbf{d}^{(\text{query},i)}, \mathbf{d}^{(\text{train},j)})}{d_{\text{Ham}}(\mathbf{d}^{(\text{query},i)}, \mathbf{d}^{(\text{train},\ell)})} < \mu, \quad (4.20)$$

where d_{Ham} is the Hamming distance, $\mathbf{d}^{(\text{train},j)}$, $\mathbf{d}^{(\text{train},\ell)}$ are the 1st and 2nd nearest neighbours to $\mathbf{d}^{(\text{query},i)}$ and μ is a threshold ranging from 0 to 1;

- (2-b) In the case of edges, each template segment is sampled into 3D points $\mathbb{P}^{(e,i)}$ which are then reprojected onto the query image using the pose estimate from the previous time-step $\hat{\mathbf{T}}^{(\kappa-1)}$, and each point is tested for a potential match by searching for the closest detected edge along a 1D normal search path with an empirically defined length;
- (3) The obtained set of 2D-3D feature correspondences is used in the LM minimisation sub-loop of Equations (4.4) and (4.5) to solve the IRLS problem. The sub-loop is initialised with the previous pose estimate $\hat{\mathbf{T}}^{(\kappa-1)}$ and outputs a current estimate $\hat{\mathbf{T}}^{(\kappa)}$;

¹Also termed the Lowe (2004) Test, after his work with SIFT descriptors for object recognition.

- (4) The reference keyframe for the next time-step $\mathcal{K}^{(\kappa+1)}$, is selected such that the Euclidean distance between $\hat{\mathbf{T}}^{(\kappa)}$ and the registered $\mathbf{T}^{(\text{train},\kappa+1)}$ is minimised. The loop returns to Point (1) and is repeated for the new acquired image.

Initialisation

In order to launch the nominal estimation loop, the initial pose estimate $\hat{\mathbf{T}}^{(0)}$ and database model keyframe $\mathbb{K}^{(0)}$ are recovered. To perform this, a search is carried out by matching the detected point features $\mathbb{P}_{\text{query}}^{(p,i)}$ in the initial frame to the union of all descriptor sets in \mathbb{K} , i.e. $\{\mathbb{D}_{\text{query}}^{(1)} \cup \dots \cup \mathbb{D}_{\text{query}}^{(m)}\}$. By using a *PnP* algorithm that does not require an initialisation, an approximate, initial pose hypothesis can be computed for each set of correspondences. In the proposed method, the EP*n*P (Lepetit, Moreno-Noguer, et al., 2008) method is used in combination with Random Sample Consensus (RANSAC; Fischler and Bolles, 1981) to simultaneously obtain a pose estimate and reject outlying matches. With the obtained $\hat{\mathbf{T}}^{(0)}$, the initial selected keyframe is chosen analogously to Point (4) of the nominal estimation procedure.

It must be noted, though, that each descriptor is a multidimensional vector and, consequently, solving the nearest-neighbour problem over the whole set \mathbb{K} simultaneously using a standard brute-force algorithm is not adequate for real-time processing. In order to overcome this hurdle, a hierarchical *k*-means tree (Gifford, 2014) is built for $\mathbb{D}_{\text{query}}^{(1)} \cup \dots \cup \mathbb{D}_{\text{query}}^{(m)}$. First, a branching factor k_{tree} that defines the number of clusters at each level of the hierarchy is selected. Then, the set of descriptors is grouped into k_{tree} clusters using a standard *k*-means algorithm, cutting the tree such that their variance is minimised. Lastly, each sub-cluster is recursively clustered until a lower bound is reached. While this represents an approximation to the exact brute-force searching, it can be performed in a fraction of the computation time, being particularly useful when there is large inter-frame motion.

Reset

In order to prevent the degradation of the IRLS solution, a monitoring scheme of the associated translation and rotation covariances is proposed to apply a reset if they exceed a certain threshold. This reset consists in generating a new pose estimate from the current 2D-3D point feature matches again with EP*n*P and RANSAC. The new solution is only accepted if it yields a given minimum number of inliers, after which the IRLS is resumed; otherwise the reset is rejected and a new one is attempted after a cool-down period, i.e. after a fixed number of frames.

Table 4.1: Simulated camera properties for the SIMPLESAT dataset.

Parameter	Unit	Value
Resolution	px \times px	640 \times 640
Focal length	mm	16.5
FOV	deg \times deg	44 \times 44
Measurement rate	Hz	10

4.4 Experiments

To validate the proposed method, the performance is evaluated in terms of the position and attitude estimation errors for a continuous rendezvous sequence. The method was implemented in C++, the OpenCV library (version 3.0) was used for computer vision and image processing related functions. The native implementations for the Fast-Hessian feature point detector, **FREAK** feature point descriptor, and **EDLines** line detector were used. For the initialisation stage with hierarchical clustering, the Fast Library for Approximate Nearest Neighbours (**FLANN**) library was used (Muja and Lowe, 2009). Images of the camera sequence and keyframes are generated using the open-source 3D computer graphics software Blender (version 2.78). All simulations are carried out on an Intel[®] Core[™] i7-6700 @ 3.40 GHz \times 8 core central processing unit (CPU), 16 GB RAM system.

4.4.1 Dataset

Rather than testing directly on the **ASTOS** dataset, it is proposed instead that the method developed in this chapter be tested first on a rendezvous sequence depicting an **ADR** scenario involving former remote sensing satellite Envisat, but under less severe conditions. Envisat is a complex object formed by several modules, namely a solar panel array, a synthetic aperture radar (**SAR**), and several antennae, among others, connected to a main body unit which is covered by multi-layer insulation (**MLI**). The modified spacecraft, which has been termed SimpleSAT, attempts to approximately equalise the size of each module in order to increase the quality of feature detection. The resulting dataset, **SIMPLESAT**, features constant illumination only, minimising specular lighting and shadows. However, each model part is still textured differently and therefore looks and reacts to illumination differently. Ultimately, the point of **SIMPLESAT** is to provide an initial analysis of the pose estimation algorithm mostly in the face of tumbling motion.

SIMPLESAT is a continuous rendezvous trajectory synthetically generated by a simulated chaser-mounted camera in the visible wavelength with properties shown in

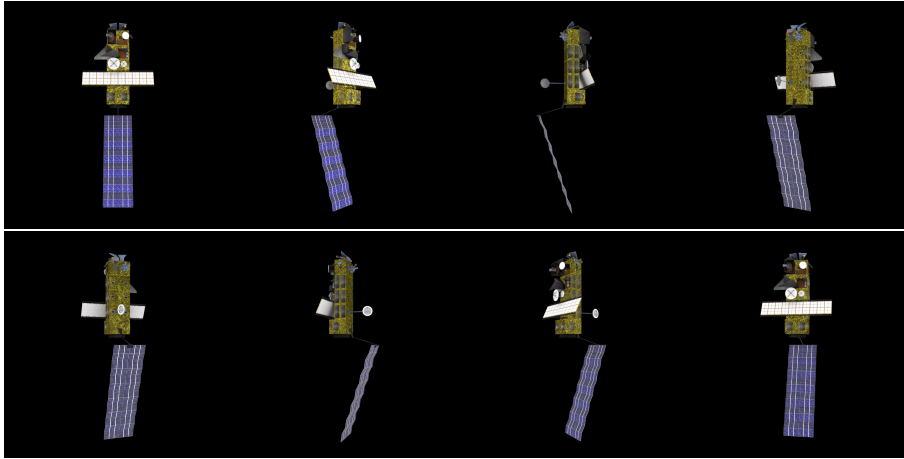


Figure 4.4: Temporally equidistantly sampled frames from the SIMPLESAT dataset (left to right, top to bottom). The target is a modified version of Envisat, tumbling at a constant rate and mode such that self-occlusion is minimised, and maintaining a fixed distance from the chaser spacecraft.

Table 4.1. The target, with body axes defined in Figure 4.2, is located at a fixed position with respect to \mathcal{F}_c measuring 8 m in distance, where the $\underline{c}^{(3)}$ and $\underline{t}^{(3)}$ axes are aligned at time $\tau = \tau_0$. The target rotates at a constant rate of 5 deg s^{-1} along the $\underline{t}^{(2)}$ axis. The sequence lasts 72 s in total, representing a full revolution. The trajectory is represented in Figure 4.4. A total of 19 keyframes are used to build the database (see Figure 4.5). All keyframes are rendered at the same resolution as SIMPLESAT.

4.4.1.1 Testing

The fine pose estimation results are presented in terms of the component-wise position and attitude errors respectively:

$$\delta \tilde{\mathbf{t}} := \hat{\mathbf{t}} - \mathbf{t}, \quad (4.21)$$

$$\delta \tilde{\boldsymbol{\psi}} := \hat{\boldsymbol{\psi}} - \boldsymbol{\psi}, \quad (4.22)$$

where $\mathbf{t} = [t_1 \ t_2 \ t_3]^\top$ is the ground truth position vector, $\boldsymbol{\psi} = [\psi_1 \ \psi_2 \ \psi_3]^\top$ is the ground truth vector of Euler angles, and $(\hat{\bullet})$ denotes an estimated quantity. The errors are also presented in terms of their norm:

$$\delta \tilde{t} := \|\delta \tilde{\mathbf{t}}\|, \quad (4.23)$$

where, regarding the attitude, the error is given in terms of the error quaternion

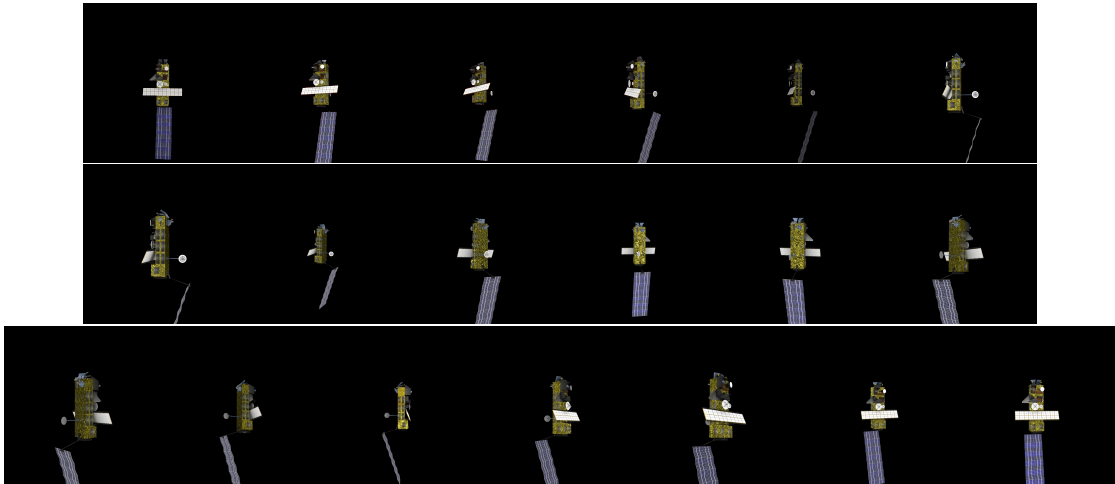


Figure 4.5: Offline-generated keyframes for validation of the proposed method with the SIMPLESAT dataset. Note how these feature variance in scale, azimuth, elevation, and FOV crop (cf. Fig. 4.4).

principal angle $\delta\tilde{q}$, obtained following the well-known relation (Shuster, 1993):

$$\delta\mathbf{q} = \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} = \begin{bmatrix} \delta\mathbf{e} & \delta q \end{bmatrix}^{\top}, \quad (4.24a)$$

$$\delta\tilde{q} := 2 \arccos(\delta q), \quad (4.24b)$$

4.4.2 Results

The results for the estimated relative pose of the target are portrayed in Figure 4.6 qualitatively and in Figure 4.7 quantitatively. It can be seen that most position and attitude dimensions are in close agreement with the ground truth for the majority of the sequence. The largest errors in the pose can be observed in two neighbourhoods centred on frames 180 and 540; effectively, these correspond to the periods when the target completes 90 deg and 270 deg rotations, respectively, showing the greatest degree of self-occlusion in the sequence. This results in the projected surface area of the target reaching a minimum, thus impacting the pose estimate. Figure 4.8 shows the estimation error for the sequence. The translation is accurate up to 0.25 m and the largest error is observed for the $\underline{c}^{(3)}$ axis, corresponding to the camera boresight, thus highlighting the challenges of depth recovery in monocular applications. With respect to rotational motion, the error is kept under 8 deg whereas the ones about the $\underline{c}^{(1)}$ and $\underline{c}^{(2)}$ axes are the largest in magnitude; these are the axes corresponding to out-of-plane rotation.

Both the position as well as the attitude error maxima occur around frame 470,

4. MARKERLESS MULTI-VIEW MONOCULAR POSE ESTIMATION

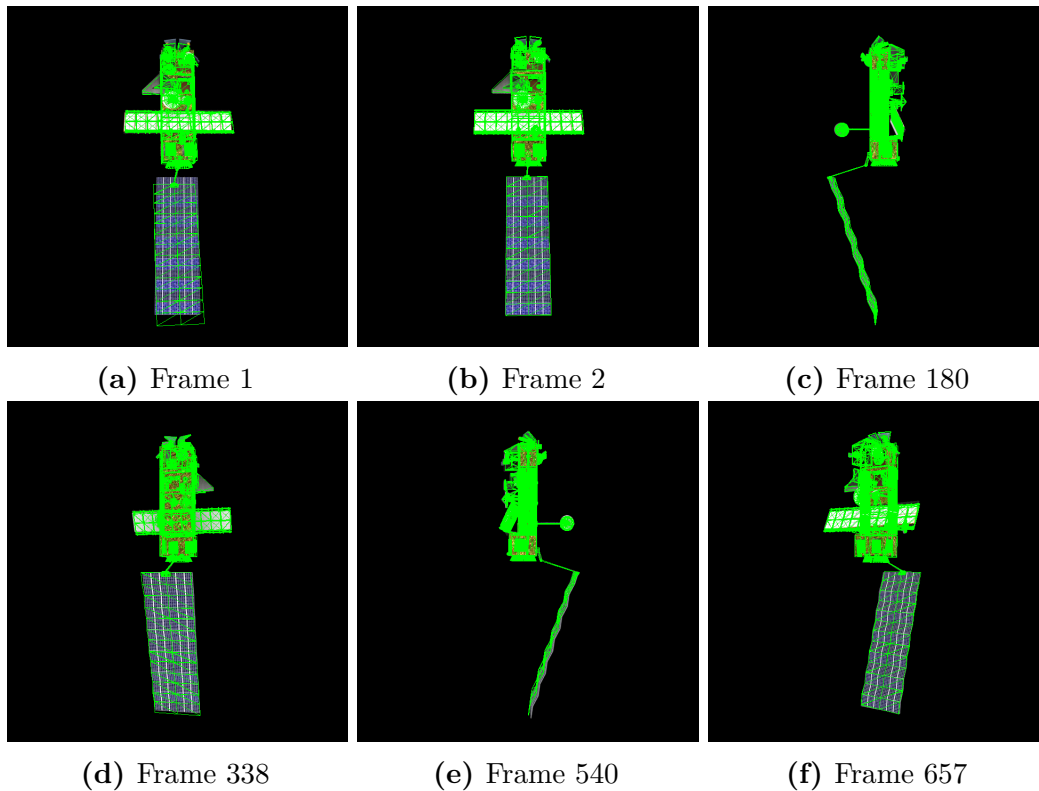


Figure 4.6: Qualitative results for the SIMPLESAT dataset. The model mesh (in green) is reprojected onto the camera image frames using the estimated pose for each time-step. (a) Pose initialisation with hierarchical clustering. (b) Immediate convergence of the solution for frame 2. (c-f) The pose estimate remains robust throughout the sequence.

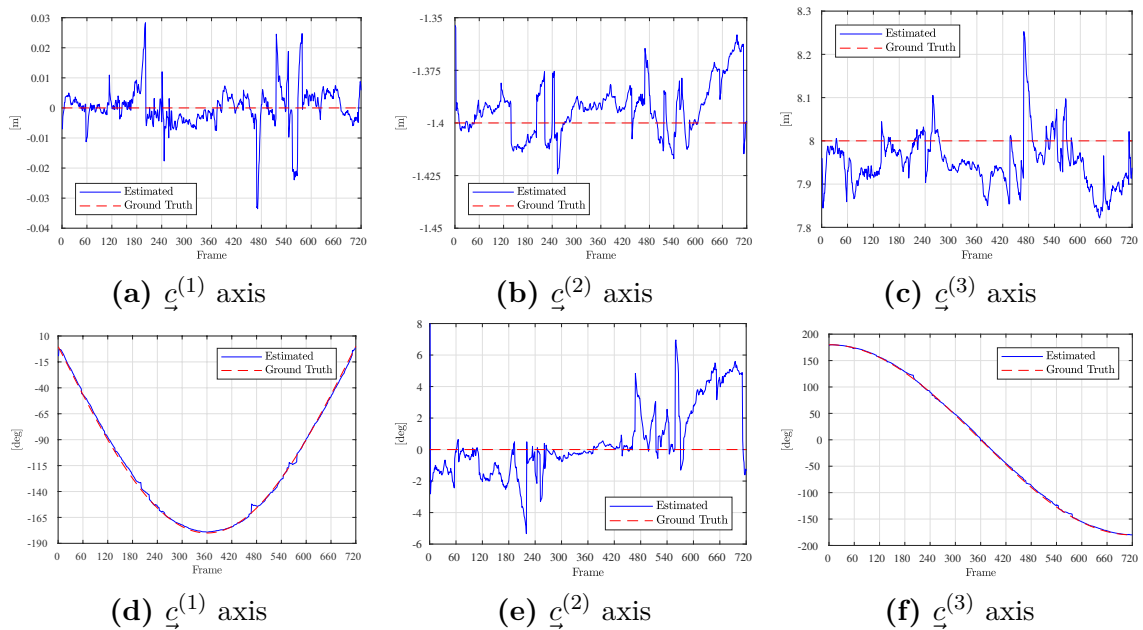


Figure 4.7: Estimated and true values for the target relative position and orientation per axis in \mathcal{F}_c . (Top Row) Position error. (Bottom Row) Attitude error.

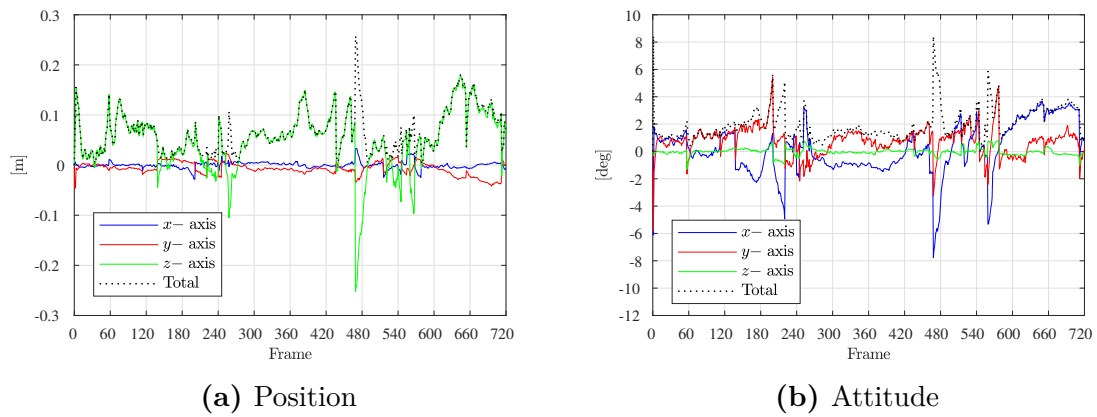


Figure 4.8: Estimation error for the target relative translation and rotation.

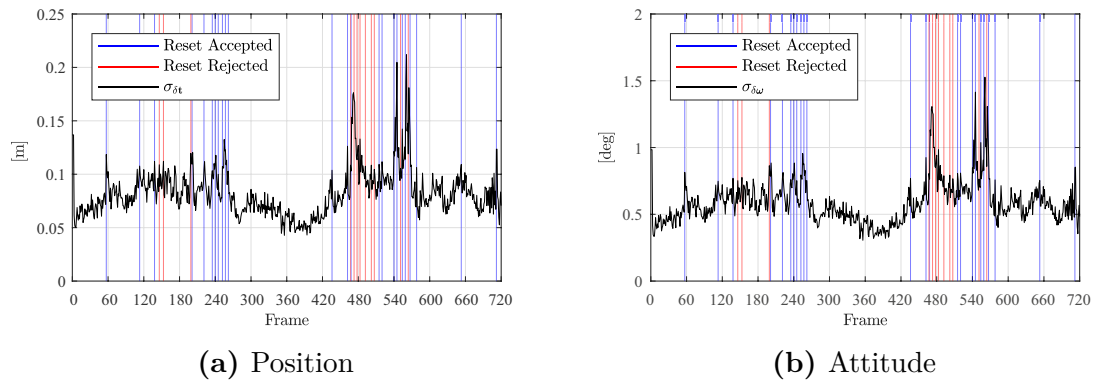


Figure 4.9: Standard deviation of the target estimated relative translation and rotation.

as Simplesat nears the three-quarter turn, whereas the second and third largest error spikes take place near frames 240 and 560, after the quarter and the three-quarter turns are carried out, respectively. Moreover, the uncertainty of the IRLS solution, obtained from the covariance matrix of the linearised solution ξ , is plotted in Figure 4.9. The reset events as described in Section 4.3.3.2 are also represented: in blue for accepted resets, and in red for rejected resets that did not produce the minimum required number of RANSAC inliers. From these plots it can be seen that the three aforementioned events correspond to successful pose resets. These represent trade-offs where estimation accuracy is necessarily sacrificed in order to prevent the Tukey M-estimator from converging to a local minimum. Note how the uncertainty of the solution is brought down after each successful reset.

By analysing Figure 4.10, it can be observed that the number of inliers and the minimisation function scores (i.e. the normalised residuals), respectively, tally with the progression of the error in time. Indeed, an decrease in the number of inliers and an increase in the residuals correlate with a degradation of the solution. The effect is

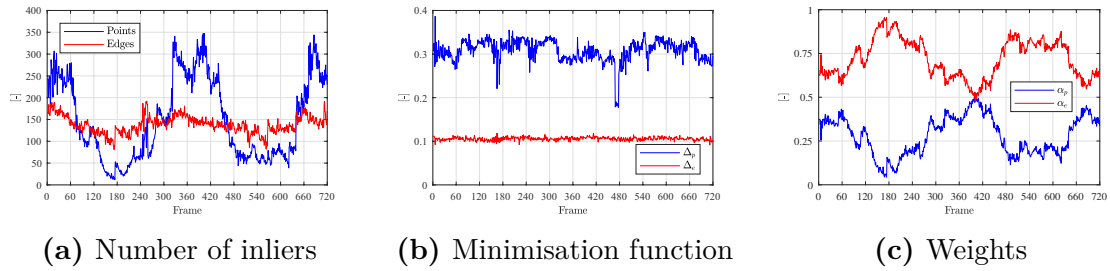


Figure 4.10: Figures of merit for point (blue) and edge (red) features.

more noticeable for the point features, where the decrease in the number of inliers is sharper and the normalised residuals suffer an increase of approximately 15% when compared to edges, for which the increase in residuals is barely noticeable. This degeneration of the point features may be explained due to errors in the matching process coming from the topography of Simplestat. The solar panel array consists of a repetitive grid pattern, whereas the MLI in the main body produces noise in the images when subject to changing illumination. For the former case, the computed descriptors might lack sufficient distinctiveness, and for the latter case it is possible that the features are too disparate from the ones in the database. Both cases will contribute to a decay in the matching process. The edge features are therefore shown to be more robust towards divergence between the camera images and the database.

Furthermore, Figure 4.10 also shows the evolution in time of the self-tuning weights used in the IRLS minimisation process. For the beginning, middle, and end parts of the sequence the weights favour both types of features nearly evenly, showing a distribution of 40–60% leaning towards the edges, which can be explained by their lower normalised residual magnitude being generally 67% lower than its feature point counterpart, even though the latter demonstrates a larger number of inliers. As the estimation error starts to increase, the weights begin to shift their influence further towards the edges; in particular, in the vicinity of frame 180, a weight of at least 90% is attributed to the edge features, as the algorithm reacts to the growth of the point features’ residuals and to the decrease in the number of inliers. Thus, the edge features and the adaptive weighing mechanism prove to be key in preventing the pose estimation from diverging.

The same effect is observed in the neighbourhood of frame 540, albeit slightly more conservatively. The point features are seldom weighed more than the edges, this occurring only for frames 398, 400, and 403, where the number of inliers of the former peak at 350.

Lastly, Table 4.2 depicts the average computation times for the pose estimation framework. The initialisation times were obtained by averaging the results of 1000

Table 4.2: Average pose estimation computation times.

Module	Time (ms)	Relative (%)
Brute-force initialisation	82.90	-
Hierarchical clustering initialisation	46.71	56.34 ^a
Point detection	17.03	21.68
Edge detection	3.52	4.49
Point description	5.56	7.09
Point matching	3.54	4.51
Edge matching	0.73	0.93
IRLS	48.13	61.30
Nominal total	78.52	100.00

^a With respect to the brute-force search counterpart

trials where a random frame from the sequence was considered at each time. This accounts for the descriptor matching, NNDR test, and pose extraction with RANSAC + EPnP. It can be seen that the proposed initialisation with the hierarchical clustering search cuts the running time in almost half when compared to brute-force searching, providing a solution with approximately 56% of the cost with acceptable accuracy (cf. Fig. 4.6). The nominal pose estimation times were attained by averaging the results for each frame of the sequence. The mean nominal pose estimation time per frame is approximately 78.5 ms, equivalent to a mean frame rate of around 12 frames per second (FPS), where it is again emphasised the potential for real-time capability as only the CPU is being utilised. This is an improvement of 4 FPS relative to the work of Petit et al. (2014) which makes use of GPU processing power. The IRLS module is clearly the costliest one, taking up approximately 61% of the total execution time. However, this could be limited by tuning the algorithm’s parameters, such as the maximum number of input point and edge matches, and the maximum number of LM iterations, possibly with a trade-off on accuracy, but ensuring the frame rate is kept above a desired minimum.

4.5 Conclusions and Future Work

In this chapter, model-based solution for relative navigation by using a three-dimensional model of the target and hybrid features was developed. The importance of the proposed framework stands on the fact that it does not depend on convoluted real-time model rendering techniques; instead, a select set of keyframes are rendered a priori for which 3D points are registered on the surface, allowing the retrieval of the pose based only on the matching of 2D point and edge features. The incorporated adaptive weighing algorithm autonomously shifts the influence of both types of

features based on the quality of their matching.

The method was tested on the **SIMPLESAT** dataset, a collection of synthetically generated images of an adapted Envisat model undergoing a tumbling motion, showing promising results for visual-based **NCRV**. The obtained solution shows an attitude error limited to 8 deg and sub-metre translation accuracy for a full target revolution at a high spin rate, relying only on the **CPU**. For future work, an inter-frame tracking module can be added to the present algorithm to further reduce the error and limit jitter. This idea is explored further in Chapter 5, where the algorithm is robustified via the inclusion of an extended Kalman filter (**EKF**) and evaluated on more challenging datasets. Additionally, the relationship between the number of keyframes and the pose estimation performance could be investigated.

CHAPTER 5

Robust On-Manifold Optimisation

This chapter builds upon the method developed in the previous one to reach additional levels of robustness for satellite relative pose estimation. The key offline-online dichotomy is maintained, and a coarse-to-fine estimation philosophy is proposed. The observed facet of the target is tackled as a classification problem, where the three-dimensional shape is learned a priori using Gaussian mixture modelling, producing a rough pose estimate. Then, the solution is refined by minimising two different robust loss functions based on local feature correspondences. The resulting pseudo-measurements are then processed and fused with an extended Kalman filter. The entire optimisation framework is designed to operate directly on the SE(3) manifold, uncoupling the process and measurement models from the global attitude state representation. The method is validated on realistic synthetic and laboratory datasets of rendezvous trajectories with complex targets. It is demonstrated how it achieves an estimate of the relative pose with high accuracy over full tumbling motions.

5.1 Motivation

THE results attained in Chapter 4 have paved the foundation towards proposing a fully autonomous, model-based spacecraft relative pose estimation pipeline for rendezvous. However, two essential questions were left unanswered. The first one is related to the keyframe selection procedure. The keypoint-based hierarchical k -means tree proposed in Section 4.3.3.2 sufficed in the case of a restricted number of possible keyframes and for the simplified SIMPLESAT dataset, but can a good initialisation be guaranteed in a “lost-in-space” scenario, where no a priori information is available about the relative pose? And how would it behave when the target is imaged under

substantially different conditions with respect to the keyframes? The second one pertains to the robustness of the iteratively reweighted least squares (IRLS) procedure. The initial results were dependant on the reset mechanic to prevent the IRLS from becoming stuck in a local minimum. Is this sufficient in the case of more complex relative motion?

This chapter focuses on answering these two key questions in order to improve the baseline method from Chapter 4 to propose a complete and innovative relative navigation framework using a monocular setup on the visible wavelength. In particular, the role of the initialisation is given special attention, in which the objective is not only to recognise the nearest keyframe, or viewpoint, from a single camera image, with no prior assumptions about the pose, but also to provide a first, approximate estimate of it. This is the first step of a coarse-to-fine approach which will then be refined using local feature matching. The second step generates two different pose hypotheses from matching point and edge features and generalised M-estimation. The previous results showed a correlation between the acceptance of a solution and its covariance, which suggest a formulation based on stochastic state estimation, such as the extended Kalman filter (EKF). The classical conundrum of formulating an (originally) linear filter for a nonlinear problem is the linearisation of the error state, which can be performed in a seemingly variety of ways. This chapter will explore the definition of the error on the tangent space of the special Euclidean group $SE(3)$ itself, providing a concise and elegant way to update the pose using the exponential map and giving physical meaning to the attitude part of the estimation covariance. This allows for a shared representation between the M-estimation block and the EKF block, in which the covariance of the former is directly integrated as the measurement noise of the latter, naturally extending the optimisation framework to run continuously on the $SE(3)$ manifold integration in a robust manner.

5.2 Related Work

The focal points of this chapter can be broadly condensed into two fronts: pose initialisation and pose refinement. Existing work related to the latter has been previously surveyed in Chapter 4, Section 4.2; this section thus reviews the literature relevant to the former.

Regardless of either approach taken in model-based pose estimation strategies, these always benefit from initialisation strategies for the incorporation of three-dimensional information, either to propagate it in the case of tracking by recursion, or to reduce the search space in the case of tracking by detection. In the computer vision literature, this has been treated as a coarse, or viewpoint-aware, object pose

estimation. Traditional solutions worked by discretising the object’s 3D appearance into multiple views according to a viewsphere and characterising each bin according to its projected shape using moment invariants (Breuers, 1999; Dudani et al., 1977; Reeves et al., 1988). A viewsphere-based approach for relative pose estimation in space was briefly studied by Chien (1992) and Grimm et al. (1992), but the procedure relied on range sensors to extract the contour of the target and used local descriptors to match the query and database views.

More recent methods adopt classification techniques by clustering local features from each bin into a global representation combined with supervised learning models such as Bayesian classification (Ozuysal et al., 2009) or support vector machines (SVMs; Glasner et al., 2011), or with unsupervised ones such as kernel density estimation (KDE; Mei et al., 2009), to recover the viewpoint. Except for singular cases (Kanani et al., 2012), initialisers for spacecraft relative pose estimation have generally not taken advantage of such formulations, resorting instead to local features and either brute-force matching (Y. Zhang et al., 2015) or iterative methods (J.-F. Shi and Ulrich, 2016); despite simplifications to the search space (Sharma, Ventura, et al., 2018; J.-F. Shi, Ulrich, and Ruel, 2017), these methods still rely on testing multiple hypothesis and discarding outliers, resulting in potentially long computation times due to the volume of features involved in the process.

Recently, deep learning methods, in particular convolutional neural networks (CNN), have shown significant improvements of the state-of-the-art for viewpoint classification (Su et al., 2015; Tulsiani and Malik, 2015). In particular, CNN-based methods have also begun to be adopted for the problem of spacecraft pose estimation, fuelled mainly by the European Space Agency (ESA) Advanced Concept Team’s Satellite Pose Estimation Challenge (SPEC; Kisantal et al., 2020). These approaches are attractive as they shift the focus away from the feature modelling task, but bear some disadvantages such as large amounts of required training data, lower robustness to data outside the training regime, and the need of hardware acceleration (i.e. graphics processing units [GPUs]) to run in real-time. Such methods are surveyed in-depth in Chapter 6.

The method developed herein picks up on the concept of the viewring introduced in Chapter 4 and expands it into a fully-formed viewsphere that encompasses the full set of possible viewpoints under which a rendezvous target can be observed from. The ample number of resulting viewing classes are appropriately modelled by a Bayesian classifier model, which allows the lost-in-space scenario to be quickly solved from a single monocular image. This initialisation provides a coarse estimate of the relative pose with an associated keyframe, which is then refined using local feature

matching. Different hypotheses generated by this refinement step are fused with an **EKF**, where the error state is defined to lie on the tangent space of the special Euclidean group $SE(3)$, providing a concise and elegant way to update the attitude using the exponential map. The prediction stage of the **EKF** is taken advantage of to help predict the locations of the features in the next frame, greatly improving the matching performance under adverse imaging conditions. The contributions of this chapter are summarised below:

- (1) The tackling of the spacecraft pose estimation for relative navigation as a connected coarse classification to fine regression task;
- (2) The development of a relative pose initialisation method modelling the global feature distribution of each viewpoint as a mixture of Gaussians to account for ambiguous shapes;
- (3) The introduction of a predictive feature matching technique to reduce the search space in tracking by detection, adding robustness to scenarios with tumbling and reflective targets where it would otherwise fail; and
- (4) The synergistic integration of geometric pose estimation methods with a navigation filter via the proposed on-manifold optimisation framework, where the measurement noise input of the latter is automatically computed as a byproduct of the former, with a consistent representation of the error-states.

Remark 5.1: Associated Publications

This chapter is based partly on the following published work:

- [J2] D. Rondao, N. Aouf, M. A. Richardson, and V. Dubanchet (2021). “Robust On-Manifold Optimization for Uncooperative Space Relative Navigation with a Single Camera”. In: *Journal of Guidance, Control, and Dynamics*. Article in advance, pp. 1–26. DOI: [10.2514/1.G004794](https://doi.org/10.2514/1.G004794)

5.3 Methodology

Similarly to Chapter 4, the objective of the proposed method is to estimate the camera pose relative to a tumbling target, making no prior assumptions about the rendezvous mission other than the knowledge of the target’s 3D structure, as is the norm with model-based approaches. Figure 5.1 presents a simplified flowchart of the relative navigation framework’s structure.

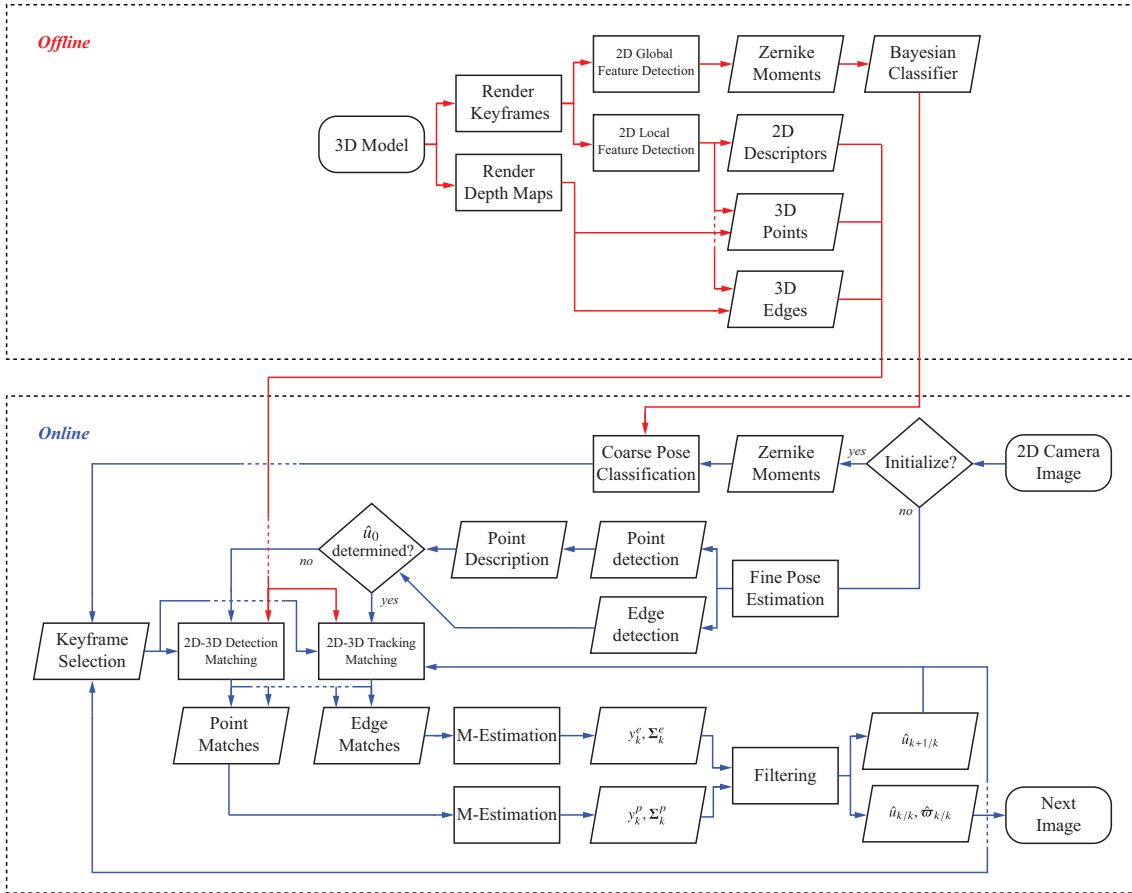


Figure 5.1: High level view of the proposed method. (Red) Offline training stage where the 3D structure of the target is decomposed into tractable models. (Blue) Online training stage where the learned models are used to recover the relative pose for each incoming image (cf. Chap. 4, Fig. 4.1).

The offline training stage has as its objective to discretise, categorise, and represent the three-dimensional structure of the target so that it can be utilised in the two-dimensional environment of the online stage. Two sets of images are sampled from different viewing angles of the target’s computer-aided design (CAD) model. The first is a set of keyframes, each of which contains textural information from the target as imaged from that viewpoint. The second is a set of depth maps, each of which has the same scene structure as its corresponding keyframe, but the value of each pixel represents the distance of that point in the target to the image plane. For each keyframe, the shape of the target is mathematically represented using complex Zernike moments (ZM). The distribution of the ZM feature vector elements per class is learned using Gaussian mixture modelling (GMM), which will define the likelihood probabilities in the training of a Bayesian classifier later employed to match the target’s facet as observed by the on-board camera to the closest keyframe in the database, defining the coarse pose classification module (§ 5.3.1).

The keyframes are also processed with a feature point detector. The aim is to identify keypoints distinguishable enough to be matched to the same keypoint in the context of the online pipeline. Each keypoint is subjected to a feature descriptor, and annotated with its position on the target’s structure using the keyframe’s depth map, generating a 3D-to-2D keypoint catalog to be used with image processing (IP) algorithms compatible with camera-based navigation. Additionally, the target’s limb (or contour) in each keyframe is locally sampled into control points using edge detection. The edge points are converted to 3D using the depth map and grouped into 3D straight keylines; as in Chapter 4, keyline descriptors are not used, and alternative strategies were instead designed. The offline training stage is illustrated in red in Figure 5.1.

The online stage has the purpose of providing a fine pose estimate based on local feature matching after the closest keyframe has been found using coarse pose classification (§ 5.3.2). If no estimate of the pose $\hat{u} \in \mathcal{U} \cong \text{SE}(3)$ has been determined, local features are matched by detection: keypoints from the database pertaining to the current keyframe are matched by brute-force to the ones detected in the camera image, whereas the edges are matched by aligning the keyframe contour to the camera image contour in the least squares sense. Otherwise, the features are matched by recursion. This is not meant in the typical sense that the features are propagated from one camera image to the next, but instead the search space is reduced by reprojecting them from 3D into 2D based on \hat{u} .

The feature matches are processed separately and used to generate direct pseudo-measurements of the six degrees-of-freedom (6-DOF) relative pose. This is achieved by minimising the reprojection error using Levenberg-Marquardt (LM) in an M-estimation framework, which implements the rejection of outlying matches. The measurements are fused with an EKF to produce an estimate of the relative pose and velocity (§ 5.3.3). Both the M-estimator and the filter are accordant in representing the pose error as an element of $\mathfrak{se}(3)$, meaning that the measurement covariance determined from the former is used directly as the measurement noise in the latter, avoiding the need for tuning. The pose predicted by the filter for the following time-step is used to select the next keyframe and in the matching by recursion, providing temporal consistency. The online stage is summarized in blue in Figure 5.1.

5.3.1 Coarse Pose Estimation

The concept of this module is to recover the viewpoint of the three-dimensional target object imaged in a two-dimensional scene using its pre-computed and known CAD model. The goal is to provide an initial, coarse, estimate of the appearance of

the object based on its view classification so that then more precise pose estimation algorithms can be used to refine its pose.

5.3.1.1 Viewsphere Sampling

In order to capture the full three-dimensional aspect of the target, sampled views from the CAD are generated by resorting to the concept of the viewsphere: the model is located at the centre of a sphere, on the surface of which several cameras are placed, pointed at its centre of mass. The necessary viewpoints can be obtained by varying the spherical coordinates of the camera’s position, i.e. the azimuth, elevation, and distance. **The aim of generating a sphere rather than a ring (cf. Chap. 4) is to be able to coarsely cover all possible viewpoints of the target when the a priori relative pose is not known.** The viewsphere is illustrated in Figure 5.2 (cf. Fig. 4.2).

Each dot represents a camera position on the target body frame $\underline{\mathcal{F}}_t$ that will sample a view. Regarding the training of the sampled data, two different approaches using this viewsphere can be outlined. The first approach involves treating each dot on the viewsphere as a class. This has the immediate disadvantage that if a very fine mesh is defined (low Δ_{mesh}), the classes will not be distinctive enough, which could affect the performance of the view classification. On the other hand, selecting a high Δ_{mesh} does not solve the issue that each class will have only exactly one training image to use for the classification scheme. In order to solve both problems, a second approach is adopted in which dots are grouped into patches of width Δ_{class} to form a class, illustrated as the cyan patch in Figure 5.2.

5.3.1.2 Global Feature Description

The following step is to select a measure that mathematically describes each training image obtained as explained above. Such a descriptor will be the basis to establish a correspondence between two viewpoints. The choice for a descriptor for viewpoint classification is motivated by two main points: 1) it must be a global representation of the target and 2) it must be robust to changes likely to be experienced during a space imaging scenario. The first point is justified by the fact that the goal is a classification of the aspect of the target, i.e. what is the view from the database that most closely resembles what the camera is observing. While it is possible in theory to use local descriptors for this task, when considering a spacecraft as the target, the same local features can be expected to be present in multiple views (e.g. those sampled from multi-layer insulation [MLI] or solar panels), which would make the view classification harder. The second point refers to robustness against the model and what is actually observed during the mission; since modelling all the expected

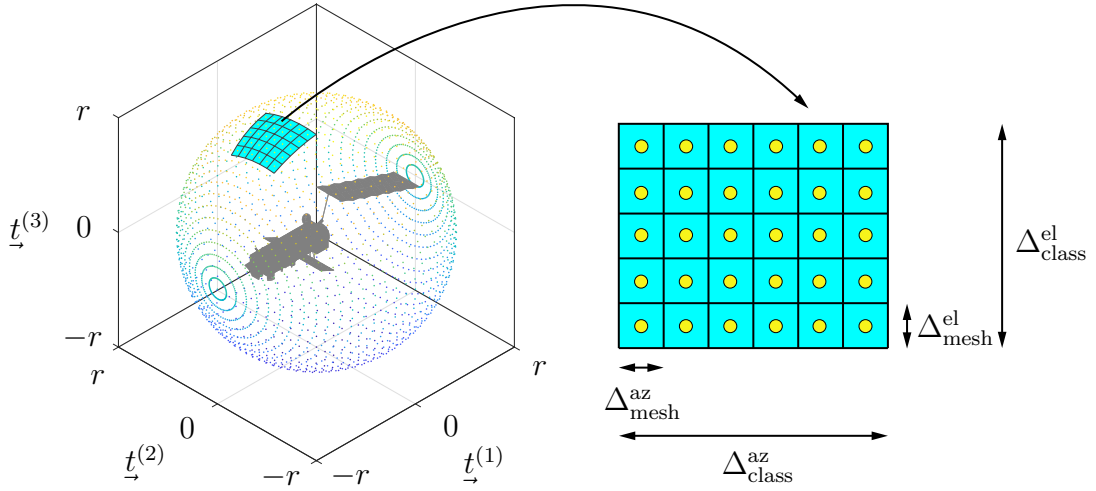


Figure 5.2: The viewsphere for aspect sampling (not to scale). *(Left)* Each dot represents a camera position on the target body frame \mathcal{F}_t that will sample a view. The cyan patch represents a set of views that will define a class for training. *(Right)* A detailed view of the definition of the view mesh resolution Δ_{mesh} , and class resolution Δ_{class} .

cases would be intractable, the descriptors should be resilient towards these, namely: translation, rotation, and scale changes (i.e. the expected 6-DOF in space), off-centre perspective distortions, and illumination changes.

One type of descriptor that satisfies the above requirements are image moments, which are reviewed below in Remark 5.2.

Remark 5.2: Image Moments

The 2D geometric moments of an image are defined as:

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q \bar{I}_{y,x} dx dy, \quad (5.1)$$

where $\bar{I}_{y,x}$ is the pixel intensity value of a greyscale image \bar{I} at the row-column coordinate pair $\{y, x\} \in \mathcal{F}_{\Pi}$, and p, q are the order and repetition of the moment, respectively. Geometric moments are frequently used in rigid-body mechanics to describe mass distributions: m_{00} is the mass of the image, $\{m_{10}/m_{00}, m_{01}/m_{00}\}$ define the centre of gravity of the image, and so on.

Moment computation over a regular image is dependant on the intensity value $\bar{I}_{y,x}$ of each pixel. This implies that the general moment computation (Eq. [5.1]) will

not be robust to illumination changes. Normalising the image would provide global illumination invariance, but not local, therefore another strategy is needed. To this end, the viewpoint image is first binarised before computing the moments. This involves processing the image such that the resulting pixel intensities are mapped to:

$$\text{bin}(\bar{I}_{y,x}) := \begin{cases} 1 & \text{if } \bar{I}_{y,x} > 127, \\ 0 & \text{otherwise.} \end{cases} \quad (5.2)$$

where $\bar{I}_{y,x} \in [0, 255]$ originally. In this way, the target is analysed in terms of its shape, independently of how each patch is illuminated.

Complex Zernike Moments

Binarising an image alone does not solve all purported challenges, in particular those of invariance to rotation, translation, and scaling, as previously mentioned, and to which and Equation (5.1) alone does not provide a solution.

Let $\mathbf{z} = [x \ y]^\top$ define a set of coordinates expressed in \mathcal{F}_Π whose elements correspond to a given column and row, respectively, in $\bar{\mathbf{I}}$. It is first noted that moments are projections of a general function $b(\mathbf{z})$ onto a D -variable polynomial basis $\chi_{\mathbf{n}}$, with $\mathbf{n} = [n_0 \ \dots \ n_D]^\top$, of the space of image functions defined on the image plane Π (Flusser et al., 2016). Then, Equation (5.1) can be generalised to:

$$M_{\mathbf{n}}^{(b)} = \int_{\Pi} \chi_{\mathbf{n}}(\mathbf{z}) b(\mathbf{z}) d\mathbf{z}, \quad (5.3)$$

from which it can be seen, with some abuse of notation, that Equation (5.1) is obtained by taking $\mathbf{n} = [p \ q]^\top$ and $\chi_{\mathbf{n}} = \mathbf{z}^{\mathbf{n}}$. By varying the basis, other types of moments with different or additional properties can be obtained.

Consider the Zernike moment of the n th degree with repetition ℓ , defined in 2D polar coordinates as (Flusser et al., 2016):

$$A_{n\ell} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 V_{n\ell}^*(r, \theta) f(r, \theta) r \, dr \, d\theta, \quad (5.4)$$

where

$$V_{n\ell}(r, \theta) = R_{n\ell}(r) e^{i\ell\theta},$$

$$R_{n\ell}(r) = \sum_{s=0}^{(n-|\ell|)/2} (n-|\ell|)/2 (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|\ell|}{2} - s\right)! \left(\frac{n-|\ell|}{2} - s\right)!} r^{n-2s},$$

with $n = \{0, 1, 2, \dots\}$, $\ell = \{-n, -n + 2, \dots, n\}$, and $(\bullet)^*$ denotes complex conjugation. ZMs have two main attractive properties. Firstly, they are circular moments, meaning they change under rotation in a simple way which allows for a consistent rotation invariant design. Secondly, they are orthogonal moments, which means that they present significant computational advantages with respect to standard moments, such as low noise and uncorrelation. Additionally, orthogonal moments can be evaluated using recurrent relations.

Since they carry these two traits, ZMs are said to be orthogonal on a disk. Hence, in order to compute the moments, the image must be appropriately pre-processed so that it is fully contained in one. By taking this disk to be the unit disk, scale invariance is achieved. Scale invariance is obtained when the image is mapped to the unit disk before calculation of the moments. Translation invariance is obtained by changing the coordinate system to be centred on the centroid. Regarding rotation invariance, one option occasionally seen is to take the ZM as the magnitude $|A_{n\ell}|$. This is not a recommended approach, as essentially the descriptor is cut in half, leading to a likely loss in recognition power. Instead, this chapter considers explicitly both real and complex parts of each ZM, in which case rotation invariance can be achieved by normalising with an appropriate, non-zero moment $A_{n'\ell'}$ (typically A_{31}):

$$A_{n\ell} \leftarrow A_{n\ell} e^{-i\ell\theta}, \quad \theta = \frac{1}{\ell'} \arctan \frac{\text{Im}(A_{n'\ell'})}{\text{Re}(A_{n'\ell'})}, \quad (5.5)$$

where $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ refer to the real and complex parts of the argument, respectively.

A fast computation of the Zernike polynomials up to a desired order can be obtained recursively since any set of orthogonal polynomials obeys a recurrent relation for three terms; in the case of ZMs the following formula has been developed by Kintner (1976):

$$k_1 R_{n+2,\ell}(r) = (k_2 r^2 + k_3) R_{n\ell}(r) + k_4 R_{n-2,\ell}(r), \quad (5.6)$$

where

$$\begin{aligned}
k_1 &= 2n \left(\frac{n+\ell}{2} + 1 \right) \left(\frac{n-\ell}{2} + 1 \right), \\
k_2 &= 2n(n+1)(n+2), \\
k_3 &= -\ell^2(n+1) - n(n+1)(n+2), \\
k_4 &= -2 \left(\frac{n+\ell}{2} \right) \left(\frac{n-\ell}{2} \right) (n+2).
\end{aligned}$$

5.3.1.3 Training the Data

Given the process of generating the data and its descriptors, the final step is defining the classification method. The classifier algorithm shall recognise the aspect of the target given a database of ZM descriptor representation of viewpoints. Given the large volume of data involved, a Bayesian classifier is considered for this task, where the probability density function of each class is approximated using Gaussian mixture models. Bayesian classification is reviewed below in Remark 5.3.

Remark 5.3: Bayesian Classification

In the context of supervised learning (Chap. 2, § 2.4), given a specific class index $y = \{1, \dots, K\}$, where K is the total number of possible classes, and a D -dimensional feature vector $\mathbf{x} = [x_1 \dots x_D]^\top$, a Bayesian classifier works by considering \mathbf{x} as the realisation of a random variable \mathbf{x} and maximising the posterior probability $p(y = k | \mathbf{x})$, i.e. the probability that the feature vector \mathbf{x} belongs to class k , $1 \leq k \leq K$. This probability can be estimated using Bayes' formula (Duda, Hart, and Stork, 2012):

$$p(y = k | \mathbf{x}) = \frac{p(\mathbf{x} | y = k) p(y = k)}{\sum_{j=1}^K p(\mathbf{x} | y = j) p(y = j)}. \quad (5.7)$$

The denominator is constant regardless of the class label and hence can be simply interpreted as a scaling factor ensuring $p(y = k | \mathbf{x}) \in [0, 1]$. Therefore, maximising the posterior is equivalent to maximising the numerator in Equation (5.7):

$$\hat{y} = \arg \max_y p(y) p(\mathbf{x} | y). \quad (5.8)$$

The prior probability, $p(y = k)$, expresses the relative frequency with which the class $y = k$ will appear during the mission scenario; for a general case where one has no prior knowledge of the relative motion, an equiprobable guess can

be made and the term can be set to $1/K$ for any k . The challenge is therefore to estimate the likelihood $p(\mathbf{x} | y = k)$ of class $y = k$, which is given by the respective probability density.

Gaussian Mixture Modelling via Unsupervised Learning

The Gaussian distribution is frequently used to model the probability density of some dataset. In the scope of the present work, it may prove overly optimistic to assume that all elements of the ZM descriptor vectors for each class are independent.¹ On the other hand, it can be too restrictive to model a joint distribution using hard-clustering techniques in case boundaries are not well defined. A more controllable approach to approximate a probability density function, while keeping the tractability of a normal distribution, is to assume the data can be modelled by a mixture of Gaussians:

$$p(\mathbf{x} | \boldsymbol{\theta}) = \sum_{i=1}^M \alpha_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)}), \quad (5.9)$$

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right),$$

where α_i are scalar weighing factors, M is the number of mixture components, $\boldsymbol{\mu}$ denotes the mean vector, and $\boldsymbol{\Sigma}$ the covariance matrix, and $\boldsymbol{\theta} = \{\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)}, \alpha_1, \dots, \boldsymbol{\mu}^{(M)}, \boldsymbol{\Sigma}^{(M)}, \alpha_M\}$ is the full set of parameters required to define the GMM.

When the number of mixture components M is known, the “optimal” mixture for each class, in the maximum likelihood estimate (MLE) sense, can be determined using the classical expectation-maximisation algorithm. Expectation-maximisation works on the interpretation that the set of known observations $\mathbb{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ is part of a broader, complete, data set $\mathbb{X}_t = \mathbb{X} \cup \mathbb{X}_u$ that includes unknown features (Duda, Hart, and Stork, 2012). In the case of GMMs, or finite mixtures in general, $\mathbb{X}_u = \{\mathbf{x}^{(u,1)}, \dots, \mathbf{x}^{(u,N)}\}$ can be defined as the set of N labels denoting which component generated each sample in \mathbb{X} . Each $\mathbf{x}^{(u,i)} = [x_1^{(u,i)} \dots x_M^{(u,i)}]^\top$ is a binary vector such that $x_p^{(u,i)} = 1, x_q^{(u,i)} = 0$ for all $p \neq q$ if sample $\mathbf{x}^{(i)}$ has been produced by the p th component. The expectation, or E-step, calculates the conditional expectation of the log-likelihood given \mathbb{X} and the current best estimate of the model $\hat{\boldsymbol{\theta}}^{(\kappa)}$, where κ denotes the current time-step $\tau = \tau_\kappa$, by evaluating the Q-function:

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(\kappa)}) := \mathbb{E} \left[\log p(\mathbb{X}, \mathbb{X}_u | \boldsymbol{\theta}) \mid \mathbb{X}, \hat{\boldsymbol{\theta}}^{(\kappa)} \right]. \quad (5.10)$$

¹This assumption leads to the so-called naive Bayes classifier (Duda, Hart, and Stork, 2012, Chap. 3).

The M-step updates the parameter estimates according to:

$$\hat{\boldsymbol{\theta}}^{(\kappa+1)} = \arg \max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(\kappa)}). \quad (5.11)$$

However, the number of components is usually not known a priori. There are several methods to iteratively estimate the M ; for this work the method of Figueiredo and Jain (2002) is adopted. The algorithm provides an alternative to the generation of several candidate models, with different numbers of mixture components, and subsequent selection of the best fit, as this approach would still suffer from the drawbacks of expectation-maximisation; namely, the fact that it is highly dependant on initialisation, and the possibility of one of the mixtures' weight α_i approaching zero (i.e. the boundary of the parameter space) and the corresponding covariance becoming close to singular. Instead, Figueiredo and Jain's (2002) method aims to find the best overall model directly. This is achieved by applying the minimum message length criterion to derive the following cost function for finite mixtures:

$$\mathcal{L}(\boldsymbol{\theta}, \mathbb{X}) = \frac{M'}{2} \sum_{i=1}^M \log \left(\frac{N\alpha_i}{12} \right) + \frac{M}{2} \log \frac{N}{12} + \frac{M(M'+1)}{2} - \log p(\mathbb{X} | \boldsymbol{\theta}), \quad (5.12)$$

where M' is the number of parameters specifying each mixture component. A modified M-step is utilized to minimise Equation (5.12), estimating the parameters of each component separately:

$$\begin{aligned} \hat{\alpha}_i^{(\kappa+1)} &= \frac{\max \left\{ 0, \left(\sum_{j=1}^N w_i^{(j)} \right) - \frac{M'}{2} \right\}}{\sum_{\ell=1}^N \max \left\{ 0, \left(\sum_{j=1}^N w_{\ell}^{(j)} \right) - \frac{M'}{2} \right\}}, \\ \hat{\boldsymbol{\theta}}_i^{(\kappa+1)} &= \arg \max_{\boldsymbol{\theta}_i} \mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(\kappa)}). \end{aligned} \quad (5.13)$$

for $i = \{1, \dots, M\}$, and where $w_i^{(j)} := \mathbb{E}[x_i^{(u,j)} | \mathbb{X}, \hat{\boldsymbol{\theta}}^{(\kappa)}]$ is the a posteriori probability that $x_i^{(u,j)} = 1$ after observing $\boldsymbol{x}^{(j)}$, computed as in the regular expectation-maximisation.

The modified M-step performs explicit component annihilation, meaning that when one of the M components becomes unsupported by the data (i.e. close to zero), it is removed, thus impeding the algorithm from approaching the boundary of the parameter space. On the other hand, robustness towards initialisation is achieved by starting the procedure with a large M and iteratively removing the unnecessary ones. If M is too large, it may occur that no component is granted enough initial support, leading the $\hat{\alpha}_i$ to be underdetermined. This is avoided by performing a

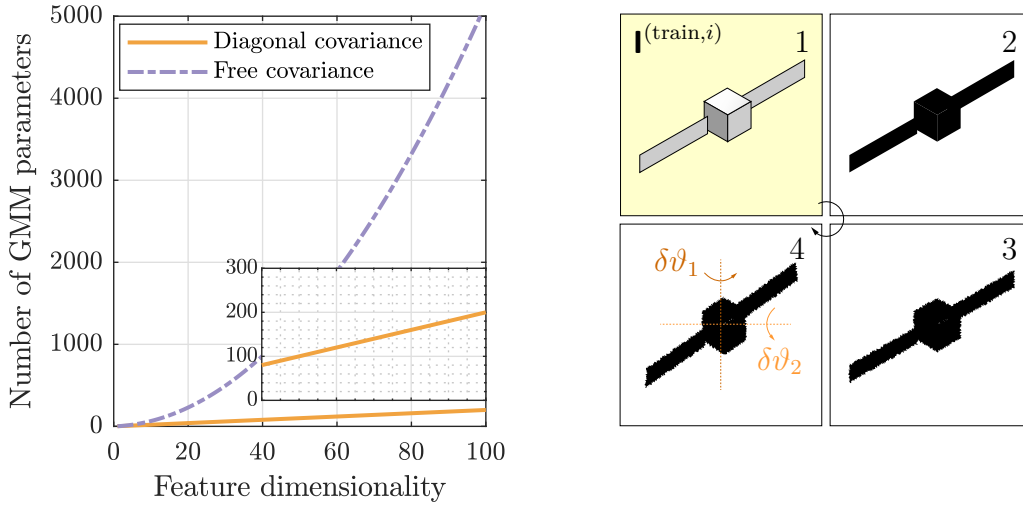


Figure 5.3: Creating a training population. (*Left*) The variation of the number of necessary mixture parameters (M') to estimate in terms of the feature dimensionality (D) for a number of mixture components $M = 1$. (*Right*) Artificial transformations introduced to generate additional training data; clockwise direction: training image, binarisation (including foreground/background segmentation), addition of noise to segmentation outcome, projected target warping.

component-wise update, i.e. recomputing the $w_i^{(j)}$ values every time each element α_i, θ_i is updated, rather than doing it until the last $i = M$; in this way, if one component dies off, its probability mass is automatically redistributed to the other components, increasing their chance of survival. The iteration loop is repeated until the \mathcal{L} -function (Eq. [5.12]) converges. The proposed modifications will allow the modelling of each training class as a probability density in an unsupervised way.

Practical Remarks

This section is concluded with some practical observations on the training procedure. Whereas there is no exact formula that says how much data should be used to train a classifier, it is certainly an element to be considered to assure adequate performance and even convergence. The number of free parameters on a GMM will depend on the dimensionality of the data D , on the number of mixture components M , and on the constraints placed on the covariance Σ . A “free” covariance matrix will have $1/2(D^2 + D)$ independent elements, since it is symmetric, and hence the total number of mixture parameters will be $M' = (1/2D^2 + 3/2D + 1)M - 1$. On the other hand, the covariance can be instead assumed to be diagonal, in which case the total number of parameters to estimate becomes $2MD - 1$. Figure 5.3 (left) plots the evolution of the number of parameters to estimate for a free covariance matrix and for a diagonal one in terms of the dimensionality of the features for $M = 1$. It can be considered as

a lower bound for the number of samples N required for training. The quadratic term in the free covariance case quickly reduces the tractability of the problem when D is increased, which can pose a problem when training data is limited.

S. Li et al. (2009) have shown that the recognition power of complex ZMs for image retrieval begins to plateau beyond moments of the tenth order, which corresponds to approximately $D = 60$. This corresponds to 1890 parameters to be estimated for the free covariance case, while only 120 are necessary if a diagonal covariance is assumed. Since the ZMs are orthogonal, the correlation between moments is minimised and a diagonal covariance is an acceptable approximation. However, even if adjacent training images are grouped to form classes, the generated data might not be enough in terms of training. To this end, each image $\mathbf{I}^{(\text{train},k,i)}$ belonging to the set $\mathbb{I}_{\text{train}}^{(k)}$ that composes class k is subjected to an image augmentation pipeline before the ZMs are computed and added to the training pool (Fig. 5.3, right). This involves adding perturbations to the closed contour (limb) of the binarised target shape and adding small perspective distortions to the image.

5.3.2 Motion Estimation

It has been shown in Chapter 2 that the problem of solving the 2D-3D point correspondences for the 6-DOF pose of a calibrated camera is termed perspective- n -point (P n P) and has a well-known closed form solution for $n = 3$ points (perspective-3-point [P3P]). Additional methods have been developed for $n \geq 4$, such as EP n P (Lepetit, Moreno-Noguer, et al., 2008), which are relatively fast to compute. However, these methods are notwithstanding less robust to noise and fail in the presence of erroneous correspondences. On the other hand, iterative approaches that take these aspects into account, giving the best possible estimate of the pose under certain assumptions are often called the “gold standard” algorithm (Hartley and Zisserman, 2004).

The application of the gold standard algorithm to spacecraft pose estimation has been shown in Chapter 4 on the SIMLESAT dataset with a set of keyframes $\mathbb{K} = \{\mathcal{K}^{(1)}, \dots, \mathcal{K}^{(N_k)}\}$ computed offline, where each keyframe constitutes a collection $\mathcal{K}^{(i)} = \{\mathbf{I}^{(i)}, \mathbf{M}^{(i)}, \mathbf{K}^{(i)}, \mathbf{T}^{(i)}, \mathbb{Z}_{\text{train}}^{(i)}, \mathbb{D}_{\text{train}}^{(i)}, \mathbb{P}_{\text{P}}^{(i)}, \mathbb{P}_{\text{e}}^{(i)}\}$ containing, respectively: a rendered image, the corresponding depth map, the intrinsic and extrinsic camera matrices, the training keypoints and corresponding descriptors, and the 3D points and edges in the target \mathcal{F}_t frame. This was achieved in an IRLS context to reject outlying matches between the keyframe features and the query frame features $\{\mathbb{Z}_{\text{query}}^{(\text{p},\kappa)}, \mathbb{Z}_{\text{query}}^{(\text{e},\kappa)}\}$ at time $\tau = \tau_\kappa$.

In this chapter, a comparable keyframe-based approach is implemented, for which

the initial one is derived from the coarse evaluation of the relative pose described in the previous section. Each class corresponds to one keyframe ($N_k = K$), where the image $\mathbf{I}^{(i)} \in \mathcal{K}^{(i)}$ corresponds to the centre of each mesh cell in the viewsphere (yellow dots in Fig. 5.2). In this section, particularly, an iterative refinement of this coarse estimate based on nonlinear manifold parameterisation is proposed.

5.3.2.1 Structural Model Constraints

From Visual Point Feature Correspondences

It was previously shown that the feature correspondence problem, characterised by the inherent topological difference between measured image points $\mathbb{Z}_{\text{query}} = \{\mathbf{z}^{(\text{query},1)}, \dots, \mathbf{z}^{(\text{query},N_p)}\}$ in two dimensions and model points $\mathbb{P}_p = \{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(N_p)}\}$ in three dimensions, was solvable by offline annotating each $\mathbf{p}^{(i)}$ with a 2D descriptor $\mathbf{d}^{(\text{train},i)}$ computed from its reprojection $\mathbf{z}^{(\text{train},i)}$ with its corresponding $\{\mathbf{I}^{(i)}, \mathbf{M}^{(i)}, \mathbf{K}^{(i)}, \mathbf{T}^{(i)}\} \in \mathcal{K}^{(i)}$. Then, computing a descriptor vector $\mathbf{d}^{(\text{query},j)}$ for the $\mathbf{z}^{(\text{query},j)}$ detected online grants the equivalence $\{\mathbb{Z}_{\text{query}} \leftrightarrow \mathbb{P}_p\} \Leftrightarrow \{\mathbb{Z}_{\text{query}} \leftrightarrow \mathbb{Z}_{\text{train}}\}$, reducing a 3D-2D correspondence problem to a 2D-2D one.

Under the assumption that the structural points are known far more accurately than the image points detected online, which is a valid one for the context of the developed methodology, since the CAD model of the target is given and hence accurate depth maps are produced to register 3D information on the training keyframes with virtually no error, the optimal cost function in the MLE sense was shown to be the reprojection error (Eq. [4.7], Chap. 4), reproduced below for convenience:

$$\hat{\mathbf{T}} = \arg \min_{\mathbf{T} \in \text{SE}(3)} \sum_{i=1}^{N_p} (\pi [\mathbf{KT} \oplus \mathbf{p}^{(i)}] - \mathbf{z}^{(i)})^2, \quad (5.14)$$

where for brevity one defined $\mathbf{T} = \mathbf{T}_{ct}$ as the relative pose in homogeneous form, and $\mathbf{z}^{(i)} = \mathbf{z}^{(\text{query},i)}$ as the detected keypoint in the query image matched to $\mathbf{p}^{(i)}$. Equation (5.14) is solved iteratively via LM with the Jacobian $\mathbf{J}^{(p)}$ denoted in Equation (4.8).

Defining the auxiliary function

$$\begin{aligned} \pi'(\mathbf{p}) &= \pi(\mathbf{Kp}) \\ &= \begin{bmatrix} c_1 + f_1 \frac{p_1}{p_3} \\ c_2 + f_2 \frac{p_2}{p_3} \end{bmatrix} \end{aligned} \quad (5.15)$$

where f_1, f_2 are the sensor dimensions-normalised focal lengths, and using on-manifold optimisation theory (Chap. 2, § 2.4.1.1), in particular, the Jacobian of SE(3) (Eq. [2.64]) and the chain rule, it can be seen that $\mathbf{J}^{(p)}$ is the product of three

terms:

$$\begin{aligned}
 \mathbf{J}_i^{(p)} &= \left. \frac{\partial \pi(\mathbf{K}(\mathbf{T} \boxplus \boldsymbol{\varepsilon}) \oplus \mathbf{p}^{(i)})}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}} \\
 &= \left. \frac{\partial \pi'(\mathbf{p}'^{(i)})}{\partial \mathbf{p}'^{(i)}} \right|_{\mathbf{p}'^{(i)}=\mathbf{T} \oplus \mathbf{p}^{(i)}} \left. \frac{\partial \mathbf{T}' \oplus \mathbf{p}^{(i)}}{\partial \mathbf{T}'} \right|_{\mathbf{T}'=\mathbf{T} \boxplus \boldsymbol{\varepsilon}=\mathbf{T}} \left. \frac{\partial \exp(\boldsymbol{\varepsilon})\mathbf{T}}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}}.
 \end{aligned} \tag{5.16}$$

where $\boldsymbol{\varepsilon} \in \mathfrak{se}(3)$ is a small perturbation of the pose. The first term is obtained by differentiating Equation (5.15) with respect to the elements of \mathbf{p}' :

$$\frac{\partial \pi'(\mathbf{p}'^{(i)})}{\partial \mathbf{p}'^{(i)}} = \begin{bmatrix} \frac{f_1}{p_3'^{(i)}} & 0 & -f_1 \frac{p_1'^{(i)}}{p_3'^{(i)2}} \\ 0 & \frac{f_2}{p_2'^{(i)}} & -f_2 \frac{p_2'^{(i)}}{p_3'^{(i)2}} \end{bmatrix}. \tag{5.17}$$

The second term, recognising that $\mathbf{T} \oplus \mathbf{p} = \mathbf{T}_{1:3,1:4} \tilde{\mathbf{p}}$, where $\tilde{\mathbf{p}} := [\mathbf{p}^\top \ 1]^\top$, is immediate from matrix calculus rules:

$$\begin{aligned}
 \frac{\partial \mathbf{T} \oplus \mathbf{p}^{(i)}}{\partial \mathbf{T}} &= \text{kron}(\tilde{\mathbf{p}}^{(i)\top}, \mathbf{I}_3) \\
 &= \begin{bmatrix} p_1^{(i)} \mathbf{I}_3 & p_2^{(i)} \mathbf{I}_3 & p_3^{(i)} \mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix},
 \end{aligned} \tag{5.18}$$

where $\text{kron}(\bullet, \bullet)$ is the Kronecker product, and \mathbf{I}_3 is the 3×3 identity matrix. The third and last term is dependant on the Jacobian of SE(3); as noted by Blanco (2019):

$$\begin{aligned}
 \left. \frac{\partial \exp(\boldsymbol{\varepsilon})\mathbf{T}}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}} &= \left. \frac{\partial \mathbf{A}\mathbf{T}}{\partial \mathbf{A}} \right|_{\mathbf{A}=\mathbf{I}_4=\exp(\boldsymbol{\varepsilon})} \left. \frac{\partial \exp(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}} \\
 &= \text{kron}(\mathbf{T}^\top, \mathbf{I}_3) \left. \frac{\partial \exp(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}} \\
 &= \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{R}_{:,1}^\wedge \\ \mathbf{0}_{3 \times 3} & -\mathbf{R}_{:,2}^\wedge \\ \mathbf{0}_{3 \times 3} & -\mathbf{R}_{:,3}^\wedge \\ \mathbf{I}_3 & -\mathbf{t}^\wedge, \end{bmatrix}
 \end{aligned} \tag{5.19}$$

where \mathbf{R}, \mathbf{t} are the rotation matrix and translation vector composing \mathbf{T} , respectively, and $(\bullet)^\wedge$ denotes a skew-symmetric matrix. The product of all terms yields the 2×6 block corresponding to $\mathbf{J}_i^{(p)} = \mathbf{J}_{2i-1:2i,:}^{(p)}$, previously illustrated in Equation (4.8):

$$\mathbf{J}_i^{(p)} = \begin{bmatrix} \frac{f_1}{p_3^{(i)}} & 0 & -f_1 \frac{p_1^{(i)}}{p_3^{(i)^2}} & -f_1 \frac{p_1^{(i)} p_2^{(i)}}{p_3^{(i)^2}} & f_1 \left(1 + \frac{p_1^{(i)^2}}{p_3^{(i)^2}} \right) & -f_1 \frac{p_2^{(i)}}{p_3^{(i)}} \\ 0 & \frac{f_2}{p_3^{(i)}} & -f_2 \frac{p_2^{(i)}}{p_3^{(i)^2}} & -f_2 \left(1 + \frac{p_2^{(i)^2}}{p_3^{(i)^2}} \right) & f_2 \frac{p_1^{(i)} p_2^{(i)}}{p_3^{(i)^2}} & f_2 \frac{p_1^{(i)}}{p_3^{(i)}} \end{bmatrix}. \quad (5.20)$$

From Visual Edge Feature Correspondences

The structural model constraints may also be formulated in terms of different types of features, such as edges. This is likewise an important element to consider in space relative navigation, as spacecraft often resemble cuboid shapes or are composed of elements shaped as such; therefore it is expected to have detectable straight edge features when imaging this kind of targets. Indeed, it has been shown in the previous chapter that edges are actually more robust than points in terms of preventing the estimation solution from diverging.

Chapter 4 saw the development of an approximate solution for edge features in which the normal vector to each projected model edge point $\mathbf{p}^{(e,i)} \in \mathbb{P}_e$ was defined by the projections of its two immediate neighbours (see Eqs. [4.10] and [4.11]). Conversely, this chapter presents an exact solution by considering the complete detected 2D straight edge features (i.e. keylines) rather than simply the discretised points $\mathbb{Z}_{\text{query}}^{(e)}$ along detected edges.

First, assume that the previously considered set of model edge points $\mathbb{P}^{(e)}$ belonging to a certain keyframe can be further divided into subsets $\mathbb{P}^{(e,i)}$ containing N_{e_i} points which are discretised from a 3D line $\ell^{(i)} \in \mathbb{LL}$ of the model. Each model line in \mathbb{LL} is matched to a detected keyline $\mathbf{l}^{(i)} \in \mathbb{L}$ in the query image, totalling N_ℓ correspondences. In two dimensions, a point \mathbf{z} lies on a line $\mathbf{l} = [l_1 \ l_2 \ l_3]^\top$ if the condition $\mathbf{z}^\top \mathbf{l} = \mathbf{0}$ is verified. Then, from the correspondences $\{\mathbb{L} \leftrightarrow \mathbb{LL}\}$, a geometric distance for 2D-3D line matches can be formulated in terms of the reprojection of a point $\mathbf{p}^{(i,j)} \in \mathbb{P}^{(e,i)}$ onto the image plane:

$$\hat{\mathbf{T}} = \arg \min_{\mathbf{T} \in \text{SE}(3)} \sum_{i=1}^{N_\ell} \sum_{j=1}^{N_{e_i}} \mathbf{l}^{(i)\top} \mathbf{K} \mathbf{T} \oplus \mathbf{p}^{(i,j)}, \quad (5.21)$$

The resulting 1×6 Jacobian block $\mathbf{J}_i^{(\ell)} = \mathbf{J}_{i,:}^{(\ell)}$ corresponding to each $\mathbf{p}^{(i,j)}$ is found analogously to Equation (5.16):

$$\begin{aligned}
 \mathbf{J}_i^{(l)} &= \left. \frac{\partial \left(\mathbf{l}^{(i)\top} \mathbf{K}(\mathbf{T} \boxplus \boldsymbol{\varepsilon}) \oplus \mathbf{p}^{(i,j)} \right)}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}} \\
 &= \mathbf{l}^{(i)\top} \left. \frac{\partial (\mathbf{K} \mathbf{p}^{(i,j)})}{\partial \mathbf{p}^{(i,j)}} \right|_{\mathbf{p}^{(i,j)}=\mathbf{T} \oplus \mathbf{p}^{(i,j)}} \left. \frac{\partial \mathbf{T}' \oplus \mathbf{p}^{(i,j)}}{\partial \mathbf{T}'} \right|_{\mathbf{T}'=\mathbf{T} \boxplus \boldsymbol{\varepsilon}=\mathbf{T}} \left. \frac{\partial \exp(\boldsymbol{\varepsilon}) \mathbf{T}}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}} \\
 &= \mathbf{l}^{(i)\top} \begin{bmatrix} \frac{f_1}{p_3^{j(i,j)}} & 0 & -\frac{f_1 p_1^{j(i,j)}}{p_3^{j(i,j)^2}} \\ 0 & \frac{f_2}{p_3^{j(i,j)}} & -\frac{f_2 p_2^{j(i,j)}}{p_3^{j(i,j)^2}} \\ 0 & 0 & 0 \end{bmatrix} \left[\mathbf{I}_3 - p_1^{(i,j)} \mathbf{R}_{:,1}^\wedge - p_2^{(i,j)} \mathbf{R}_{:,2}^\wedge - p_3^{(i,j)} \mathbf{R}_{:,3}^\wedge - \mathbf{t}^\wedge \right],
 \end{aligned} \tag{5.22}$$

where the result is left in matrix product form for succinctness.

In practice, as it was seen previously for edges, matching keylines is not as straightforward as matching keypoints, due to the former being typically less distinctive than the latter. For the scope of this chapter, again only the contour of the target is considered, which is discretised into a finite number of edge points that are assumed to belong to a (straight) keyline. Additionally, edge points can be registered in the same way as structural keypoints through the use of depth maps.

5.3.2.2 Local Feature Processing

Detection

For the framework developed in Chapter 4, the Fast-Hessian keypoint detector (Bay et al., 2006) was chosen based on the performance exhibited throughout the analysis of Chapter 3. Despite demonstrating good metrics in terms of repeatability and matching score, Fast-Hessian only ranked fifth out of six in terms of average detection times (see Tab. 3.5).

In this chapter, to compensate for the added computational complexity (see Fig. 5.1), the Oriented FAST and Rotated BRIEF (ORB) keypoint detector (Rublee et al., 2011) is instead used. ORB is based on Features from Accelerated Segment Test (FAST; Rosten and Drummond, 2006), which ranked first in average detection times in the analysis of Chapter 3, but modifies it to allow for multi-scale feature detection and assignment of an orientation to each one by defining a vector from its origin to the intensity barycentre of its support region.

Similarly, for keyline detection, the Edge Drawing Lines (EDLines; Akinlar and

Topal, 2011) algorithm previously used was found to be biased towards lengthier features, missing smaller keylines when the target appears small in the camera field of view (FOV), as in some trajectories of the ASTOS dataset (Chap. 2, § 2.5.2). Instead, Lee et al.’s (2014) method is used instead on the Canny (1987) edge map of the query frame to efficiently extract keylines by incrementally connecting edge pixels in straight lines and merging those with small enough differences in overlap and orientation.

Description

For each detected keypoint, the surrounding support region is encoded into a binary string using the Fast Retina Keypoint (FREAK; Alahi et al., 2012) descriptor. This stems directly from the analysis done in Chapter 3 and remains unaltered with respect to the framework developed in Chapter 4.

Brute-Force Matching by Detection

In an initial stage, the features are matched using brute force, since no estimate of the pose is yet available.

In the case of the point features, this implies that all those detected in the initial frame are compared against those in the train keyframe. This is achieved by computing the Hamming distance $d_{\text{Ham}}(\bullet, \bullet)$ between their corresponding descriptors $\mathbb{D}^{(\text{query},0)}$, $\mathbb{D}^{(\text{train},0)}$ for time $\tau = \tau_0$. For each query, the two closest train descriptor matches are selected and subjected to a nearest-neighbour distance ratio (NNDR) test alike Equation (4.20):

$$\frac{d_{\text{Ham}}(\mathbf{d}^{(\text{query},i)}, \mathbf{d}^{(\text{train},j)})}{d_{\text{Ham}}(\mathbf{d}^{(\text{query},i)}, \mathbf{d}^{(\text{train},\ell)})} < \mu, \quad (5.23)$$

where $\mathbf{d}^{(\text{train},j)}$, $\mathbf{d}^{(\text{train},\ell)}$ are the 1st and 2nd nearest neighbours to $\mathbf{d}^{(\text{query},i)}$ and μ is a threshold ranging from 0 to 1.

As descriptors for edge features are not employed, an alternative strategy was devised to match them. Let $\mathcal{C}_{\text{query}}$ denote the closed contour of the target in the query image, defined by the set of detected edge points $\mathbb{Z}_{\text{query}}^{(e)} = \{\mathbf{z}^{(e,1)}, \dots, \mathbf{z}^{(e,N_e)}\}$ and their sequential ordering. Analogously, let $\mathcal{C}_{\text{train}}$ denote the contour of the initial keyframe. Even though the query image and train keyframe represent the same aspect of the target, there will be differences that are reflected on the contours. In particular, $\mathcal{C}_{\text{query}}$ and $\mathcal{C}_{\text{train}}$ will be different by a 2D affine transformation:

$$\mathbf{\Lambda} = \begin{bmatrix} \beta_{\text{aff}} \cos \phi_{\text{aff}} & -\beta_{\text{aff}} \sin \phi_{\text{aff}} & t_1^{\text{aff}} \\ \beta_{\text{aff}} \sin \phi_{\text{aff}} & \beta_{\text{aff}} \cos \phi_{\text{aff}} & t_2^{\text{aff}} \end{bmatrix}, \quad (5.24)$$

where $\beta_{\text{aff}} > 0$ is the scaling factor, $\phi_{\text{aff}} \in [-180, 180[$ deg is the angle of rotation, and $\mathbf{t}^{\text{aff}} = [t_1^{\text{aff}} \ t_2^{\text{aff}}]^\top$ is the translation vector. The contour alignment problem is posed in the least squares sense as

$$\arg \min_{\{\mathbf{t}^{\text{aff}} \in \mathbb{R}^2, \beta_{\text{aff}}, \phi_{\text{aff}} \in [-180, 180\}} \|\mathcal{C}_{\text{query}} - \mathbf{\Lambda} \mathcal{C}_{\text{train}}\|_{\text{F}}, \quad (5.25)$$

where $\|\mathbf{A}\|_{\text{F}} := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |A_{i,j}|^2}$ is the Frobenius norm of an $n \times m$ matrix \mathbf{A} . Because of the multiplicative trigonometric terms of $\mathbf{\Lambda}$, Equation (5.25) is nonlinear. However, the problem can be converted into an equivalent linear one by a change of variables (Markovsky and Mahmoudi, 2009):

$$\arg \min_{\{t_1^{\text{aff}}, t_2^{\text{aff}}, b_1^{\text{aff}}, b_2^{\text{aff}}\} \in \mathbb{R}^4} \left\| \begin{bmatrix} z_1^{(\text{e}, \text{query}, 1)} \\ z_2^{(\text{e}, \text{query}, 1)} \\ \vdots \\ z_1^{(\text{e}, \text{query}, N_e)} \\ z_2^{(\text{e}, \text{query}, N_e)} \end{bmatrix} - \begin{bmatrix} 1 & 0 & z_1^{(\text{e}, \text{test}, 1)} & -z_2^{(\text{e}, \text{test}, 1)} \\ 0 & 1 & z_2^{(\text{e}, \text{test}, 1)} & z_1^{(\text{e}, \text{test}, 1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & z_1^{(\text{e}, \text{test}, N_e)} & -z_2^{(\text{e}, \text{test}, N_e)} \\ 0 & 1 & z_2^{(\text{e}, \text{test}, N_e)} & z_1^{(\text{e}, \text{test}, N_e)} \end{bmatrix} \begin{bmatrix} t_1^{\text{aff}} \\ t_2^{\text{aff}} \\ b_1^{\text{aff}} \\ b_2^{\text{aff}} \end{bmatrix} \right\|_2, \quad (5.26)$$

where the change of variables is:

$$\begin{bmatrix} b_1^{\text{aff}} \\ b_2^{\text{aff}} \end{bmatrix} = \beta_{\text{aff}} \begin{bmatrix} \cos \phi_{\text{aff}} \\ \sin \phi_{\text{aff}} \end{bmatrix} \Leftrightarrow \begin{bmatrix} \phi_{\text{aff}} \\ \beta_{\text{aff}} \end{bmatrix} = \begin{bmatrix} \arcsin \left(b_2^{\text{aff}} / \sqrt{b_1^{\text{aff}^2} + b_2^{\text{aff}^2} \right) \\ \sqrt{b_1^{\text{aff}^2} + b_2^{\text{aff}^2}} \end{bmatrix}, \quad (5.27)$$

where $\mathbf{z}^{(\text{e})} = [z_1^{(\text{e})} \ z_2^{(\text{e})}]^\top$. In this way, a global solution of the minimum can be calculated using standard linear algebra. However, Equation (5.27) depends on the correspondences between the query and train edge points, which are not known a priori. To simultaneously solve for the edge point correspondence problem and contour alignment, the algorithm is modified by solving N_e linear least squares problems, each time shifting the order of the edge points in $\mathcal{C}_{\text{train}}$ by one, and selecting the minimum of the N_e residual norms. Thus, the only necessary inputs are two sets of sequential but not necessarily correspondent edge points.

The contour alignment algorithm is also used to generate an estimate of the pose to initialise the nominal estimation module, once the first keyframe is output by the coarse estimation module. If $\mathbf{T}^{(i)} \in \mathcal{K}^{(i)}$ is the train pose registered to the i th keyframe recovered at time $\tau = \tau_0$, then, an initial estimate of the 6-DOF pose is computed as:

$$\mathbf{R}^{(0)} = \mathbf{R}_3(\phi^{\text{aff}}) \mathbf{R}^{(i)}, \quad (5.28)$$

$$\mathbf{t}^{(0)} = \begin{bmatrix} \Delta t_1^{\text{aff}} \\ \Delta t_2^{\text{aff}} \\ \beta^{\text{aff}} t_3^{(i)} \end{bmatrix}, \quad (5.29)$$

where $\mathbf{R}_3(\bullet)$ is a rotation matrix applying a rotation about the 3-axis, and $\{\Delta t_1^{\text{aff}}, \Delta t_2^{\text{aff}}\}$ is the centroid of the transformed $\Lambda \mathcal{C}_{\text{train}}$ train contour from $\mathcal{K}^{(i)}$.

Predictive Matching by Recursion

Once the algorithm has been initialised, knowledge of the current solution can be used to improve the performance of the feature matching processes. In particular, the predicted estimate of the pose output by the filtering module is used to help anticipate where the features will be located in the next frame in time, in this way introducing a temporal tracking constraint that improves the pose estimation accuracy.

In the case of point features, recursive matching is achieved by fitting a grid of $p \times q$ cells on the boundary of the target in the query camera image. The detected keypoints are binned into the resulting cells. Then, the 3D structural points of the currently selected database keyframe are reprojected onto the query image according to the predicted pose and equally binned according to the grid. Lastly, descriptor-based matching is applied on a per-cell basis, vastly reducing the number of possible matching candidates. This step was found essential in order to maintain the accuracy of the algorithm during sequences where ambiguous modules are imaged (e.g. *MLI*) or when the query image is too distinct from the train one (e.g. due to reflections).

In the case of edge features, recursive matching is done by first detecting keylines on the query edge image. Each query keyline is then drawn on the image plane with a unique color. The 3D edge points and corresponding keylines from the train keyframe are reprojected onto the image plane. Then, the matching algorithm iterates over each reprojected edge point and a 1D search is performed perpendicularly to it according to the corresponding keyline, obtained in the offline training stage, until the closest coloured pixel is found. Hence, 3D edge points are matched to 2D keylines satisfying the conditions to minimise Equation (5.21).

5.3.2.3 Preliminary Results: Effect of Scale in Robust Pose Estimation

In Section 2.4.1.1, two ways of defining the normal equations for M-estimation were introduced for the robust estimation of the pose, derived from the form:

$$\hat{u} = \arg \min_{u \in \mathcal{U} \cong \text{SE}(3)} \sum_{i=1}^N \rho \left(\frac{r_i}{\hat{\sigma}} \right), \quad (5.30)$$

where the Tukey M-estimator, or **IRLS** (Beaton and Tukey, 1974), was used in Chapter 4 due to its popularity in the computer vision (and classical spacecraft pose estimation) literature due to its hard redescender properties.

However, the scale σ estimation step warrants special attention. In several applications, it can be found that σ is often ignored and set to 1. This is erroneous since Equation (5.30) is non-equivariant with respect to scale (Rousseeuw and Leroy, 1987). Whereas the Huber algorithm (Eq. [2.69], Chap. 2) grants a procedure to jointly estimate the parameter and scale, convergence is not guaranteed when applying the scale estimation step to **IRLS** (Eq. [2.70]; P. J. Huber, 2009). Instead, a common method when resorting to **IRLS** is to recursively estimate σ using the median absolute deviation (**MAD**) for the first few iterations, and then allowing the minimisation to converge on u with fixed σ (Stewart, 1999; Z. Zhang, 1997).

In order to study the effect of scale estimation on the parameter estimation and to compare the different possible approaches, the following experiment has been devised. First, a number of 3D world points is randomly sampled from the volume of a cube. These are subsequently projected onto the image plane according to a random pose. Points that fall outside the image plane are culled. Matches between 3D world points and 2D camera points are contaminated artificially with outliers. Then, the pose is M-estimated with $\rho_{\text{Hub}}(x)$ according to the cost function of Equation (5.14), where the initial guess is defined by contaminating the true pose with zero-mean, white, Gaussian noise. Five distinct methods are benchmarked: (1) least squares (**LS**), (2) Huber's algorithm, (3) **IRLS** with $\sigma = 1$, (4) **IRLS** with σ estimated by one iteration of **MAD**, (5) **IRLS** with σ estimated by three iterations of **MAD**, (6) **IRLS** with σ estimated by Huber's algorithm. The experiment is repeated for several trials.

The results are shown in Figure 5.4. The pose estimation error is decomposed into translation and rotation normalised according to the initial guess. The evolution of the scale estimation is also shown. The percentage of outliers present in the data ranges from 10% to 30%. It can be seen that Huber's algorithm yields the best estimate for every case. The regular **LS** is able to somewhat reduce the attitude error in the presence of outliers, but diverges in the case of translation. Interestingly, all the **IRLS** methods that estimate the scale perform worse than the case where the scale is ignored. These results show the impact on the solution of proper scale estimation and the preference of Huber's algorithm over others. This suggests that robust estimation should be initiated with Huber's algorithm until convergence; to

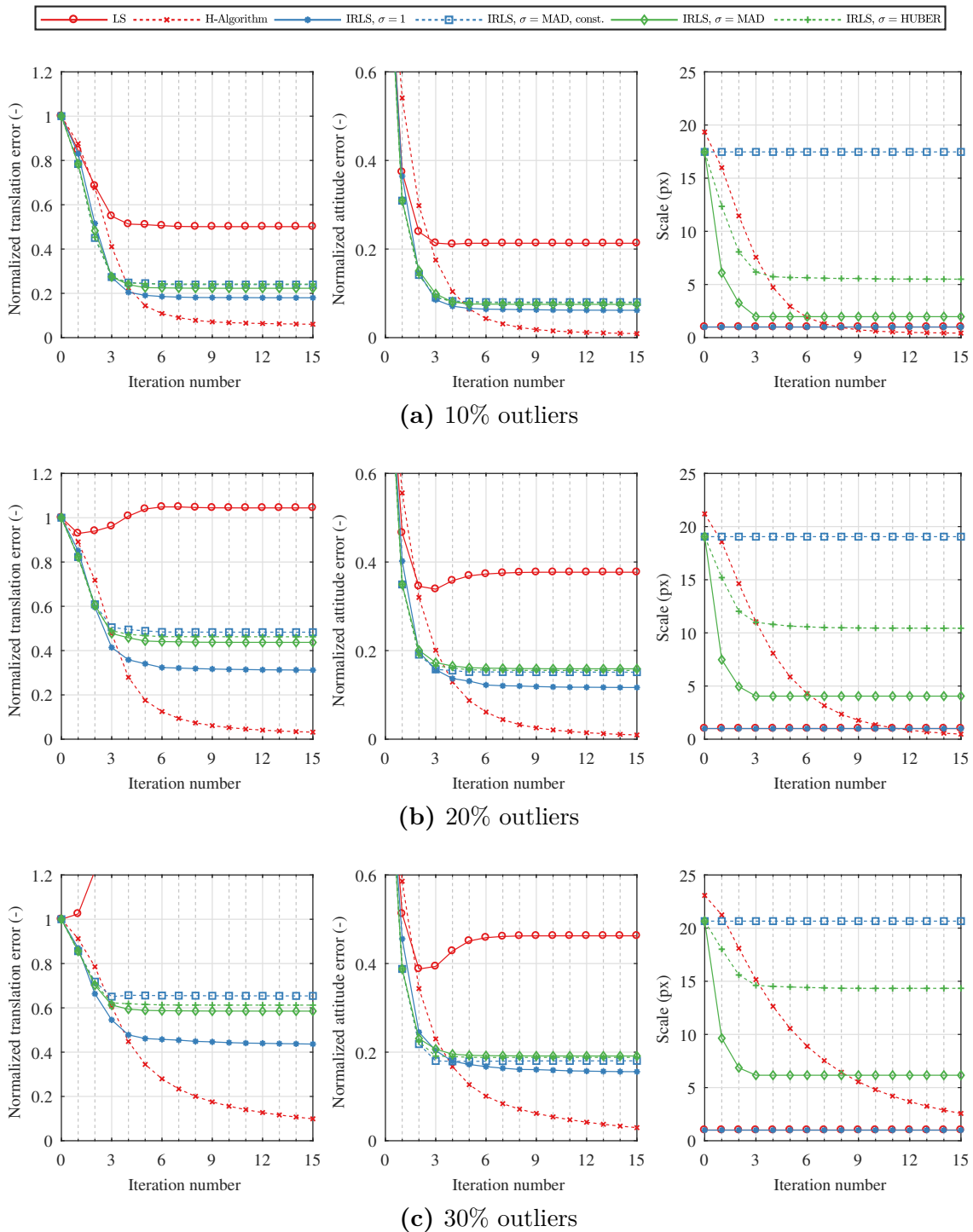


Figure 5.4: Minimisation of the reprojection function from the images of a randomly generated point cloud, averaged over 100 runs, with different amounts of contamination by outlying correspondences.

ensure that the rejection of outliers is maximised, some additional iterations can be performed with IRLS and a hard redescender, such as Tukey's function, using the (fixed) previously obtained estimate of σ , as suggested by Z. Zhang (1997).

5.3.3 Filtering

In this section, the EKF developed to filter the IP-based measurements used in the proposed framework is presented. The general Kalman filter equations are first summarised below in Remark 5.4; then, the adopted modifications to achieve the manifold EKF for spacecraft pose estimation are reported.

Remark 5.4: Kalman Filter Equations

The Kalman filter discrete-time model adopts the form of a general linear stochastic filter (Grewal and Andrews, 2015):

$$\mathbf{x}^{(\kappa)} = \mathbf{\Phi}^{(\kappa-1)} \mathbf{x}^{(\kappa-1)} + \mathbf{G}^{(\kappa-1)} \mathbf{w}^{(\kappa-1)}, \quad (5.31a)$$

$$\mathbf{y}^{(\kappa)} = \mathbf{H}^{(\kappa)} \mathbf{x}^{(\kappa)} + \boldsymbol{\eta}^{(\kappa)}, \quad (5.31b)$$

where \mathbf{x} is the state vector, $\mathbf{\Phi}$ is the state transition matrix, \mathbf{G} is the noise input matrix, \mathbf{y} is the sensor measurement, \mathbf{H} is the measurement matrix, \mathbf{w} , $\boldsymbol{\eta}$ are zero-mean, white random processes with covariances $\mathbf{\Gamma}$ and \mathbf{R} , respectively, and the superscript (κ) denotes evaluation at time-step $\tau = \tau_\kappa$. Equation (5.31a) is termed the process, or motion, model, whereas Equation (5.31b) is the measurement model.

The corresponding Kalman filter equations can be subdivided into a prediction stage and correction stage, where the former consists in:

$$\hat{\mathbf{x}}^{(\kappa)[-]} = \mathbf{\Phi}^{(\kappa-1)} \hat{\mathbf{x}}^{(\kappa-1)[+]}, \quad (5.32a)$$

$$\mathbf{P}^{(\kappa)[-]} = \mathbf{\Phi}^{(\kappa-1)} \mathbf{P}^{(\kappa-1)[+]} \mathbf{\Phi}^{(\kappa-1)\top} + \mathbf{G}^{(\kappa-1)} \mathbf{\Gamma}^{(\kappa-1)} \mathbf{G}^{(\kappa-1)\top}, \quad (5.32b)$$

and the latter consists in:

$$\hat{\mathbf{x}}^{(\kappa)[+]} = \hat{\mathbf{x}}^{(\kappa)[-]} + \mathbf{K}^{(\kappa)} (\mathbf{y}^{(\kappa)} - \mathbf{H}^{(\kappa)} \hat{\mathbf{x}}^{(\kappa)[-]}), \quad (5.33a)$$

$$\mathbf{K}^{(\kappa)} = \mathbf{P}^{(\kappa)[-]} \mathbf{H}^{(\kappa)\top} \left(\mathbf{H}^{(\kappa)} \mathbf{P}^{(\kappa)[-]} \mathbf{H}^{(\kappa)\top} + \mathbf{R}^{(\kappa)} \right)^{-1}, \quad (5.33b)$$

$$\mathbf{P}^{(\kappa)[+]} = \mathbf{P}^{(\kappa)[-]} - \mathbf{K}^{(\kappa)} \mathbf{H}^{(\kappa)} \mathbf{P}^{(\kappa)[-]}, \quad (5.33c)$$

where \mathbf{P} is the state covariance matrix, \mathbf{K} is the Kalman gain matrix, \mathbf{v} is the innovation vector, $(\hat{\bullet})$ denotes an estimate, the $[-]$ superscript denotes the a priori values of the variables (i.e. before correction), and the $[+]$ superscript indicates the a posteriori values.

When the model is nonlinear, the extended Kalman filter (EKF) can be instead adopted, which linearises about the current time-step's state and covariance, and applies the Kalman filter equations to the error state and error state covariance.

5.3.3.1 Rigid Body Kinematics

Let ${}^c\mathbf{p}, {}^t\mathbf{p}$ define one point in the target's rigid body expressed in \mathcal{F}_c and \mathcal{F}_t , respectively. Then, one has

$${}^c\mathbf{p} = \mathbf{R}_{ct} {}^t\mathbf{p} + {}^c\mathbf{t}_{ct}, \quad (5.34)$$

where ${}^c\mathbf{t}_{ct}$ represents the origin of \mathcal{F}_t with respect to \mathcal{F}_c expressed in \mathcal{F}_c . Differentiating with respect to time yields

$$\begin{aligned} {}^c\dot{\mathbf{p}} &= \dot{\mathbf{R}}_{ct} {}^t\mathbf{p} + \dot{{}^c\mathbf{t}}_{ct} \\ &= {}^c\boldsymbol{\omega}_{ct}^\wedge {}^c\mathbf{p} + {}^c\boldsymbol{\nu}_{ct}, \end{aligned} \quad (5.35)$$

where ${}^c\boldsymbol{\omega}_{ct}$ is the angular velocity of \mathcal{F}_t with respect to \mathcal{F}_c resolved in \mathcal{F}_c , and the relationship $\dot{\mathbf{R}}_{ct} = {}^c\boldsymbol{\omega}_{ct}^\wedge \mathbf{R}_{ct}$ was used. The term ${}^c\boldsymbol{\nu}_{ct} := \dot{{}^c\mathbf{t}}_{ct} - {}^c\boldsymbol{\omega}_{ct}^\wedge {}^c\mathbf{t}_{ct}$ represents the velocity of the point in \mathcal{F}_t that corresponds instantaneously to the origin of \mathcal{F}_c . Defining $\dot{\mathbf{T}}_{ct} := \begin{bmatrix} \dot{\mathbf{R}}_{ct} & \dot{{}^c\mathbf{t}}_{ct} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$, one writes the kinematics equation for SE(3) in matrix form (R. M. Murray et al., 1994):

$$\dot{\mathbf{T}}_{ct} = {}^c\boldsymbol{\varpi}_{ct}^\wedge \mathbf{T}_{ct}, \quad (5.36)$$

where

$${}^c\boldsymbol{\varpi}_{ct} := \begin{bmatrix} {}^c\boldsymbol{\nu}_{ct} \\ {}^c\boldsymbol{\omega}_{ct} \end{bmatrix} \quad (5.37)$$

is the rigid body velocity of \mathcal{F}_t with respect to \mathcal{F}_c expressed in \mathcal{F}_c . Dropping the subscripts and superscripts for succinctness, Equation (5.36) is a first-order ordinary differential equation, and hence admits a closed-form solution of the form:

$$\mathbf{T}(\tau) = \exp([\tau - \tau_0]\hat{\boldsymbol{\omega}})\mathbf{T}(\tau_0). \quad (5.38)$$

Equation (5.38) has the same form as Equation (2.28) (Chap. 2), implying that $\boldsymbol{\omega}$ is an element of $\mathfrak{se}(3)$. In agreement with the previous chapters, this fact suggests that uncertainty can be introduced in the pose kinematics by modelling it as a local distribution in $\mathfrak{se}(3)$. As such, it is of interest to develop perturbation equations in terms of the kinematics in $\mathfrak{se}(3)$ so that these can be included as additive noise in a filtering scheme.

Following the approach of Barfoot (2017), the first two terms of Equation (2.13) are used to linearise Equation (2.28) as $\mathbf{T}' \approx (\mathbf{I} + \delta\hat{\boldsymbol{\xi}})\mathbf{T}$, where \mathbf{T} is the nominal pose, $\delta\hat{\boldsymbol{\xi}}$ is a small perturbation in $\mathfrak{se}(3)$, and hence \mathbf{T}' is the resulting perturbed pose. Since $\boldsymbol{\omega} \in \mathfrak{se}(3)$, this generalised velocity can be written directly as the sum of a nominal term with a small perturbation $\boldsymbol{\omega}' = \boldsymbol{\omega} + \delta\boldsymbol{\omega}$. Substituting in Equation (5.36), one has:

$$\frac{d}{d\tau} ([\mathbf{I} + \delta\hat{\boldsymbol{\xi}}]\mathbf{T}) \approx (\boldsymbol{\omega} + \delta\boldsymbol{\omega})^\wedge (\mathbf{I} + \delta\hat{\boldsymbol{\xi}})\mathbf{T}. \quad (5.39)$$

Expanding,

$$\begin{aligned} \delta\dot{\hat{\boldsymbol{\xi}}}\mathbf{T} + \delta\hat{\boldsymbol{\xi}}\dot{\mathbf{T}} + \dot{\mathbf{T}} &= \boldsymbol{\omega}^\wedge\mathbf{T} + \delta\boldsymbol{\omega}^\wedge\mathbf{T} + \boldsymbol{\omega}^\wedge\delta\hat{\boldsymbol{\xi}}\mathbf{T} + \delta\boldsymbol{\omega}^\wedge\delta\hat{\boldsymbol{\xi}}\mathbf{T} \\ (\delta\dot{\hat{\boldsymbol{\xi}}}\mathbf{T} + \delta\hat{\boldsymbol{\xi}}\dot{\mathbf{T}} + \dot{\mathbf{T}})\mathbf{T}^{-1} &= (\boldsymbol{\omega}^\wedge\mathbf{T} + \delta\boldsymbol{\omega}^\wedge\mathbf{T} + \boldsymbol{\omega}^\wedge\delta\hat{\boldsymbol{\xi}}\mathbf{T})\mathbf{T}^{-1} \\ \dot{\mathbf{T}} + \delta\dot{\hat{\boldsymbol{\xi}}} &= \boldsymbol{\omega}^\wedge\mathbf{T} + \delta\boldsymbol{\omega}^\wedge + (\boldsymbol{\omega}^\wedge\delta\hat{\boldsymbol{\xi}} - \delta\hat{\boldsymbol{\xi}}\boldsymbol{\omega}^\wedge), \end{aligned}$$

where after the first step the product of small terms was ignored, and after the second step the identities $\mathbf{T}\mathbf{T}^{-1} = \mathbf{I}$ and $\mathbf{T}^{-1}\dot{\mathbf{T}} = \boldsymbol{\omega}^\wedge$ were used. Noting that the last term is the Lie bracket of $\mathfrak{se}(3)$, subtracting Equation (5.36), and applying the $(\bullet)^\vee$ operator on both sides, the perturbation kinematics equation for SE(3) is obtained:

$$\delta\dot{\hat{\boldsymbol{\xi}}} = \text{ad}(\boldsymbol{\omega}^\wedge)\delta\hat{\boldsymbol{\xi}} + \delta\boldsymbol{\omega}, \quad (5.40)$$

which is linear in both $\delta\hat{\boldsymbol{\xi}}$ and $\delta\boldsymbol{\omega}$.

5.3.3.2 Extended Kalman Filter Formulation

Motion Model

Equation (5.40) describes effectively the linearisation of the rigid body kinematics around a nominal pose. Since it is defined with respect to elements of $\mathfrak{se}(3)$, pertur-

bations in the motion can be modelled stochastically in terms of a local distribution. The mean of this distribution may be injected into the nominal values via the exponential map. Under the assumption of Gaussian noise, this equation can therefore be regarded as the first step in defining an error-state to model how the motion evolves in time in the framework of an extended Kalman filter.

The kinematics of the target's motion with respect to the chaser spacecraft are correctly modelled by Equations (5.36) and (5.40). Modelling the relative dynamics, however, is not a clear-cut task. In the case of an asteroid mission, for example, the chaser could be considered to be inside the sphere of influence of the target and then Newton's second law of motion and Euler's rotation equation could be applied. However, in the case where both chaser and target are under the influence of the same primary, the relative dynamics cannot be shaped as such.

In order to design a filter exclusively with relative states, and inspired by Davison et al.'s (2007) method, a broader constant generalised velocity motion model is adopted:

$$\dot{\boldsymbol{\omega}}(\tau) = \boldsymbol{\eta}_{\boldsymbol{\omega}}(\tau), \quad \boldsymbol{\eta}_{\boldsymbol{\omega}}(\tau) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}[\tau]\delta[\tau - \tau']), \quad (5.41)$$

where $\boldsymbol{\eta}_{\boldsymbol{\omega}}(\tau)$ is a 6×1 zero-mean, white (uncorrelated) process noise, $\mathbf{Q}(\tau)$ is a 6×6 dynamic disturbance noise covariance, and $\delta(\tau)$ is the Dirac delta function. Note that, as stated by Davison et al. (2007), this model does not assume that the chaser moves at a constant velocity over the entire sequence, but instead that undetermined accelerations with a Gaussian profile are expected to occur on average. In other words, one assumes that sizeable (relative) accelerations are unlikely to be experienced, which is a valid expectation for a space rendezvous.

Integrating Equation (5.41) yields:

$$\boldsymbol{\omega}(\tau) = \boldsymbol{\omega}(\tau_0) + \int_{\tau_0}^{\tau} \boldsymbol{\eta}^{(\boldsymbol{\omega})}(\tau') d\tau'. \quad (5.42)$$

The relation $\boldsymbol{\omega}' = \boldsymbol{\omega} + \delta\boldsymbol{\omega}$ was assumed earlier, meaning that one can admit:

$$\delta\boldsymbol{\omega}(\tau) = \int_{\tau_0}^{\tau} \boldsymbol{\eta}^{(\boldsymbol{\omega})}(\tau') d\tau'. \quad (5.43)$$

Defining the error state $\delta\mathbf{x} := [\delta\boldsymbol{\xi}^{\top} \ \delta\boldsymbol{\omega}^{\top}]^{\top}$, the continuous-time error kinematics are written directly:

$$\begin{aligned} \frac{d}{d\tau} \delta \mathbf{x}(\tau) &= \mathbf{F}(\tau) \delta \mathbf{x}(\tau) + \mathbf{G}(\tau) \mathbf{w}(\tau) \\ &= \begin{bmatrix} \text{ad}(\boldsymbol{\varpi}^\wedge) & \mathbf{I}_6 \\ \mathbf{0}_{6 \times 6} & \mathbf{0}_{6 \times 6} \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{\xi} \\ \delta \boldsymbol{\varpi} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{6 \times 6} \\ \mathbf{I}_6 \end{bmatrix} \begin{bmatrix} \mathbf{0}_{6 \times 1} \\ \boldsymbol{\eta}^{(\boldsymbol{\varpi})} \end{bmatrix}, \end{aligned} \quad (5.44)$$

which shows that process noise is introduced in the system through the error generalised velocity vector. Equation (5.44) has the familiar solution (Grewal and Andrews, 2015):

$$\delta \mathbf{x}(\tau) = \Phi(\tau, \tau_0) \delta \mathbf{x}(\tau_0) + \int_{\tau_0}^{\tau} \Phi(\tau, s') \mathbf{G}(s') \mathbf{w}(s') ds' \quad (5.45)$$

$$\Phi(\tau, s) := \exp \int_s^{\tau} \mathbf{F}(\tau') d\tau' \quad (5.46)$$

Setting $\tau_0 = \tau_{\kappa-1}$ and $\tau = \tau_{\kappa}$, the error-state transition matrix has a known closed form (Barfoot, 2017):

$$\begin{aligned} \Phi_{\kappa-1} &:= \Phi(\tau_{\kappa}, \tau_{\kappa-1}) \\ &= \begin{bmatrix} \text{Ad}(\exp[\Delta\tau \boldsymbol{\varpi}_{\kappa-1}^\wedge]) & \Delta\tau \mathbf{B}(\Delta\tau \boldsymbol{\varpi}_{\kappa-1}) \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix}, \end{aligned} \quad (5.47)$$

where $\Delta\tau := \tau_{\kappa} - \tau_{\kappa-1}$ is the discrete time-step increment, $\boldsymbol{\xi} := [\boldsymbol{\rho}^\top \ \boldsymbol{\phi}^\top]^\top$, and $\boldsymbol{\phi} := \|\boldsymbol{\phi}\|$, plus the additional definitions:

$$\mathbf{B}(\boldsymbol{\xi}) := \begin{bmatrix} \mathbf{M}(\boldsymbol{\phi}) & \mathbf{N}(\boldsymbol{\xi}) \\ \mathbf{0}_{3 \times 3} & \mathbf{M}(\boldsymbol{\phi}) \end{bmatrix}, \quad (5.48a)$$

$$\begin{aligned} \mathbf{M}(\boldsymbol{\xi}) &:= \frac{1}{2} \boldsymbol{\rho}^\wedge + \left(\frac{\boldsymbol{\phi} - \sin \boldsymbol{\phi}}{\boldsymbol{\phi}^3} \right) (\boldsymbol{\phi}^\wedge \boldsymbol{\rho}^\wedge + \boldsymbol{\rho}^\wedge \boldsymbol{\phi}^\wedge + \boldsymbol{\phi}^\wedge \boldsymbol{\rho}^\wedge \boldsymbol{\phi}^\wedge) \\ &+ \left(\frac{\boldsymbol{\phi}^2 + 2 \cos \boldsymbol{\phi} - 2}{2\boldsymbol{\phi}^4} \right) (\boldsymbol{\phi}^\wedge \boldsymbol{\phi}^\wedge \boldsymbol{\rho}^\wedge + \boldsymbol{\rho}^\wedge \boldsymbol{\phi}^\wedge \boldsymbol{\phi}^\wedge - 3\boldsymbol{\phi}^\wedge \boldsymbol{\rho}^\wedge \boldsymbol{\phi}^\wedge) \\ &+ \left(\frac{2\boldsymbol{\phi} - 3 \sin \boldsymbol{\phi} + \boldsymbol{\phi} \cos \boldsymbol{\phi}}{2\boldsymbol{\phi}^5} \right) (\boldsymbol{\phi}^\wedge \boldsymbol{\rho}^\wedge \boldsymbol{\phi}^\wedge \boldsymbol{\phi}^\wedge + \boldsymbol{\phi}^\wedge \boldsymbol{\phi}^\wedge \boldsymbol{\rho}^\wedge \boldsymbol{\phi}^\wedge), \end{aligned} \quad (5.48b)$$

and $\mathbf{N}(\boldsymbol{\xi})$ is given by Equation (2.17). A closed-form of the discrete-time error process noise covariance matrix is found by directly solving the integral (Grewal and Andrews, 2015):

$$\boldsymbol{\Gamma}_{\kappa-1} := \boldsymbol{\Gamma}(\tau_{\kappa}, \tau_{\kappa-1}) = \int_{\tau_{\kappa-1}}^{\tau_{\kappa}} \Phi(\tau_{\kappa}, s) \mathbf{G}(s) \mathbf{Q}(s) \mathbf{G}^\top(s) \Phi^\top(\tau_{\kappa}, s) ds, \quad (5.49)$$

The derivation is monotonous but a matter of integrating each matrix element. From Equation (5.47), Equation (2.21), and Equation (2.16), $\Phi(\tau, s)$ can be written in block matrix form as:

$$\Phi(\tau, s) = \begin{bmatrix} \mathbf{A} & \mathbf{EA} & \mathbf{Z} & \mathbf{\Theta} \\ \mathbf{0}_{3 \times 3} & \mathbf{A} & \mathbf{0}_{3 \times 3} & \mathbf{Z} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix}, \quad (5.50)$$

with

$$\mathbf{A} = \mathbf{A}([\tau - s]\boldsymbol{\varpi}) = \exp([\tau - s]\boldsymbol{\omega}^\wedge), \quad (5.51a)$$

$$\mathbf{E} = \mathbf{E}([\tau - s]\boldsymbol{\varpi}) = [\mathbf{N}([\tau - s]\boldsymbol{\omega})[\tau - s]\boldsymbol{\nu}^\wedge], \quad (5.51b)$$

$$\mathbf{Z} = \mathbf{Z}([\tau - s]\boldsymbol{\varpi}) = [\tau - s]\mathbf{M}([\tau - s]\boldsymbol{\omega}), \quad (5.51c)$$

$$\mathbf{\Theta} = \mathbf{\Theta}([\tau - s]\boldsymbol{\varpi}) = [\tau - s]\mathbf{N}([\tau - s]\boldsymbol{\omega}), \quad (5.51d)$$

and $\boldsymbol{\varpi} = [\boldsymbol{\nu}^\top \ \boldsymbol{\omega}^\top]^\top$ are taken to be evaluated at time s . Assuming that:

$$\mathbf{Q}(\tau) = \mathbf{Q} = \begin{bmatrix} \sigma_\nu^2 \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_\omega^2 \mathbf{I}_3 \end{bmatrix}, \quad (5.52)$$

where $\sigma_\nu^2, \sigma_\omega^2$ are the linear and angular velocities variances, respectively. Equation (5.49) can then be simplified:

$$\mathbf{\Gamma}_{\kappa-1} = \int_{\tau_{\kappa-1}}^{\tau_\kappa} \begin{bmatrix} \sigma_\nu^2 \mathbf{Z} \mathbf{Z}^\top + \sigma_\omega^2 \mathbf{\Theta} \mathbf{\Theta}^\top & \sigma_\omega^2 \mathbf{\Theta} \mathbf{Z}^\top & \sigma_\nu^2 \mathbf{Z} & \sigma_\omega^2 \mathbf{\Theta} \\ \sigma_\omega^2 \mathbf{Z} \mathbf{\Theta}^\top & \sigma_\omega^2 \mathbf{Z} \mathbf{Z}^\top & \mathbf{0}_{3 \times 3} & \sigma_\omega^2 \mathbf{Z} \\ \sigma_\nu^2 \mathbf{Z}^\top & \mathbf{0}_{3 \times 3} & \sigma_\nu^2 \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ \sigma_\omega^2 \mathbf{\Theta}^\top & \sigma_\omega^2 \mathbf{Z}^\top & \mathbf{0}_{3 \times 3} & \sigma_\omega^2 \mathbf{I}_3 \end{bmatrix} ds, \quad (5.53)$$

depending only on $\mathbf{Z}, \mathbf{\Theta}$, and on the variances. The small angle approximation ($\sin \phi \approx \phi, \cos \phi \approx 1 - \phi^2/2$) is applied to these two matrices in terms of $(\tau - s)\boldsymbol{\omega}$ to produce a simplified expression for them:

$$\mathbf{Z} \approx (\tau - s)\mathbf{I}_3 + \frac{(\tau - s)^2}{2}\boldsymbol{\omega}^\wedge, \quad (5.54a)$$

$$\mathbf{\Theta} \approx \frac{(\tau - s)^2}{2}\boldsymbol{\nu}^\wedge - \frac{(\tau - s)^3}{4\omega}(\boldsymbol{\omega}^\wedge \boldsymbol{\nu}^\wedge \boldsymbol{\omega}^\wedge + \boldsymbol{\omega}^\wedge \boldsymbol{\nu}^\wedge \boldsymbol{\omega}^\wedge) \quad (5.54b)$$

with $\omega := \|\boldsymbol{\omega}\|$. This is a valid assumption for small inter-frame rotational motion,

i.e. $(\tau - s)\omega \ll 1$. Replacing the quantities in Equation (5.53), integrating each entry is a lengthy task, but it only depends on the coefficient $(\tau - s)$. The obtained closed, approximate form of the discrete-time process noise covariance is thus:

$$\mathbf{\Gamma}_{\kappa-1} \approx \begin{bmatrix} \frac{(\Delta\tau)^3}{3}\sigma_\nu^2\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \frac{(\Delta\tau)^2}{2}\sigma_\nu^2\mathbf{I}_3 + \frac{(\Delta\tau)^3}{6}\sigma_\nu^2\boldsymbol{\omega}^\wedge & \frac{(\Delta\tau)^3}{6}\sigma_\omega^2\boldsymbol{\nu}^\wedge \\ \mathbf{0}_{3\times 3} & \frac{(\Delta\tau)^3}{3}\sigma_\omega^2\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \frac{(\Delta\tau)^2}{2}\sigma_\omega^2\mathbf{I}_3 + \frac{(\Delta\tau)^3}{6}\sigma_\omega^2\boldsymbol{\omega}^\wedge \\ \frac{(\Delta\tau)^2}{2}\sigma_\nu^2\mathbf{I}_3 - \frac{(\Delta\tau)^3}{6}\sigma_\nu^2\boldsymbol{\omega}^\wedge & \mathbf{0}_{3\times 3} & (\Delta\tau)\sigma_\nu^2\mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ -\frac{(\Delta\tau)^3}{6}\sigma_\omega^2\boldsymbol{\nu}^\wedge & \frac{(\Delta\tau)^2}{2}\sigma_\omega^2\mathbf{I}_3 - \frac{(\Delta\tau)^3}{6}\sigma_\omega^2\boldsymbol{\omega}^\wedge & \mathbf{0}_{3\times 3} & (\Delta\tau)\sigma_\omega^2\mathbf{I}_3 \end{bmatrix}, \quad (5.55)$$

where terms with coefficients $(\Delta\tau)^k$ for $k > 3$ have been dropped.

Measurement Model

The correction stage of the **EKF** admits pseudo-measurements of the relative pose $y_i \in \mathcal{Y} \cong \text{SE}(3)$ as obtained through the refinement scheme of visual features correspondence from Section 5.3.2. These pseudo-measurements are acquired at each sampling time and modelled as being corrupted by a zero-mean white Gaussian noise term. One can thus write directly in discrete-time and matrix form:

$$\mathbf{Y} = \exp(\boldsymbol{\eta}^{(y)\wedge})\mathbf{T}, \quad \boldsymbol{\eta}^{(y)} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \quad (5.56)$$

where $\mathbf{Y} \in \text{SE}(3)$ is the homogeneous form of y . To linearise Equation (5.56), similarly to the motion model, the elements of $\text{SE}(3)$ are rewritten as a small perturbation around a nominal term, i.e.:

$$\mathbf{Y}' = \exp(\delta\mathbf{y}^\wedge)\mathbf{Y}, \quad (5.57a)$$

$$\mathbf{T}' = \exp(\delta\boldsymbol{\xi}^\wedge)\mathbf{T}. \quad (5.57b)$$

Replacing in Equation (5.56), and approximating the exponential map by its first-order expansion, yields:

$$(\mathbf{I} + \delta\mathbf{y}^\wedge)\mathbf{Y} \approx (1 + \boldsymbol{\eta}_y^\wedge)(\mathbf{I} + \delta\boldsymbol{\xi}^\wedge)\mathbf{T}.$$

Expanding and neglecting the product of small terms, the following linearised

relationship is obtained:

$$\mathbf{Y} = \mathbf{T}, \quad (5.58a)$$

$$\delta \mathbf{y} = \delta \boldsymbol{\xi} + \boldsymbol{\eta}^{(y)}. \quad (5.58b)$$

The full linearised measurement model is therefore:

$$\begin{aligned} \delta \mathbf{y}^{(i)} &= \mathbf{H} \delta \mathbf{x} + \boldsymbol{\eta}^{(y,i)} \\ &= \begin{bmatrix} \mathbf{I}_6 & \mathbf{0}_{6 \times 6} \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{\xi} \\ \delta \boldsymbol{\varpi} \end{bmatrix} + \boldsymbol{\eta}^{(y,i)}, \quad \boldsymbol{\eta}^{(y,i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^{(i)}) \end{aligned} \quad (5.59)$$

where $i = \{\text{p}, \text{e}\}$ refers to measurements derived from either point- or edge-based M-estimates, respectively. The covariance matrices of each pseudo-measurement are obtained as a product of the minimisation scheme. In the case of the structural model constraints, the Jacobians $\mathbf{J}^{(i)}$ are of rank 6, so the covariance of the solution is given by backpropagation of the visual feature correspondences' own covariance (Hartley and Zisserman, 2004):

$$\mathbf{R}^{(i)} = \left(\mathbf{J}^{(i)\top} \Sigma^{(z,i)} \mathbf{J}^{(i)} \right)^{-1}, \quad (5.60)$$

with $\Sigma^{(z,i)} = \sigma_{z,i}^2 \mathbf{I}$ and $\sigma_{z,i}$ is the scale obtained via M-estimation. The EKF innovation term at time $\tau = \tau_\kappa$ is:

$$\mathbf{v}^{(i,\kappa)} = \mathbf{y}^{(i,\kappa)} \boxminus \hat{\mathbf{u}}_\kappa^{[-]}, \quad \mathbf{v}^{(i,\kappa)} \in \mathfrak{se}(3), \quad (5.61)$$

where $\hat{\mathbf{u}}_\kappa^{[-]}$ is the predicted pose at τ_κ .

Data Fusion

Both measurements can be modeled as coming from two different synchronous sensors. In order to perform the correction stage in a single step, the inverse-covariance form of the EKF is employed (Durrant-Whyte, 2001; Maybeck, 1979). This avoids a double computation of the Kalman gain matrix (if a sequential-sensor correction method were adopted) or the inversion of an innovation covariance matrix of size proportional to the number of sensors (group-sensor method).

In the linear Kalman filter, the inverse-covariance method involves reorganising

the prediction and observation equations to yield the relations:

$$\mathbf{K}^{(\kappa)} = \mathbf{P}^{(\kappa)[+]} \mathbf{H} \mathbf{R}^{-1}, \quad (5.62a)$$

$$\mathbf{I} - \mathbf{K}^{(\kappa)} \mathbf{H} = \mathbf{P}^{(\kappa)[+]} (\mathbf{P}^{(\kappa)[-]})^{-1}, \quad (5.62b)$$

where \mathbf{K} is the Kalman gain matrix, $\mathbf{P}^{[-]}$, $\mathbf{P}^{[+]}$ are the predicted and corrected covariance matrices, respectively, and the superscript (κ) denotes evaluation at time-step τ_κ . Substituting Equations (5.62a) and (5.62b) into Equation (5.33a), and noting from the group-sensor method (Durrant-Whyte, 2001) that, for an S number of sensors:

$$\begin{aligned} \mathbf{H}^\top \mathbf{R}^{(-1)} \mathbf{y}^{(\kappa)} &= \begin{bmatrix} \mathbf{H}^{(1)\top} & \dots & \mathbf{H}^{(S)\top} \end{bmatrix} \begin{bmatrix} \mathbf{R}^{(1)-1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{R}^{(S)-1} \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(1,\kappa)} & \dots & \mathbf{y}^{(S,\kappa)} \end{bmatrix} \\ &= \sum_{i=1}^S \mathbf{H}^{(i)\top} \mathbf{R}^{(i)-1} \mathbf{y}^{(i,\kappa)}, \end{aligned} \quad (5.63)$$

the correction equations are reworked as follows:

$$\hat{\mathbf{x}}^{(\kappa)[+]} = \mathbf{P}^{(\kappa)[+]} \left[(\mathbf{P}^{(\kappa)[-]})^{-1} \hat{\mathbf{x}}^{(\kappa)[-]} + \sum_{i=1}^S \mathbf{H}^{(i)\top} \mathbf{R}^{(i)-1} \mathbf{y}^{(i,\kappa)} \right], \quad (5.64a)$$

$$\mathbf{P}^{(\kappa)[+]} = \left[(\mathbf{P}^{(\kappa)[-]})^{-1} + \sum_{i=1}^S \mathbf{H}^{(i)\top} \mathbf{R}^{(i)-1} \mathbf{H}^{(i)\top} \right]^{-1}, \quad (5.64b)$$

where the summation in Equation (5.64b) is obtained similarly to Equation (5.63). Equation (5.64a) cannot be directly applied to the manifold EKF due to the dimension mismatch between the error covariance matrix and the full state vector, since the rotation part of the former is defined on $\mathfrak{se}(3)$. However, it can be reworked to yield instead the a posteriori error state vector using the EKF correction equation:

$$\delta \mathbf{x}^{[+](\kappa)} = \mathbf{K}^{(\kappa)} \mathbf{v}^{(\kappa)}. \quad (5.65)$$

This relation can be used to replace $\mathbf{K}^{(\kappa)}$ in one of the previously derived expressions by multiplying both sides by $\mathbf{v}^{(\kappa)}$:

$$\begin{aligned} \mathbf{K}^{(\kappa)} &= \mathbf{P}^{[+](\kappa)} \mathbf{H}^\top \mathbf{R}^{-1} \Leftrightarrow \\ \Leftrightarrow \delta \mathbf{x}^{[+](\kappa)} &= \mathbf{P}^{[+](\kappa)} \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{v}^{(\kappa)}. \end{aligned} \quad (5.66)$$

Then, the a posteriori error can be computed as a linear combination of each sensors' quantities:

$$\delta \mathbf{x}^{[+](\kappa)} = \mathbf{P}^{[+](\kappa)} \left[\sum_{i=1}^2 \mathbf{H}^{(i)\top} \mathbf{R}^{(i)-1} \mathbf{v}^{(i,\kappa)} \right]. \quad (5.67)$$

Measurement Gating

An additional step is employed prior to the correction to ensure the accurate functioning of the filter. This involves subjecting the incoming measurements to a validation gate, thus discarding potential spurious data. The validation gate is a threshold on the root mean squared error (RMSE) of the stacked residuals \mathbf{r} as obtained by the M-estimation:

$$\text{RMSE}(\mathbf{r}) := \sqrt{\frac{\sum_{i=1}^N r_i^2}{N}}, \quad (5.68)$$

where N is the number of correspondences. The RMSE provides an objective and clear interpretation of how close, in pixels, does the feature matching agree with the estimate of the pose.

5.3.3.3 Manifold State Prediction and Correction

The nominal state is construed, with some abuse of notation, as:

$$\mathbf{x} = \begin{bmatrix} u \\ \boldsymbol{\varpi} \end{bmatrix}, \quad (5.69)$$

with $u \in \mathcal{U} \cong \text{SE}(3)$ representing the relative pose mapping $\mathcal{F}_t \rightarrow \mathcal{F}_c$ and $\boldsymbol{\varpi} \in \mathbb{R}^6$ is the generalised velocity satisfying the kinematics equation for SE(3) (Eq. [5.36]). The nominal state estimate is updated via pose composition (Eq. [2.27], Chap. 2) with a linearised error state estimate:

$$\delta \mathbf{x} = \begin{bmatrix} \delta \boldsymbol{\xi} \\ \delta \boldsymbol{\varpi} \end{bmatrix}, \quad (5.70)$$

with $\delta \boldsymbol{\xi} \in \mathfrak{se}(3) \times \mathbb{R}^6$, ensuring that u remains an element of $\mathcal{U} \cong \text{SE}(3)$. The algorithm's equations are valid for any chosen representation of u provided the

appropriate composition \boxplus is used. State prediction is performed as:

$$\hat{u}^{(\kappa)[-]} = \hat{u}^{(\kappa-1)[+]} \boxplus \Delta\tau \hat{\omega}^{(\kappa-1)[+]}, \quad (5.71)$$

$$\hat{\omega}^{(\kappa)[-]} = \hat{\omega}^{(\kappa-1)[+]}. \quad (5.72)$$

The state correction is given by:

$$\hat{u}^{(\kappa)[+]} = \hat{u}^{(\kappa)[-]} \boxplus \delta \hat{\xi}^{(\kappa)[+]}, \quad (5.73)$$

$$\hat{\omega}^{(\kappa)[+]} = \hat{\omega}^{(\kappa)[-]} + \delta \hat{\omega}^{(\kappa)[+]}. \quad (5.74)$$

The covariance is calculated using the standard **EKF** equations.

In terms of the parametrisation of u , the unit quaternion is adopted due to its popular choice for attitude representations, particularly in aerospace applications, as it is both compact and singularity-free (see § 2.3.2, Chap. 2). In this case, the state vector becomes:

$$\mathbf{x} = \begin{bmatrix} t \\ \mathbf{q} \\ \boldsymbol{\varpi} \end{bmatrix}, \quad (5.75)$$

which has dimensions 13×1 .

5.4 Experiments

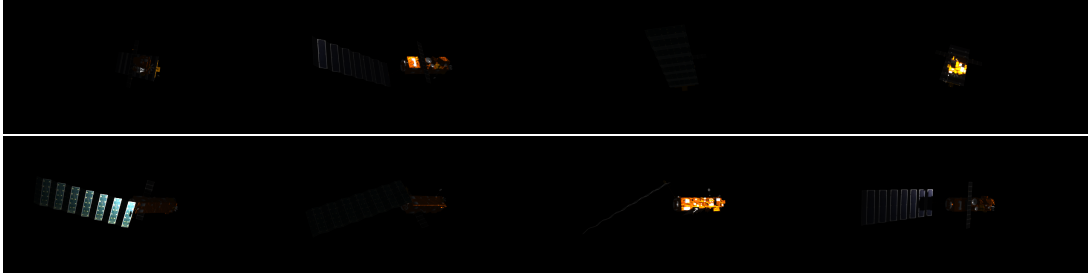
Experiments were conducted on both synthetic and laboratory datasets to validate the proposed method. Table 5.1 summarises the experiments conducted in this chapter.

The coarse pose estimation module was built in MATLAB and the **GMM** Bayesian classifier was implemented using the **GMMBayes** toolbox.² The fine pose estimation module was developed in C++, the **OpenCV** library (version 3.0) was used for computer vision and image processing related functions. Keyframes are generated using the open-source 3D computer graphics software **Blender** (version 2.78). All simulations are carried out on an Intel® Core™ i7-6700 @ 3.40 GHz \times 8 core central processing unit (**CPU**), 16 GB RAM system.

²<http://www.it.lut.fi/project/gmmbayes>

Table 5.1: Summary of experiments in Chapter 5.

Section	Description	Dataset
Section 5.4.3	Evaluation of the coarse pose estimation module	BLENDER, SPEED
Section 5.4.4.1	Evaluation of the fine pose estimation module	ASTOS, UASL, SPEED

**Figure 5.5:** Randomly sampled images from the BLENDER dataset.

5.4.1 Datasets

The datasets used to evaluate the performance of the proposed method are described below.

Blender Dataset The BLENDER dataset (Fig. 5.5) is a series of synthetic images of Envisat using the open-source 3D computer graphics software Blender by uniformly sampling the viewsphere with a mesh step Δ_{mesh} equal to 1 deg, a class step Δ_{class} equal to 10 deg, at a constant radius equal to 50 m, yielding 648 classes with approximately 100 samples per class and over 64 400 in total.³ For each rendered image, two illumination sources are added: a constant, uniform, low-intensity lighting emulating Earth’s albedo; and a high-intensity lighting with a randomly varying direction to emulate different Sun angles. All images feature a black, deep-space background, where the target is the only object present in the FOV. This is to allow for a straightforward binary segmentation of the shape prior to the computation of the ZMs. Images are rendered at a resolution of 640 px \times 640 px. The same CAD model and camera characteristics used in the ASTOS dataset were used (Chap. 2, § 2.5.2).

SPEED Dataset The Spacecraft PosE Estimation Dataset (SPEED) is a collection of images emulating the Hyperspectral Precursor of the Application Mission

³On the poles (0 deg and 180 deg elevation) a change in azimuth only produces an in-plane rotation and not a change in viewpoint, leading to fewer renderings in these cases and resulting in a total sample number lower than the expected $648 \times 100 = 64\,800$.

(PRISMA) rendezvous (RV) of the Tango and Mango spacecraft which was used to benchmark the entries of the ESA SPEC (Kisantal et al., 2020, Chap. 2). Overall, SPEED where the target is characterised by a cuboid shape, albeit much more compact in comparison to Envisat, leading to more ambiguous viewpoints, which could prove to be limiting for shape-based classifiers. SPEED is composed of both synthetic and laboratory-acquired data divided into train (SPEED/TRAIN, SPEED/REAL) and test (SPEED/TEST, SPEED/REAL-TEST) sets, numbering 12 000, 5, 2998, and 300 images, respectively. The images were scaled to a size of 640 px \times 400 px.

Astos Dataset ASTOS (Chap. 2) consists of 28 different multimodal rendezvous trajectories with Envisat, featuring three distinct guidance profiles, three tumbling modes, and two approach vectors. The tests in this chapter focus on the visible-imaged ASTOS/G2/R1/VBAR and ASTOS/G1/R2/VBAR trajectories which do not contain Earth in the FOV. Long-wavelength infrared (LWIR) images are not considered in this chapter since the methodology relies on feature detectors and descriptors that are affected by local changes in pixel intensities, and thus a model would only be feasible if the target’s thermal signature at inference time were known, which is unrealistic in practice.. The sequences are acquired at 10 Hz and processed at a resolution of 640 px \times 480 px.

UASL Dataset The Unmanned Autonomous Systems Laboratory (UASL) dataset is a RV trajectory with a 1:17 scaled down mock-up of Envisat acquired at Cranfield University (Chap. 1). The camera acquires images in a similar configuration to ASTOS/G2/R1/VBAR, at a constant relative distance of approximately 1.95 m. The mock-ups initial relative attitude is shifted 90 deg around the $t^{(2)}$ axis, displaying the radar-bearing face to the camera in the initial frame. The target’s rotation rate is constant and equal to 5.73 deg s⁻¹. The background is masked using the ground truth to eliminate illumination artefacts on the blackout-curtains. For training, the mock-up is modelled in Blender and textured with real images to generate the offline keyframe database.

5.4.2 Testing

The coarse pose estimation results are presented in terms of the integer difference between the true class, y , and the predicted class, \hat{y} :

$$\begin{aligned}\Delta\tilde{y}_{az} &:= \text{mod}(|\hat{y}_{az} - y_{az}|, y_{\text{hem}}), \\ \Delta\tilde{y}_{el} &:= |\hat{y}_{el} - y_{el}|,\end{aligned}\tag{5.76}$$

where y_{hem} corresponds to the class the hemisphere at 180 deg. The fine pose estimation results are presented in terms of the position and attitude error metrics, respectively:

$$\delta \tilde{t} := \|\hat{\mathbf{t}} - \mathbf{t}\|, \quad (5.77)$$

$$\delta \tilde{q} := 2 \arccos (\hat{\mathbf{q}}^{-1} \otimes \mathbf{q})_4. \quad (5.78)$$

For the filtering results, the velocity errors are defined as:

$$\delta \tilde{\nu} := \|\hat{\nu} - \nu\|, \quad (5.79)$$

$$\delta \tilde{\omega} := \|\hat{\omega} - \omega\|. \quad (5.80)$$

When evaluating on **SPEED**, a combined pose score is computed across all N test images according to the **SPEC** competition rules:

$$\delta \tilde{T}_{\text{SPEC}} := \frac{1}{N} \sum_{i=1}^N \frac{\delta \tilde{t}_i}{\|\mathbf{t}^{(i)}\|} + \delta \tilde{q}_i. \quad (5.81)$$

5.4.3 Evaluation of Coarse Pose Estimation

5.4.3.1 Simulations on Blender Dataset

Table 5.2 summarises parameters used in the current analysis. Images from the **BLENDER** dataset are grouped into bins of 10 deg in azimuth and elevation, thus defining the minimum achievable accuracy for the coarse pose classifier. This reduces the classification problem to 648 possible classes. To expand the size of the training population, an image augmentation campaign is performed, consisting in post-processing the renders by randomly applying some transformations; variations are introduced in terms of scale, in-plane rotation, perspective transforms, and binary segmentation threshold, yielding 500 images per class. To test the performance of the algorithm, a stratified k -fold cross-validation is then performed on a 80–20% train-test split.

In order to have a comparative insight of how the algorithm performs, two competitor methods are additionally benchmarked. The first competitor method is similar in the sense that it also relies on the **ZM**-based description of the target’s shape but follows a naive Bayes classification strategy, i.e. a single Gaussian is used to model each feature in the **ZM** vector in each class. The second competitor method uses local features rather than a global descriptor and is based on the bags-of-visual-

Table 5.2: Settings used for k -folds validation of the coarse pose classification module on the BLENDER dataset.

Parameter	Symbol	Units	Value
Viewsphere azimuth mesh step	$\Delta_{\text{mesh}}^{\text{az}}$	deg	1
Viewsphere elevation mesh step	$\Delta_{\text{mesh}}^{\text{el}}$	deg	1
Viewsphere azimuth class step	$\Delta_{\text{class}}^{\text{az}}$	deg	10
Viewsphere elevation class step	$\Delta_{\text{class}}^{\text{el}}$	deg	10
Total classes	---	---	648
Training images per class	---	---	500
ZM vector dimension	---	---	60
Folds	k	---	5

words (BoVW) method (Csurka et al., 2004). BoVW is briefly summarised below in Remark 5.5.

Remark 5.5: Bags-of-visual-words

BoVW (Csurka et al., 2004) is an image classification and scene recognition algorithm inspired on the bags-of-words model from the field of natural language processing. First, feature extraction is performed on the images from each class. The full set of keypoints is then iteratively clustered using k -means into a pre-defined number of compact, non-overlapping groups, forming the visual vocabulary (the bag-of-keypoints) of the full domain. Each cluster center is therefore a word of the vocabulary. The features from each image are binned according to the cluster centres of the vocabulary, yielding one fixed-size histogram of visual word occurrences per image. The histograms belonging to each class are then used to train a classifier.

The BoVW model for this analysis considers ORB+FREAK features. Since the latter is a binary keypoint descriptor, the clustering is performed using k -medoids instead. Due to the large number of classes considered, rather than clustering the full set of features directly, 10 clusters are extracted for each class and then concatenated to form the global vocabulary as done by J. Zhang et al. (2006). For similar reasons, a Bayesian classifier is used rather than an SVM. Both competitor methods are also implemented in MATLAB and trained with the same number of images.

The results are illustrated in Figure 5.6 for azimuth and elevation classification performance in terms of the histogram probability mass function (PMF) of the $\Delta\tilde{y}_{\text{az}}, \Delta\tilde{y}_{\text{el}}$ class distance errors (a value of zero in the horizontal axis represents a

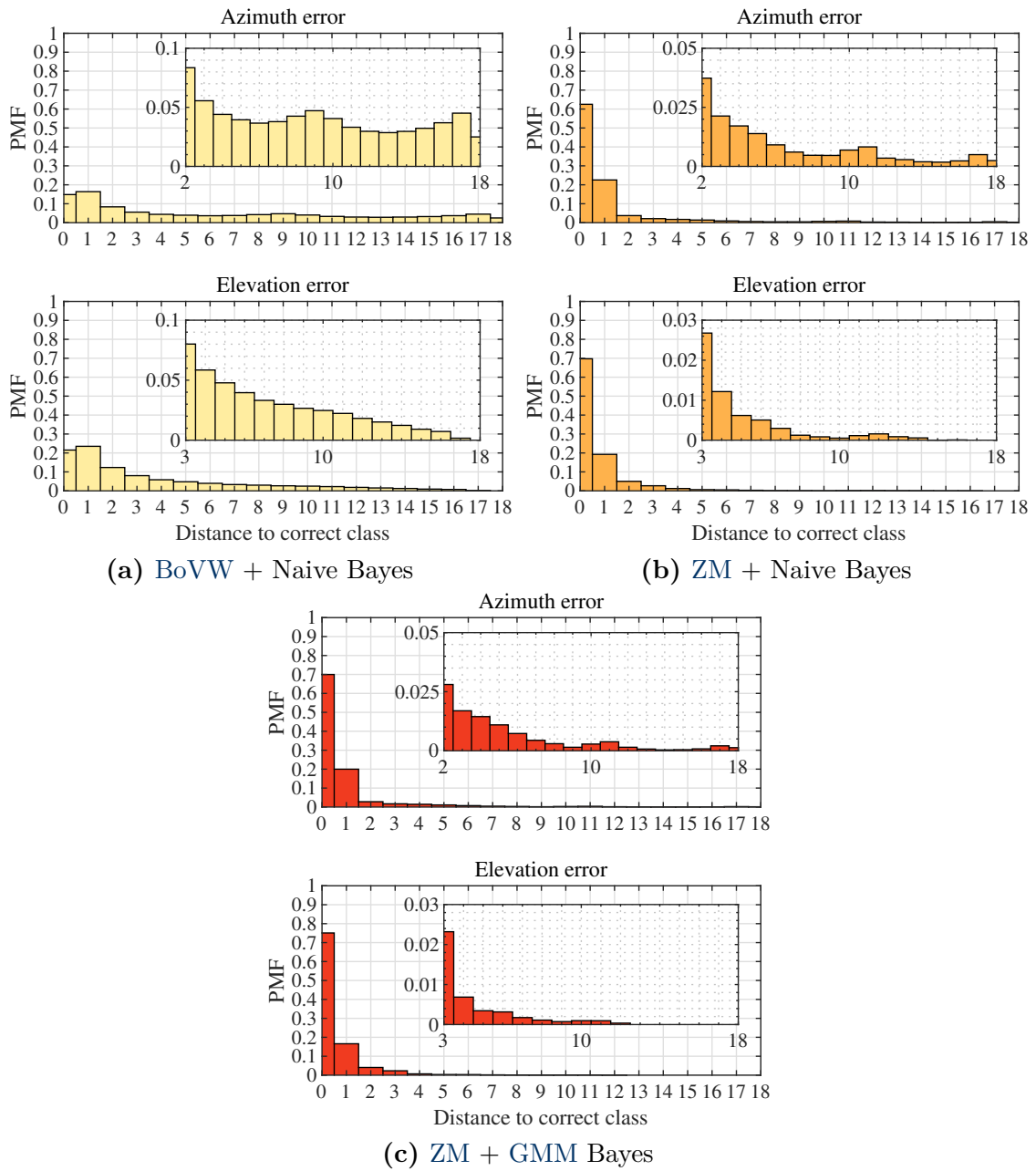


Figure 5.6: Histogram of results of the k -folds validation for the coarse pose classification on the BLENDER dataset.

Table 5.3: Average expected attitude error for the coarse pose classification on the BLENDER dataset.

Metric	Units	BoVW + Naive Bayes	ZM + Naive Bayes	ZM + GMM Bayes
$\delta\tilde{q}$	Mean	deg	76.86	14.21
	Median	deg	70.62	0.00

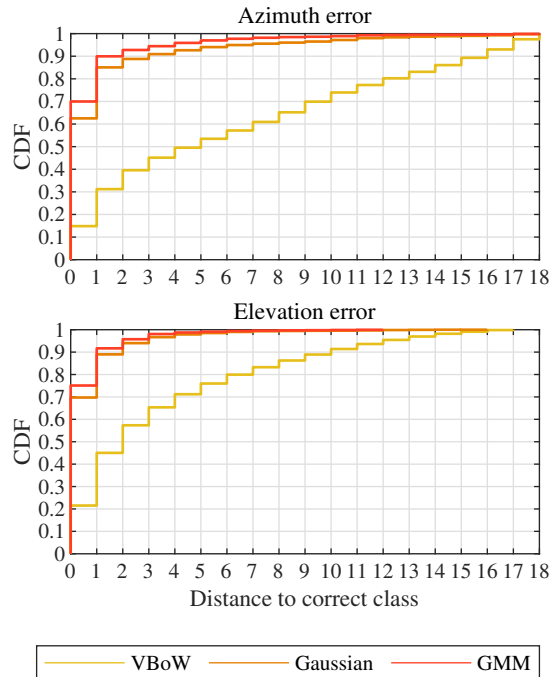


Figure 5.7: Cumulative performance of the k -folds validation for the coarse pose classification on the BLENDER dataset.

correct classification, see Eq. [5.76]). An alternative representation of the errors is shown in Figure 5.7 in terms of the (discrete) cumulative distribution function (CDF) for a better comparison of the three methods. The average expected combined attitude error is also showcased (Tab. 5.3); this is calculated by converting the azimuth and elevation from the center of each class into the equivalent predicted relative attitude quaternion $\hat{\mathbf{q}}$ (the roll angle is assumed the same for each class). The error quaternion is then computed from the ground truth and Equation (5.78).

The overall performance of the BoVW classifier is poor and vastly outperformed by the two shape-based methods. The correct classification rate for the azimuth is approximately 15%. A cumulative score of 50% is only achieved for a distance of 5 classes, i.e. a maximum average expected error of 45 deg. The results for the elevation classification are improved, which is expected, as it involves fewer classes (19 as opposed to 35 for the azimuth); this is the case for all benchmarked methods. However, the performance in this case remains far below the rest. The mean average expected attitude error is over 75 deg, making it unfit for coarse pose classification, and demonstrating that local features are not distinctive enough for the viewpoint classification of the spacecraft.

The performance of both shape-based classifiers is comparable. However, the details on Figure 5.6 expose the presence of multiple modes in the case of the naive Gaussian modelling. This phenomenon is more prominent for the azimuth error (also

Table 5.4: Mean computational execution times per image for the coarse pose classification on the BLENDER dataset.

Operation	Units	BoVW + Naive Bayes	ZM + Naive Bayes	ZM + GMM Bayes
Feature detection	ms	81.99	62.78	62.78
Inference	ms	87.75	545.58	138.93
Total	ms	169.73	608.36	201.71

present on the BoVW) and stems from the ambiguous projected shape of the target when imaged from opposing viewpoints. There is an additional error peak around the 90 deg difference in azimuth; overall this profile is consistent with a target of cuboid shape, such as the main bus of Envisat. The effect is also observable, to an extent, for the elevation error. On the other hand, these peaks have practically been mitigated in the case of the GMM. The correct classification rate is 71.35 % for azimuth and 75.86 % for elevation. These represent gains of approximately 10 % and 5 %, respectively, comparatively to the naive Gaussian modelling. In terms of average expected attitude error, this translates into an mean improvement of 5 deg. Overall, the GMM leads to 90.41 % of the data being classified with a bin distance less than or equal to 1, i.e. with a maximum expected error of 20 deg, and equivalently 92 % for the elevation.

Table 5.4 displays the computational times of the three benchmarked methods. The feature detection cost is comparable for all; note that the feature detection is the same for both the naive Bayes and GMM Bayes classifiers. For these two, the inference cost is superior, particularly in the case of the former. However, this could be attributed to the fact that a different MATLAB toolbox was used for the classification process of each. The total computational cost of the proposed GMM-based classifier is comparable to that of the BoVW, averaging 200 ms. This does not represent a significant bottleneck to the fine pose estimation module as it is only ran once, and the runtime is expected to decrease even more when implemented on a lower level programming language.

5.4.3.2 Simulations on SPEED Dataset

To offer some degree of comparison with the current state-of-the-art, the proposed coarse pose classification module is also tested on the publicly available SPEED dataset. Since the absence of a ground truth does not allow for an in-depth analysis of the performance for the coarse pose classification pipeline, the method is tested exclusively on the SPEED/TRAIN set.

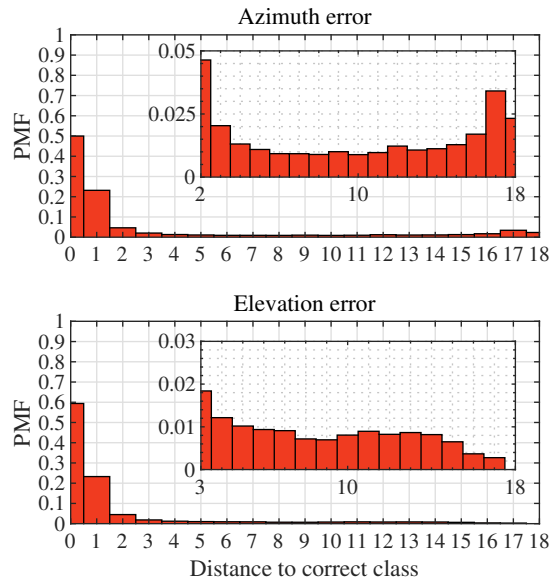


Figure 5.8: Histogram of results of the k -folds validation for the coarse pose classification on the SPEED/TRAIN dataset.

Table 5.5: Average expected attitude error for the coarse pose classification on the SPEED/TRAIN dataset.

	Metric	Units	Value
$\delta\tilde{q}$	Mean	deg	34.81
	Median	deg	10.00

Furthermore, the dataset contains some aspects related to the extraction of the target’s shape that had to be adapted in order for the current analysis to be performed. Firstly, synthetic SPEED images partly contain cases where Earth is present in the background. The proposed pipeline does not include a target-background segmentation strategy, and therefore these images have been removed prior to testing. Secondly, all images are contaminated by Gaussian noise, making image binarisation non-trivial, which leads to noisy extracted shapes and more than often to deficient segmentations, which will affect the quality of the classifier. As such, prior to binarisation, the images have to be pre-processed in an impromptu way, under which some errors still remain. This is not meant to be an optimal process, and future work will include the development of a target segmentation module to cope with these limitations.

Similarly to the analysis for BLENDER, a k -folds cross validation is performed on datasetSPEED/TRAIN using same settings from Table 5.2. The results for the proposed method in terms of the classification PMF are presented in Figure 5.8. The

Table 5.6: Pose estimation pipeline configuration and numerical settings used for the experiments on the ASTOS and UASL datasets.

Parameter	Symbol	Units	Value
Viewsphere azimuth mesh step	$\Delta_{\text{mesh}}^{\text{az}}$	deg	1
Viewsphere elevation mesh step	$\Delta_{\text{mesh}}^{\text{el}}$	deg	1
Viewsphere azimuth class step	$\Delta_{\text{class}}^{\text{az}}$	deg	9
Viewsphere elevation class step	$\Delta_{\text{class}}^{\text{el}}$	deg	9
Total keyframes	--	--	800

average expected attitude error is shown in Table 5.5. As expected, the benchmarked performance is inferior to that attained for the BLENDER dataset. Notably, there is a clear presence of a second azimuth mode at 180 deg which is not fully tackled by the GMM. Additionally, it can be seen that the elevation PMF tail also flattens out at a higher value. The correct classification rates for azimuth and elevation are approximately 50 % and 60 %, respectively. However, 75 % of the data azimuth accuracy is concentrated at a bin distance less than or equal to 1, equivalently 85 % for the elevation. The overall mean attitude error is situated at 34.81 deg. This represents a lower performance than the current state-of-the-art deep learning methods which competed in SPEC, but despite not explicitly tackling the target segmentation problem, the proposed classifier still obtains good enough results to be used as an initialisation method, as it is meant, to the fine estimation module, at a fraction of the required training and inference computational times.

5.4.4 Evaluation of Fine Pose Estimation

5.4.4.1 Simulations on Astos Dataset

In this section, the performance of the full spacecraft relative pose estimation pipeline is assessed. This includes the initialisation procedure with the coarse pose classifier followed by the pose refinement using local features. Tests are ran on the ASTOS/G2/R1/VBAR and ASTOS/G1/R2/VBAR to benchmark the robustness of the method towards two different tumbling modes and changes in relative distance.

Table 5.6 shows the parameters employed in the test. A step of 9 deg both in azimuth and elevation was chosen to build the offline database, resulting in 800 keyframes. Note that, for the simulated motions, a far lower number of keyframes would be required since the rotation is periodic; however, to stress the algorithm, the full set of possible keyframes to choose from is kept. The EKF is run with a timestep of 0.1 s (the sampling rate of the camera). The initial filter pose state $\hat{\mathbf{t}}^{(0)}, \hat{\mathbf{q}}^{(0)}$ is

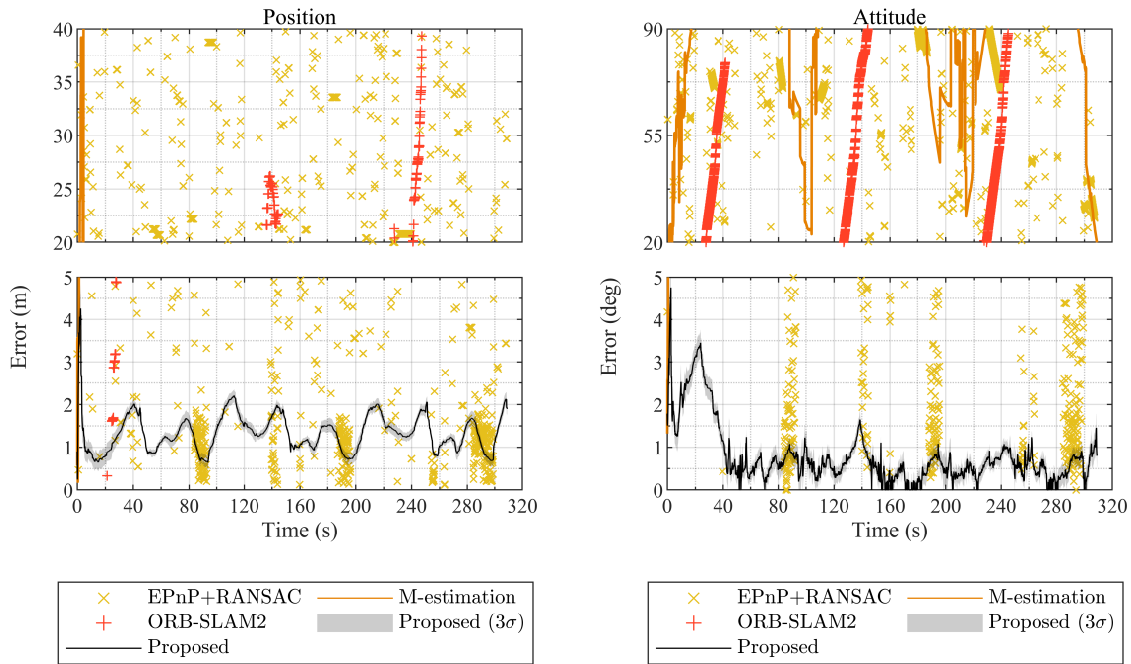


Figure 5.9: Nominal pose estimation errors for the ASTOS/G2/R1/VBAR/VIS/HOT trajectory.

initialised with the result of the coarse pose estimation, while the velocity state $\hat{\omega}^{(0)}$ is pessimistically assumed to be equal to zero. The initial covariance $\hat{P}^{(0)}$ and the process noise covariance $\sigma_v^2, \sigma_\omega^2$ are tuned empirically, whereas the measurement noise covariance is automatically determined via M-estimation.

Four different methods are compared: 1) EPnP with feature point matches (Lepetit, Moreno-Noguer, et al., 2008) and Random Sample Consensus (RANSAC) for outlier rejection; 2) the method developed in Chapter 4, using M-estimation fusing point and edge features; 3) the model-free ORB-SLAM2 (Mur-Artal et al., 2015); and 4) the framework proposed in this chapter. In the case of the first two methods, the next keyframe is determined by the pose estimated in the previous time-step. As ORB-SLAM2 is a model-free method, the first built keyframe is arbitrarily oriented and scaled; as such, for the context of this analysis, it is scaled with the corresponding trajectory ground truth.

ASTOS/G2/R1/VIS/HOT Sequence

The results of the relative pose estimation for ASTOS/G2/R1/VIS/HOT are shown in Figure 5.9. It can be seen that EPnP+RANSAC is not able to converge at all. The pure M-estimator yields a decent estimate for the first few frames, but the error quickly begins to grow until the algorithm diverges completely at time $\tau = 5$ s.

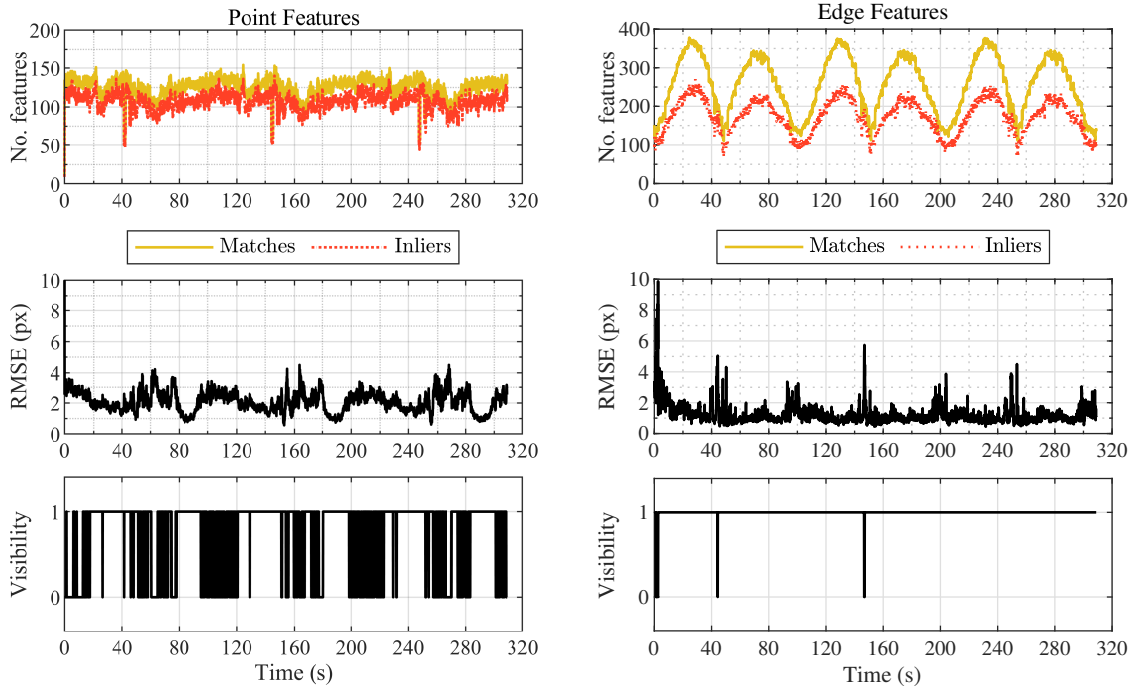


Figure 5.10: Feature statistics for nominal pose estimation sequence of the ASTOS/G2/R1/VBAR/VIS/HOT trajectory.

ORB-SLAM2 is only capable of providing a pose estimate for three segments of the trajectory, corresponding to the parts where the radar-side is facing the camera and the number of keypoints is maximal and relatively stable. Nevertheless, the estimate quickly drifts in the case of the position, and it is entirely wrong in the case of the attitude. On the other hand, the proposed framework converges at around $\tau = 10$ s. The steady state error is bounded at approximately 2 m for position, which corresponds to 4% of the range distance, whereas the attitude error is bounded at 1.5 deg. Figure 5.10 exhibits some figures of merit pertaining to the point and edge features in the simulation run, namely the number of matches and inliers, the RMSE of the M-estimation, and the feature visibility with respect to the validation gating applied prior to the filtering. A threshold of 2.5 px was applied for the points and 5 px for the edges. The number of matches fluctuates more in the case of edges; this is due to the relative circular trajectory in which the imaging area of the target changes. The peaks correspond to the sections where the $\underline{t}^{(1)} - \underline{t}^{(2)}$ plane $\in \mathcal{F}_t$ is imaged by the chaser, whereas the valleys correspond to an imaging of the $\underline{t}^{(2)} - \underline{t}^{(3)}$ plane (see Chapter 2, Fig. 2.16). However, the RMSE of the point features is on average greater than that of the edges, which results in fewer periods of visibility for the former. The periods of higher RMSE correspond to images of the $\underline{t}^{(1)} - \underline{t}^{(2)}$

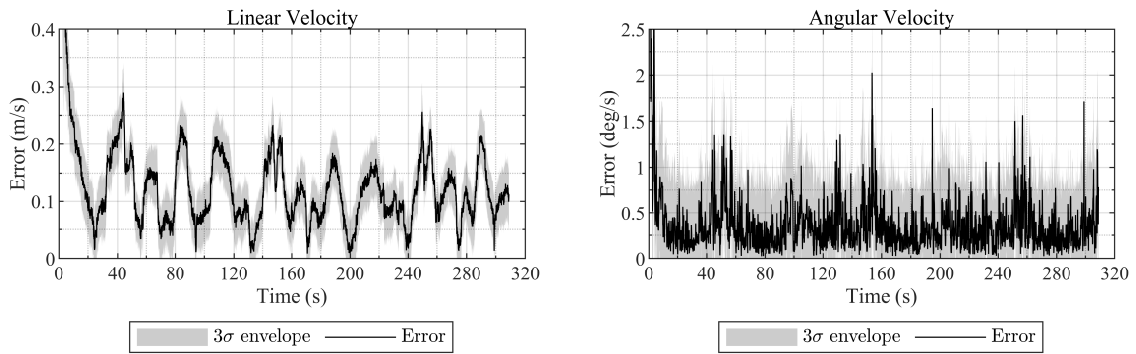


Figure 5.11: Nominal velocity estimation errors for the ASTOS/G2/R1/VBAR/VIS/HOT sequence for the proposed framework.

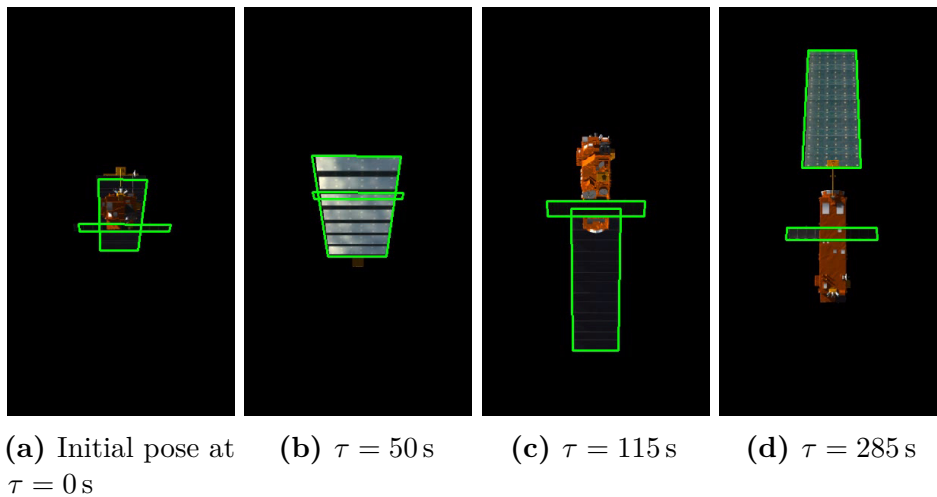


Figure 5.12: Qualitative results of the relative pose estimation for the ASTOS/G2/R1/VBAR/VIS/HOT sequence. The edges of the radar and solar panel are projected in green using the estimated pose.

plane, where the image of the target is dominated by the MLI coverage and the solar panel. Despite this, the guided feature matching algorithm prevents the point features' visibility from being constantly null during these periods.

The relative velocity estimation errors as output by the filter are also shown (Fig. 5.11). The linear velocity steady-state error does not exceed 0.3 m s^{-1} , whereas the angular velocity is bounded at 2 deg s^{-1} . The latter quantity is much noisier than the former, since there are two out-of-plane dimensions, compared to one for the linear velocity, highlighting the challenge of depth estimation with a monocular setup. Lastly, Figure 5.12 illustrates the some frames of the synthetic dataset with the estimated pose superimposed.

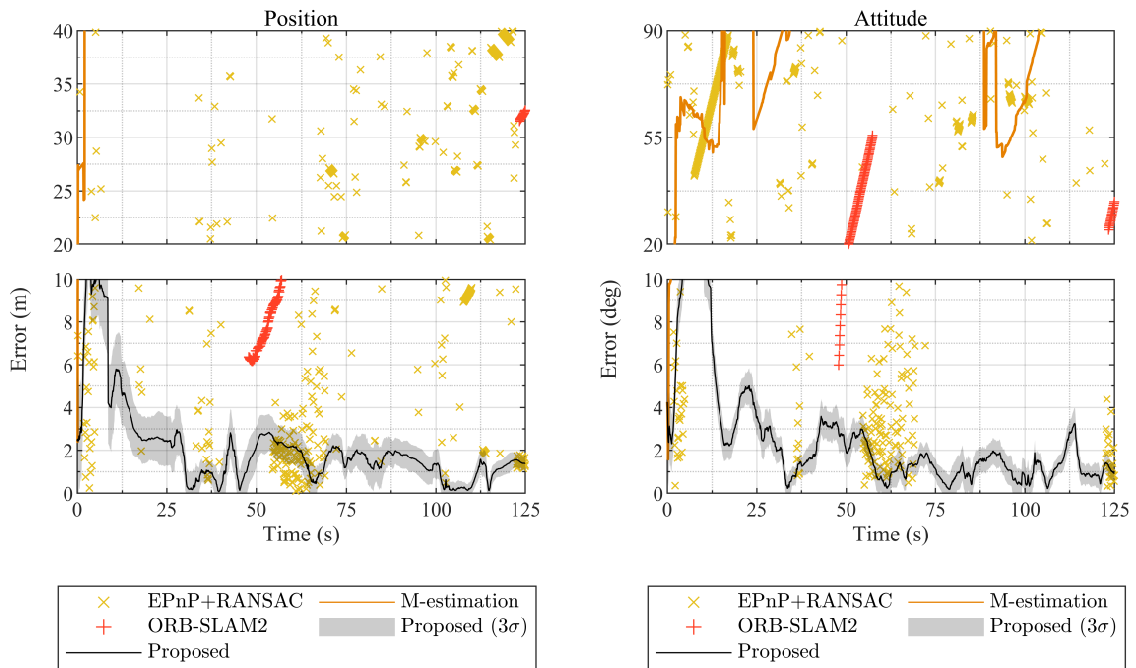


Figure 5.13: Nominal pose estimation errors for the ASTOS/G1/R2/VBAR/VIS/HOT sequence.

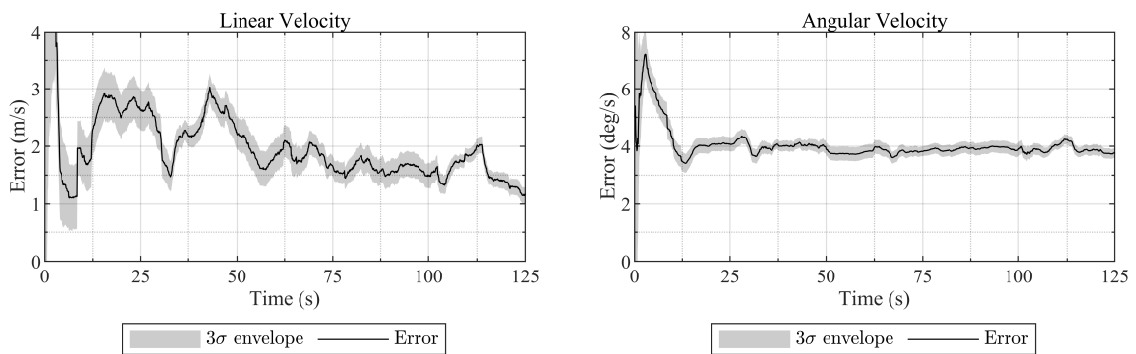


Figure 5.14: Nominal velocity estimation errors for the ASTOS/G1/R2/VBAR/VIS/HOT sequence for the proposed framework.

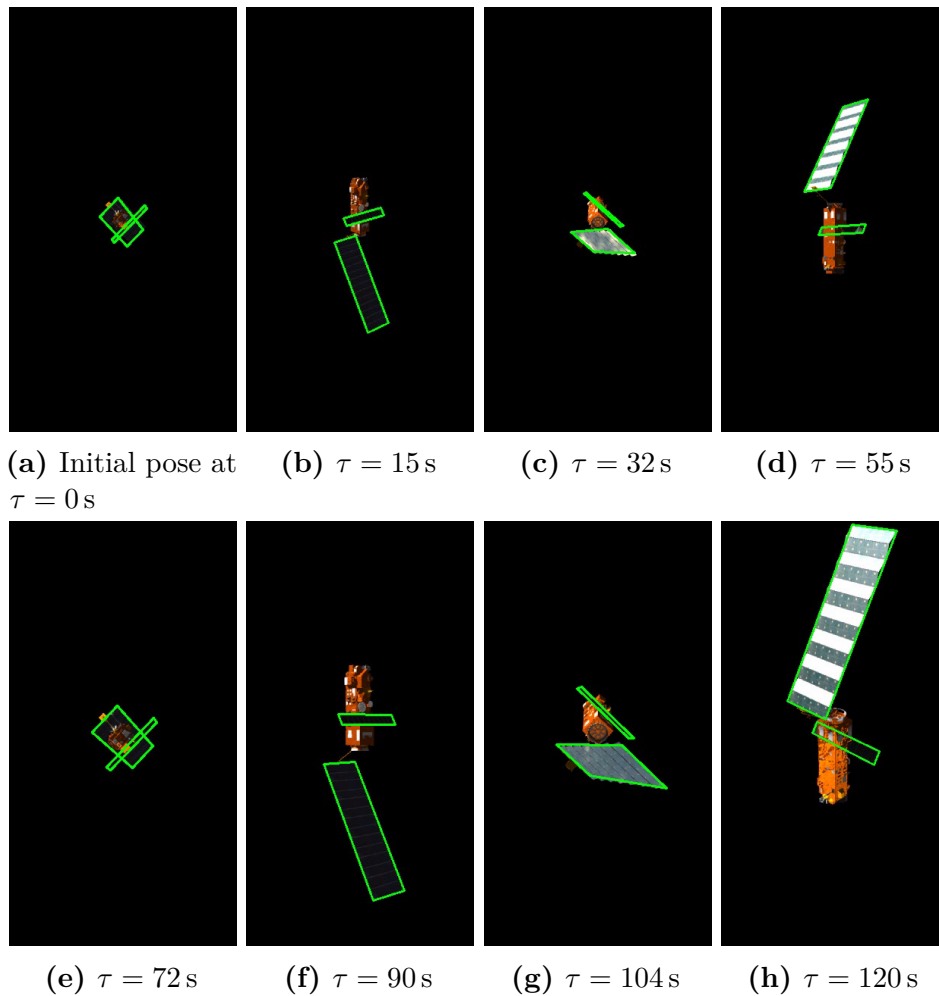


Figure 5.15: Qualitative results of the relative pose estimation for the ASTOS/G1/R2/VBAR/VIS/HOT sequence. The edges of the radar and solar panel are reprojected in green using the estimated pose.

ASTOS/G1/R2/VIS/HOT Sequence

The pose estimation results for the ASTOS/G1/R2/VIS/HOT trajectory are portrayed in Figure 5.13. This trajectory is more challenging due to the change of position depth and more complex tumbling mode. EPnP+RANSAC and the pure M-estimator, again, quickly diverge. ORB-SLAM2 is now only capable of briefly providing a solution for two sections and with unsatisfactory quality. The proposed method takes slightly longer to converge (at around time $\tau = 15$ s). From the 3σ standard deviation envelope estimated by the filter, the uncertainty is noticeably higher comparatively to the previous ASTOS/G2/R1/VBAR sequence. The steady state position error behaves similarly, not exceeding 4% of the range. The attitude error is slightly worse, reaching a peak of 5 deg right after the transient, but remaining bounded at 3.8 deg from thereon. The introduced variations in the relative motion drive the velocity errors considerably higher (Fig. 5.14): while the linear velocity error appears to decrease along with the range, the angular velocity error converges to a steady state value of 4 deg s^{-1} . In spite of this, the predictive matching module that relies on the EKF prediction remains robust enough to provide an accurate solution of the pose. Qualitative results are exhibited in Figure 5.15.

ASTOS/G2/R1/VBAR/VIS/COLD Sequence

In this section, the robustness of the proposed algorithm is evaluated during eclipse, in particular, on the ASTOS/G2/R1/VBAR/VIS/COLD sequence. For this sequence, the target spacecraft is not under direct sunlight, meaning that less light enters the camera's sensor, in turn reducing the signal-to-noise ratio. To simulate this effect, the images are corrupted with white, zero-mean Gaussian noise with a standard deviation $\sigma = 1 \times 10^{-2}$ (equivalent to 2.5 on a pixel range of 0–255). Then, to facilitate the segmentation process and increase the visibility of the target, contrast limited adaptive histogram equalisation (CLAHE) is applied as in the analysis done in Chapter 3. Since this step also now increases the sensor noise, a denoising step must be added before applying a threshold-based segmentation; the function `fastNlMeansDenoising` readily available on OpenCV is used for this purpose. The outcomes of these preprocessing steps are illustrated in Figure 5.16, along with the result of the segmentation, for the initial frame at time $\tau = \tau_0$.

The pose and velocity estimation results are shown in Figure 5.17 (cf. Figs. 5.11 and 5.9). The corresponding feature statistics are illustrated in Figure 5.18 (cf. Fig. 5.10). It can be observed that the point feature matching volume now follows a cycle entailing a much larger amplitude when compared to the hot case, where the valleys correspond to periods where the apparent area of the target in the

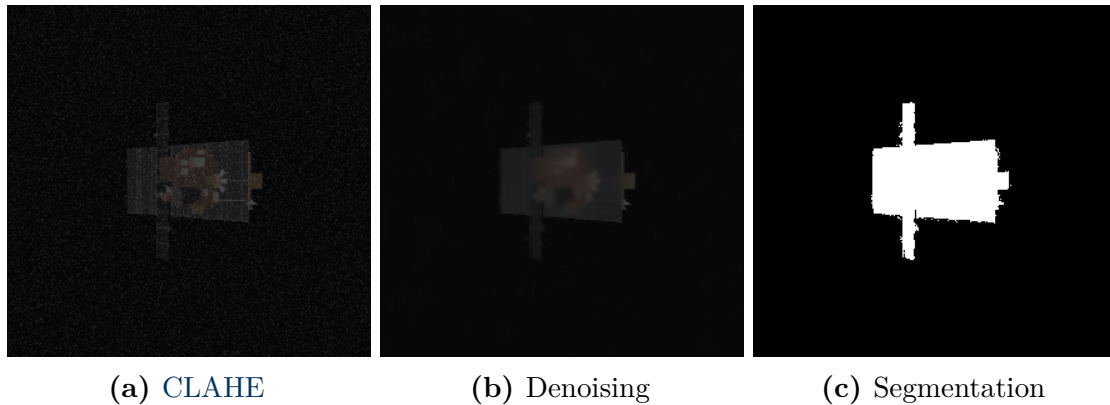


Figure 5.16: Image processing for the ASTOS/G2/R1/VBAR/VIS/COLD trajectory.

camera’s FOV is minimised. In general, fewer points are detected and matched due to the degradation from the denoising step. Despite this, the predictive matching component still manages to keep the number of inliers on par with the matches, and the RMSE-based gating successfully rejects any spurious pose pseudo-measurements. Despite the loss of quality in the segmentation process, the behaviour of the edge features remains stable as in the hot case, with no significant differences.

The position estimation error does not change much with respect to the hot case, yielding a mean value of approximately 2.75 % of range in steady-state. The maximum attitude error now reaches 2.5 deg, and showcases a mean value slightly over 0.90 deg, compared to a mean of 0.78 deg for the hot case. The linear and angular velocities also perform similarly. Overall, the proposed method is shown to be robust to eclipse conditions as long as the silhouette of the target can still be extracted. It was noted, however, that the denoising step took on average 570 ms per frame on its own to run, which represents 446 % of the complete algorithm’s runtime in nominal sunlight conditions (see Tab. 5.7), deeming such an implementation impractical as-is. Potential solutions could include running the denoising step at lower resolutions to increase speed and then upscaling the result in exchange for some performance loss, or exploring less computationally-intensive denoising algorithms, which are left as future work.

Computation Times

The mean computational cost per image of the four methods is benchmarked in Table 5.7. ORB-SLAM2 is by far the fastest method. Note, however, that for most of the benchmarking, the algorithm was unable to initialise, and thus the majority of the heavy-lifting was avoided. Conversely, the proposed method (non-optimised code) exhibits the highest cost, which is broken down in Table 5.8. It is clear that

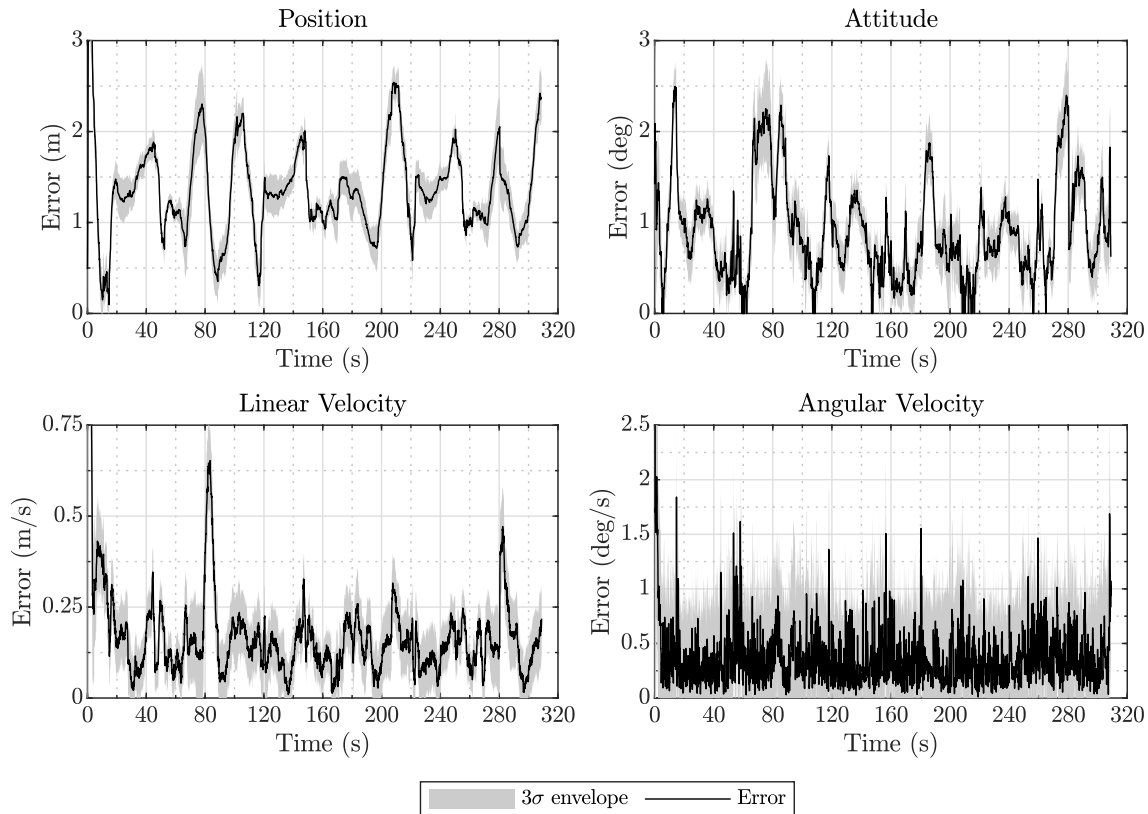


Figure 5.17: Nominal pose and velocity estimation errors for the ASTOS/G2/R1/VBAR/VIS/COLD trajectory.

Table 5.7: Mean computational execution times per image for the fine pose estimation on the ASTOS dataset.

EPnP+RANSAC	M-estimation	ORB-SLAM2	Proposed	Units
63.32	72.81	20.00	127.72	ms

the bottleneck resides in the feature extraction and M-estimation tasks. Notably, the computation of the **FREAK** descriptors takes approximately 40 ms, and the combined M-estimation routines take 55 ms to run; both make up almost 75% of the total runtime. By limiting the number of detected features, both of these figures of merit can be decreased on one go. Future work will include a trade-off analysis on the influence of limiting the number of features on the accuracy of the pipeline.

5.4.4.2 Validation on UASL Dataset

In this section, the framework is validated in laboratory; in particular, on the UASL dataset.

Figure 5.19 displays the pose estimation errors as achieved by the framework;

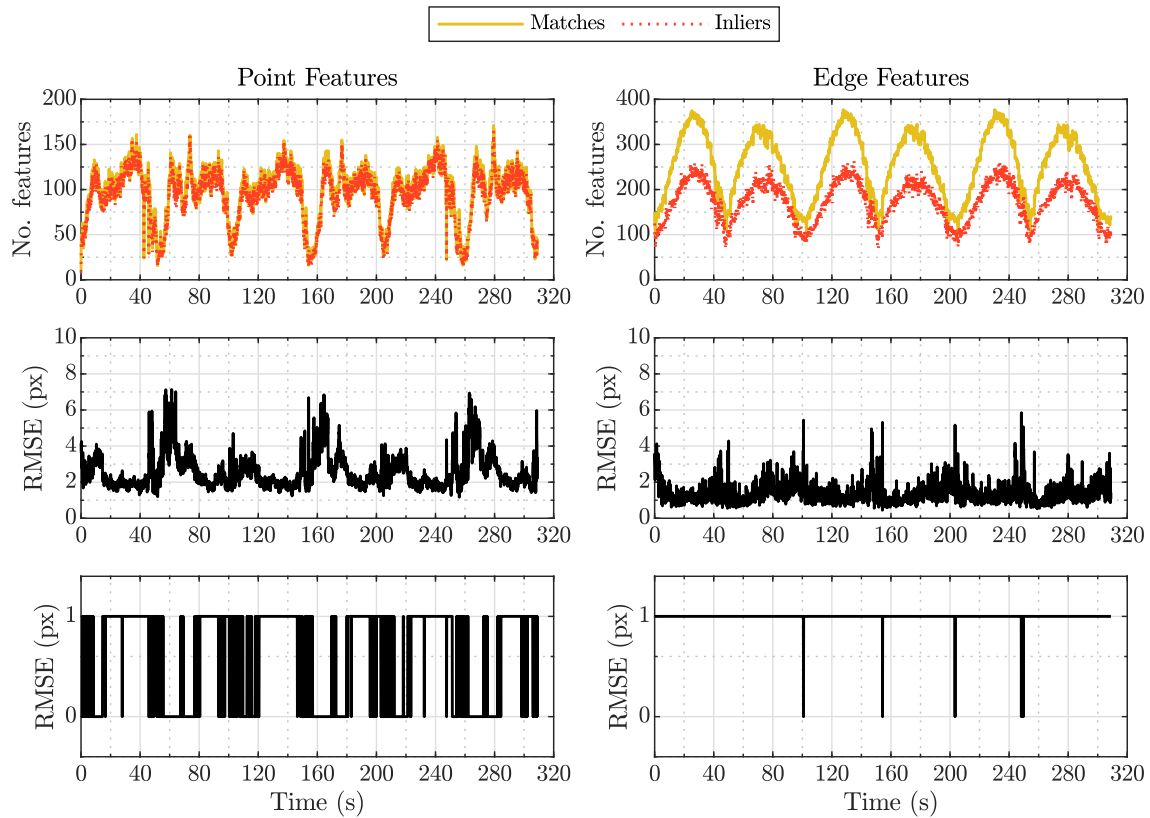


Figure 5.18: Feature statistics for nominal pose estimation sequence of the ASTOS/G2/R1/VBAR/VIS/COLD trajectory.

Figure 5.20 shows the velocity estimation errors; Figure 5.21 depicts a set of frames from the lab sequence including initialisation and reprojection of the pose. The magnitude of the attained errors is analogous to that obtained for the synthetic dataset: a maximum of 5% position estimation error with respect to the range, whereas the attitude error in steady-state does not exceed 2.5 deg (while remaining generally under 1.5 deg). As expected, the angular velocity error estimation is more noisy than the linear velocity one.

5.4.4.3 Validation on SPEED Dataset

Despite the analysis of the coarse pose estimator done on the synthetic SPEED/TRAIN dataset, it would be interesting to obtain a more direct comparison with the state-of-the-art by assessing the pose estimation performance of the proposed method on the actual test data through the SPEC score (see Eq. [5.81]).

When submitting the results on the website, the score is computed automatically for both SPEED/TEST and SPEED/REAL-TEST sets, although only the former was used to decide the winners of the competition; the latter was shown for reference and to evaluate the transferability of the algorithm to laboratory data. During

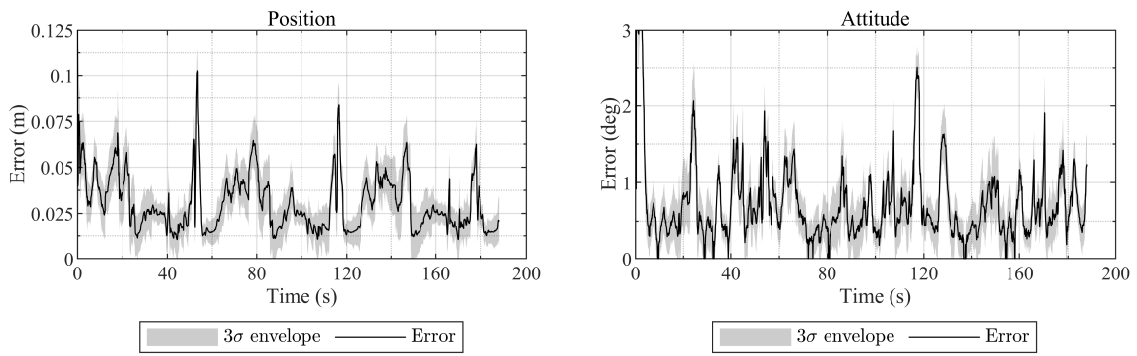


Figure 5.19: Pose estimation errors for the UASL dataset.

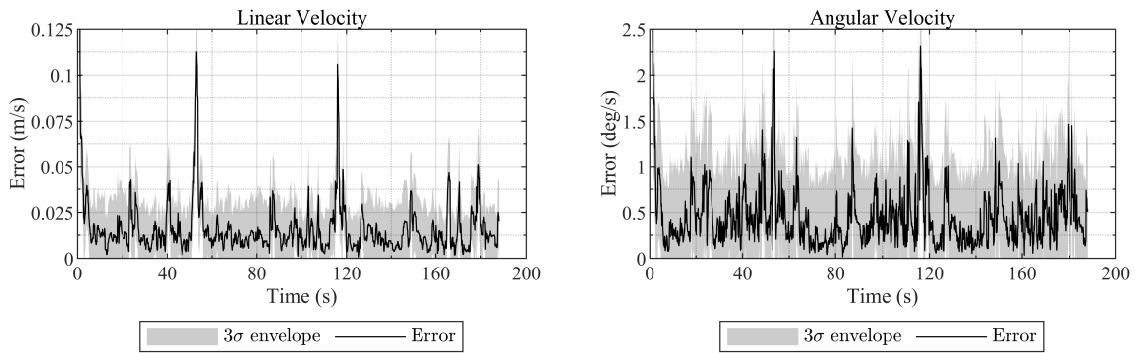


Figure 5.20: Velocity estimation errors for the UASL dataset.

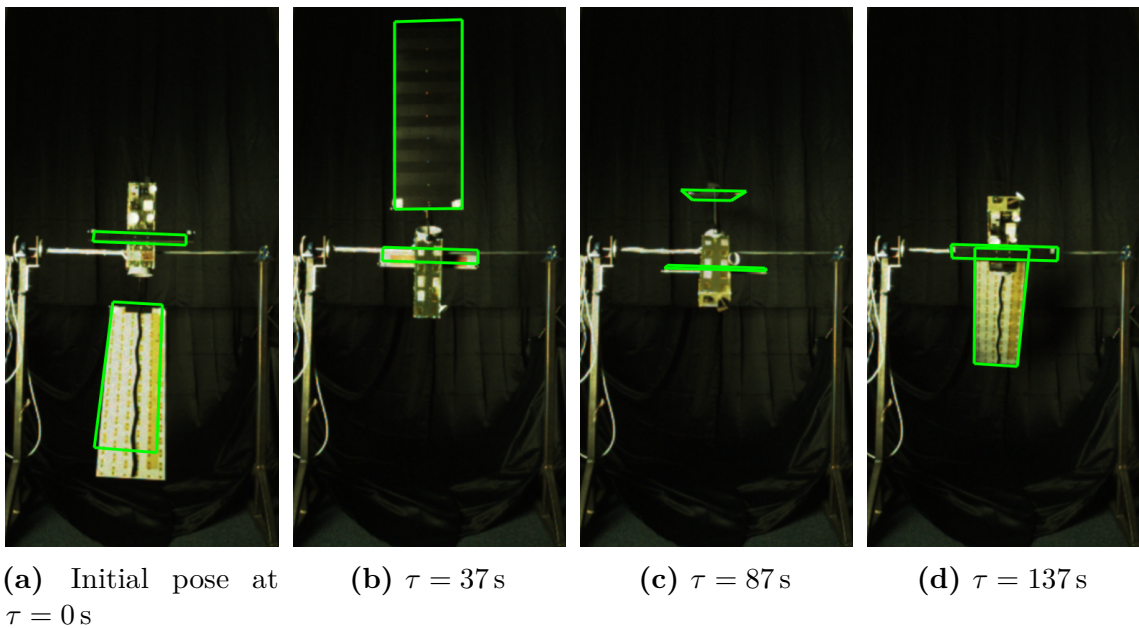


Figure 5.21: Results of the relative pose estimation for the UASL dataset. The edges of the radar and solar panel are reprojected in green using the estimated pose.

Table 5.8: Breakdown of mean computational execution times per image for the proposed fine pose estimation method on the ASTOS dataset.

Operation	Time (ms)	Time (%)
Point detection	7.84	6.14
Point description	39.14	30.65
Image binarisation	0.15	0.11
Edge detection	2.54	1.99
Point matching	0.53	0.42
Edge matching	2.32	1.82
Point M-estimation	7.83	6.13
Edge M-estimation	56.98	44.61
Keyframe selection	10.24	8.02
EKF prediction	0.06	0.05
EKF correction	0.09	0.07
Total	127.72	100.00

the competition, submissions were evaluated on a subset of all test images (also undisclosed) in order to avoid overfitting. At the end of the competition, the submissions were re-evaluated on the complete test sets and ranked accordingly. It is still possible to obtain a score on this subset by submitting the estimated 6-DOF pose values on the SPEC website. Since SPEED/TEST also contains images having Earth in the background, which the present algorithm does not tackle, the performance is assessed for SPEED/REAL-TEST alone (i.e. the “real image score”).

The CAD model of Tango has not been provided for SPEC. Therefore, for this section, the 3D structure of the spacecraft was first reconstructed using a few selected images from SPEED/TEST using multi-view triangulation from manually selected keypoints and the provided 6-DOF relative pose (Hartley and Zisserman, 2004). This was achieved using the MATLAB Computer Vision Toolbox, yielding the corresponding set of 3D structural points, which were then imported to Blender and used as the reference to model Tango’s geometric primitives and to texture the object. The reconstructed model was then used to render a number of keyframes covering the attitude range of the train set, SPEED/REAL. Figure 5.22 illustrates some sample keyframes rendered from this reconstructed model. As the resulting viewsphere is much more reduced compared to the evaluations in Section 5.4.3, values of $\Delta_{\text{class}}^{\text{az}} = \Delta_{\text{class}}^{\text{el}} = 5 \text{ deg}$ are used instead (cf. Tab. 5.2), yielding a total of 12 possible classes for coarse pose determination.

The complete framework achieves a real image score $\delta\tilde{T}_{\text{SPEC}} = 0.2692$. A qualitative illustration of the results can be observed in Figure 5.23. Despite featuring a black

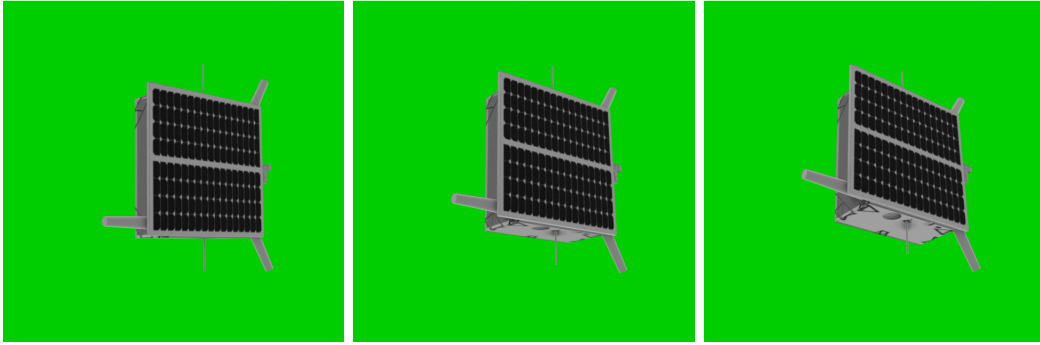


Figure 5.22: Sample keyframes rendered from the reconstructed target spacecraft of the Spacecraft Pose Estimation Dataset (SPEED).

Table 5.9: Achieved SPEED/REAL-TEST subset $\delta\tilde{T}_{\text{SPEC}}$ score in the context of the scores obtained by the SPEC top-5 rankers in this metric.

Method	Username	Score	SPEC Final Rank
EPFL	EPFL_cvlab	0.1040	2
University of Surrey	pedro_fairspace	0.1476	3
Proposed	dr_uas1	0.2692	n/a
Stanford University	stanford_slab	0.3221	4
University of Adelaide	UniAdelaide	0.3634	1
Motoki Kimura	motokimura1	0.5714	6

background, the images from SPEED/REAL-TEST were found to contain some artefacts that made a typically straightforward threshold-based segmentation sub-optimal. Nonetheless, the algorithm is shown to generate a robust estimate of the relative pose (figure 5.23a). In some cases, the algorithm converges to a local minimum due to the target being partially outside of the FOV, which affects either the performance of the coarse module or the fine module, or both (figure 5.23b). However, these are a minority, and the attained $\delta\tilde{T}_{\text{SPEC}}$ is well below the Pytorch and Keras baseline scores of 2.6636 and 3.5359, respectively, provided by the SPEC organisers using deep learning.

The obtained score is set side-by-side to those achieved by the top-5 SPEC participants on the same dataset in Table 5.9; details about the methods can be found in (Kisantal et al., 2020). It is clear that the proposed framework is comparable to the best scores on the SPEED/REAL-TEST subset, ranking third place. Notably, it is better than that achieved by the University of Adelaide (0.3634), the winners of the competition based on their SPEED/TEST score. It is reiterated that only synthetic data has been used for training, which demonstrates the robustness of the algorithm to the domain gap. For context, note that four out of the five competitor

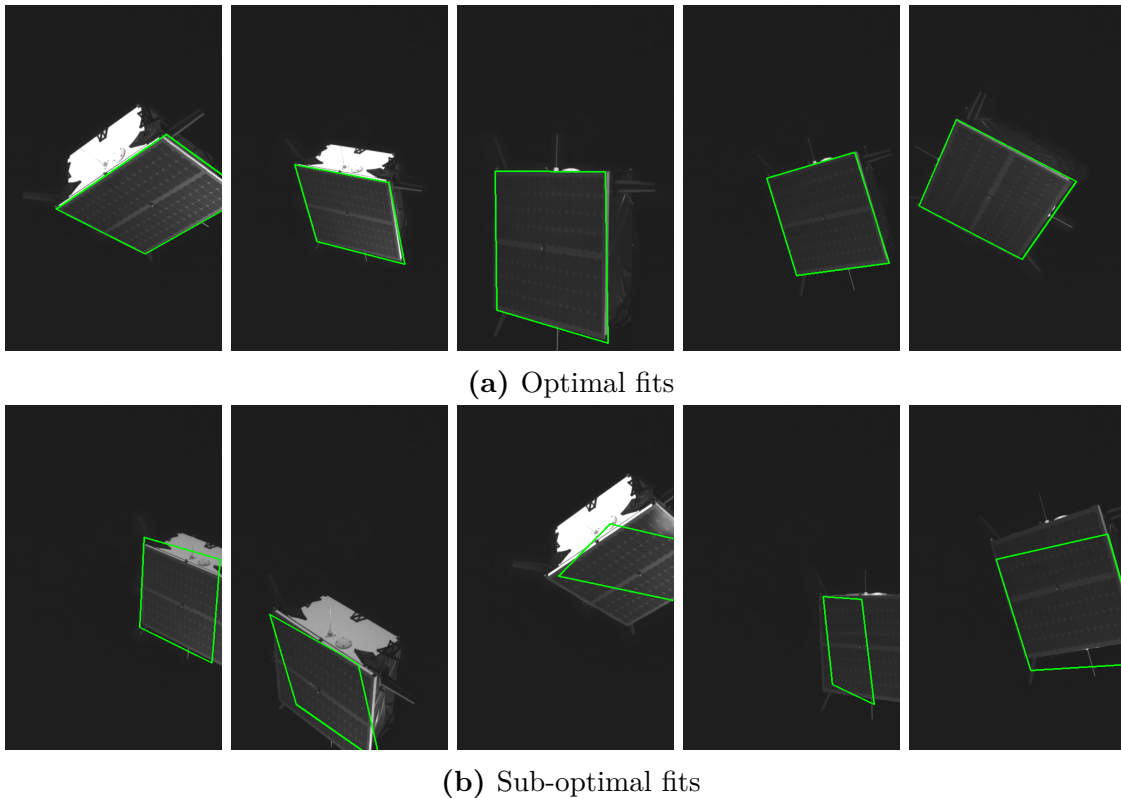


Figure 5.23: Results of the relative pose estimation for the `SPEED/REAL-TEST` subset. The edges of the solar array are reprojected in green using the estimated pose.

methods illustrated in Table 5.9 are confirmedly based on deep learning frameworks. Additionally, teams `UniAdelaide`, `EPFL_cvlab`, and `stanford_slab` all rely also on the ground truth 3D structural points of the target, which are used to train neural networks that detect their 2D coordinates in images; the actual pose is solved using *PnP*. At the time of submission, the result ranked fourth place on the post-mortem leaderboard in terms of real image score, although no details about the competing submissions are known.

5.5 Conclusions and Future Work

In this chapter, a robust, innovative model-based solution for spacecraft relative navigation using a single visible wavelength camera has been developed. The proposed contribution stands on the fact that the relative navigation solution is achieved using a set of discrete keyframes rendered in an offline stage from a three-dimensional model of the target spacecraft, where a 3D-2D problem is converted into a 2D-2D approach relying only on computer vision methods which do not require hardware acceleration such as GPUs. The proposed method was validated using synthetic

datasets closely simulating the imaging conditions experienced in-orbit, including scale changes and tumbling motion of the target, and a real dataset generated in laboratory featuring the complex spacecraft Envisat. The aspect of the target in each keyframe is learned using shape features and **GMMs** that are used to train a Bayesian classifier; the coarse viewpoint as seen by the camera was identified approximately 90 % of the time with an error under 20 deg on the showcased synthetic dataset. The classifier was also tested on synthetic images from the open-source **SPEED** dataset, where it was found that a combination of sensor noise and low lighting conditions negatively affect the target shape extraction, bringing the performance down to 75 % for 20 deg bounds.

The pose estimate is refined by matching hybrid features between the current query image and train keyframe. Automatic outlier rejection is assured via M-estimation. An **EKF** is employed to fuse the pose hypotheses generated by each feature type; it is designed to operate on the tangent space of $SE(3)$, allowing it to seamlessly integrate the previous stage by assimilating the covariance matrices generated by the M-estimation directly as the measurement noise, independently of the pose parametrisation. The attained solutions for both synthetic and laboratory datasets targeting Envisat under sunlit orbits showcase rapid convergence and yield a maximum error of 5 % of the range distance in position and 3.8 deg in attitude; on average, steady-state errors are observed in the order of 2.5 % of the range in position and 1 deg in attitude. A novel matching strategy by predicting feature locations using the **EKF**, alongside a **RMSE**-based validation gate, assures the stability and accuracy of the solution is maintained, even in the face of highly discrepant frames with respect to the database keyframes caused by light-scattering **MLI** and solar array reflections. This stability was also ascertained for eclipse periods, although the noise removal processing for low-light images significantly increased the overall execution runtime.

Lastly, the proposed pipeline was validated on laboratory test images of **SPEED**, obtaining a pose score calculation of 0.2693, which demonstrates its robustness in bridging the domain gap between synthetic and real data and is comparable to the best scores obtained in the **SPEC** competition with deep learning. While the advances in the latter field towards computer vision tasks such as image classification are undeniable, the results achieved herein cast doubts on the belief that any deep learning-based technique is automatically capable of achieving a lower error than classical spacecraft relative pose estimation methods. It is noted, though, that the method is dependent on a proper extraction of the shape of the target and hence the score for synthetic images featuring Earth in the background could not be computed,

something that could be easily surpassed using deep learning. Future work will thus focus on the development of a dedicated segmentation module to enhance the framework.

CHAPTER 6

Pose Estimation for Multimodal Sequences via Deep Recurrent Convolutional Learning

This chapter presents a method to estimate the relative pose of a spacecraft by incorporating the temporal information from a rendezvous sequence into a deep learning pipeline. It leverages the performance of long short-term memory (LSTM) units in modelling sequences of data for the processing of features extracted by a convolutional neural network (CNN) backbone. To improve end-to-end pose estimation by regression, a difficult problem due to the vast response domain (especially for $SO(3)$), the complete framework, dubbed ChiNet, combines three distinct training strategies along a coarse-to-fine funnelled approach, facilitating feature learning. The capability of CNNs to automatically ascertain feature representations from images is used to fuse infrared data with red-green-blue (RGB) inputs.

6.1 Motivation

WITH recent advances in computing power, neural network-based algorithms have evolved from traditional networks containing one to three hidden layers toward deep networks capable of having hundreds. As each layer includes nonlinear activation functions, building deeper networks allows for more accurate approximations to the intricacies of complex environments. This stands as a clear advantage with respect to more traditional approaches involving the linearisation of systems, which demands significant resources in terms of modelling and is limited to particularly favourable conditions.

Autonomous vision-based spacecraft navigation is one key area with the potential of largely benefiting from deep neural network (DNN) estimation methods. Since the

introduction of compact and lightweight passive optical sensors as feasible on-board instruments, the focus has been on the development of robust image processing (IP) and machine learning (ML) techniques to accurately estimate the target spacecraft's relative state, typically through the six degrees-of-freedom (6-DOF) pose. Deep learning can be considered the natural next step in this regard, as such methods would adequately capture the intrinsic nonlinearities between the input sensor data and the state estimates (see Chap. 2). Additionally, convolutional neural networks (CNNs; LeCun et al., 1989) are naturally tailored to process such image inputs. To cope with the harsh conditions that are characteristic of on-orbit operations and the space environment in general, and also to provide redundancy to the guidance, navigation and control (GNC) system, other sensors capable of producing image-like inputs such as thermal cameras and lidar can be fused in the solution and fully availed using CNNs.

As researched in Chapters 3 and 4, and ultimately demonstrated in Chapter 5, applying traditional IP methods to space scenarios is a fruitful, albeit arduous, task as algorithms must be carefully crafted and adapted to cope with challenges such as a tumbling target and unfavourable illumination conditions. A further advantage of applying DNNs translates into bypassing this step; the image processing task is shifted completely to the network, and the effort becomes concentrated towards parameter optimisation and data modelling, potentially allowing for the generalisation of the model to a wider swath of imaging conditions. Despite deep learning models requiring large amounts of data to yield an acceptable accuracy, existing data can be artificially augmented to increase the sample size while simultaneously optimising for robustness; in the space domain specifically, synthetic imaging datasets can be created using physics rendering engines that adequately simulate the environment. On the other hand, while for traditional ML-based models performance with respect to amount of training data eventually reaches a plateau, research suggests that for deep learning methods this relationship increases logarithmically (C. Sun et al., 2017).

The popularity of deep learning for computer vision tasks exploded in the early 2010s due to the admirable performance of the newly rediscovered CNN-type architecture for image classification in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) relative to classical ML-based techniques (Krizhevsky et al., 2012). Since then, the state-of-the-art has advanced in the direction of increasingly deeper CNNs, which nowadays often reach tens of millions of parameters (He et al., 2016; Szegedy, W. Liu, et al., 2015). Near the end of the decade, it permeated onto the field of spacecraft relative pose estimation for rendezvous, mainly due to

the Satellite Pose Estimation Challenge (SPEC), which ran in 2019, where the vast majority — if not all — of the competitors used some kind of deep learning-based approach. Despite traditional techniques outperforming most of such approaches under the right conditions, as the results from Chapter 5 have shown, the competition did highlight clear advantages of using deep learning in relative navigation for space (Kisantal et al., 2020). For instance, if trained well, a CNN will almost certainly learn to extract features from the target and ignore the background, eliminating the need for a specialised segmentation block if Earth appears in the field of view (FOV) of the chaser’s camera.

SPEC benchmarked the participating algorithms on the Spacecraft Pose Estimation Dataset (SPEED; see Chap. 5), which consists of images of the Tango satellite generated under randomised poses. As such, most candidate approaches tackled the problem with CNNs. However, during a rendezvous sequence, it is expected that the pose of the observed target continually varies as the operation progresses, i.e. the poses are correlated through time. This chapter proposes the use of a recurrent neural network (RNN) module to process the features extracted by a CNN front-end model and exploit this temporal correlation between acquired image frames in the rendezvous sequence. The resulting deep recurrent convolutional neural network (DRCNN) architecture, dubbed ChiNet,¹ is shown to provide a smoother and lower-error estimate of the 6 degree-of-freedom (DOF) pose when compared to a single CNN. Furthermore, ChiNet proposes a three-step training regimen to learn features in a coarse-to-fine manner, which is inspired from the approaches of Chapters 4 and 5 in traditional ML. Lastly, ChiNet also explores the impact of multimodal sensing in the pose estimating by augmenting the number of input channels to the network with images from a long-wavelength infrared (LWIR) camera, thus exploiting the natural ability of CNNs to autonomously extract features from images.

Remark 6.1: Associated Publications

This chapter is based partly on the following published work:

- [J3] D. Rondao, N. Aouf, and M. A. Richardson (2021). “ChiNet: Deep Recurrent Convolutional Learning for Multimodal Spacecraft Pose Estimation”. In: *IEEE Transactions on Aerospace and Electronic Systems*. Manuscript in submission

¹Pronounced “kai-net”, the first term is an abbreviation of “chimera” (from the Greek “Χίμαιρα”), meaning “something made up of parts of things that are different from each other”.

6.2 Related Work

Deep learning-based computer vision techniques applied to space relative navigation saw a modest and deferred beginning when compared to the drastic revamp of the more general research field brought about by Krizhevsky et al.'s (2012) AlexNet: early practices focused on entry, descent and landing (EDL) employing DNN architectures varying from two-layer multilayer perceptrons (MLP) to CNNs with a limited number of layers to tackle problems such as selecting optimal landing sites for spacecraft (Campbell et al., 2017; Lunghi et al., 2016) or predicting the fuel-optimal control actions to perform the landing itself (Furfaro et al., 2018). Eventually, more modern CNN architectures started to be adopted (Silburt et al., 2018).

The first use of DNNs for spacecraft relative pose estimation, concretely, was proposed by Sharma, Beierle, et al. (2018), who leveraged a pre-trained AlexNet to approach the determination of the $SO(3)$ state as a classification rather than regression. The last fully connected (FC) layers were retrained using synthetic images of the Tango spacecraft flown in the Hyperspectral Precursor of the Application Mission (PRISMA) mission by acquiring snapshots of the target at different azimuths and elevations, with the addition of zero mean Gaussian white noise. The method was shown to yield a better accuracy than a baseline method using classical pose estimation from 2D-3D point correspondences through EPnP (Lepetit, Moreno-Noguer, et al., 2008) and Random Sample Consensus (RANSAC; Fischler and Bolles, 1981), but deemed not fine enough for any application other than a coarse initialisation. Later on, the authors improved their original work by introducing the Spacecraft Pose Network (Sharma and D'Amico, 2019), which used a five-layer CNN backbone taking $224 \text{ px} \times 224 \text{ px}$ image inputs, connected to three different output branches: the first one used the Faster R-CNN architecture (Ren et al., 2017) to detect the bounding box of the target; the second classified the relative attitude in terms of a probability distribution over discrete classes; and the last branch took the top-rated candidates from the previous to learn a weighed combination that refined the attitude. A coarse estimate of the relative position was obtained from the bounding box and refined using a Gauss-Newton algorithm and four control points reprojected from the target model onto the image. The network was initially trained on the ImageNet dataset, and then the branch layers were retrained on the SPEED dataset, which was also introduced in the paper (see Chaps. 2 and 5).

Around the same time, the authors open-sourced the SPEED dataset through a collaboration with the European Space Agency's (ESA) Advanced Concepts Team which culminated in SPEC. As reported by Kisantal et al. (2020), the majority of the participating teams developed architectures that predict the relative pose

of the target in an end-to-end, regressive fashion, i.e. the input is an image and the output is directly in $SO(3) \times \mathbb{R}^3$. This is a desirable design decision since it minimises the modularity of a DNN, facilitating both training and testing processes. However, the top scorers embarked on alternative approaches to achieve a lower estimation error. Namely, four teams trained CNNs to predict, in each image, the 2D locations of 3D pre-selected model points, and then used a perspective- n -point (PnP) technique to retrieve the pose from the correspondences (Chap. 2, § 2.1.4). On average, this approach yielded an improvement of 74.7% for the position estimation and 86.5% for the attitude estimation. A second procedure consisted in using a separate localisation step to predict a bounding box around the target and processing only the sub-image. This, in turn, was shown to improve the attitude estimation, but not the position. The winners of the competition, Chen et al. (2019), combined both approaches: the localisation was performed using an HRNet (K. Sun et al., 2019) and Faster-RCNN combination; the processing of the cropped and resized region of interest was done with a pure HRNet trained by minimising a mean squared error (MSE) loss between the predicted heatmaps and ground truth heatmaps of the visible landmarks in each image. An initial pose hypothesis was then extracted via PnP and RANSAC and refined with Levenberg-Marquardt (LM). The runner-up, a team from the Swiss Federal Institute of Technology in Lausanne who did not publish their results, followed a similar approach. Interestingly, the team that achieved third place investigated the potential of end-to-end deep pose estimation (Proença and Gao, 2019). The authors employed a ResNet architecture with some of the higher layers replaced with additional convolutions to keep spatial feature resolution, where the position was learned through the minimisation of the relative error, as they allege it allows the loss weights to better generalise to other datasets. Regarding attitude estimation, the authors compared a direct regression on the quaternion angular error against a classification approach, similar to Sharma and D’Amico (2019), with an extra probabilistic Gaussian mixture modelling (GMM) step to resolve potentially ambiguous outputs. Initial testing demonstrated that the attitude classification method outperformed the regression-based one, and the team went with the former as their submission. After SPEC, published work continued to focus on individual images, either greyscale or RGB, and did not stray too far from the competition’s findings in terms of innovation (Cassinis et al., 2020; Harvard et al., 2020; Oestreich et al., 2020).

Indeed, spacecraft relative pose estimation solutions have mostly not ventured beyond the visible wavelength either. Yilmaz, Aouf, Majewski, et al.’s (2017) model-free solution used simultaneous localisation and mapping (SLAM) based on interest

point tracking on thermal imagery for active debris removal (ADR). However, the lack of texture, variations in temperature, and complex relative motion makes long term feature tracking and navigation very difficult. Additionally, supplementary information must be given to solve the scale ambiguity inherent to monocular systems (Chap. 2, §. 2.1.3). Model-based approaches are also challenging under the scope of classical ML techniques since the local aspect of an object in the LWIR band depends on the temperature of its components, which is time-variant in-orbit due to changing exposure to sunlight. Added to the fact that these are difficult to model accurately, thermal signatures are not a reasonable choice for model-based estimation and hence particular attention must be paid to which features to select. J.-F. Shi, Ulrich, Ruel, and Anctil (2015) applied the SoftPOSIT algorithm (David et al., 2004) for simultaneous feature correspondence and pose estimation of corners from a simplified 3D model of the target, as these are a constant property of its shape. However, the proposed algorithm is dependant on a good initialisation and no consistency between frames is enforced. Gansmann et al. (2017) minimised the distance between the reprojected edge contour of a training image and the one detected by the thermal camera, but it assumed the target faced the chaser in a constant way and hence could only track translational motion. Both of these techniques only make use of shape information, and hence do not fully exploit the benefit of using the LWIR band over or in conjunction with the visible. None of these techniques make use of deep learning.

The processing of time-series data in deep learning saw its origin with the so-called “vanilla” RNN (Rumelhart et al., 1986, Chap. 2, § 2.4.2.5). However, the propensity of these cells towards vanishing and exploding gradients made them unable to learn long-term sequences. The introduction of the LSTM cell by Hochreiter and Schmidhuber (1997), designed in terms of a gated architecture with a bypass system along the temporal axis, help solve this issue, and today they are ubiquitous in many sequence modelling tasks, including handwriting and speech recognition, machine translation, and image captioning (I. Goodfellow et al., 2016). LSTMs have recently been combined with features extracted by CNN front-ends to model the intrinsic motion dynamics from sequences of imaging data rather than individual inputs. Concretely, VINet (Clark et al., 2017) and DeepVO (S. Wang et al., 2017) have proposed DRCNNs for visual odometry (VO) to estimate a car’s egomotion from image sequences from the KITTI autonomous driving dataset (Geiger et al., 2013). The output of the networks is the inter-frame Lie algebra element $\xi^\wedge \in \mathfrak{se}(3)$ of the tangent space to the pose, trained on an MSE loss, which is then chained with previous estimates to track the global estimate of SE(3). The latter network

works with raw images only, while the former combines them with the output of inertial measurement units (IMU). Kechagias-Stamatis et al. (2020) introduced DeepLO, which followed the same philosophy for lidar-based relative navigation with an uncooperative space target. The authors pre-process lidar data by quantising and projecting it into each plane in the \mathcal{F}_t frame of reference, thus creating three 2D depth images that can be processed by a regular CNN. Due to the rich information contained in the depth images, the full pipeline avoided a large number of layers. The SO(3) representation was chosen to be the rotation matrix minimised directly over an MSE loss.

The research developed in this chapter is similar in spirit to that of Proença and Gao (2019) as a DNN is used to directly compute the relative pose of a target spacecraft, and to that of Kechagias-Stamatis et al. (2020) as the network is a combined DRCNN pipeline. However, the resulting architecture is trained exclusively on image inputs, and hence the following contributions are proposed, to the best of the knowledge yielded by the current literature survey:

- (1) This work represents the first use of RNNs, in particular LSTMs, to tackle the problem of spacecraft pose estimation for rendezvous using on-board cameras as the sole sensor;
- (2) It is also the first to explore the potential benefit of a multimodal sensor input for the task, in particular in the visible and LWIR modalities; and its influence in challenging orbital illumination conditions; and
- (3) A three-step approach to DNN training is devised to facilitate the learning and reduce the overall estimation error.

6.3 Methodology

The goal of this chapter is to train an end-to-end deep neural network to learn the relative pose of a target spacecraft from on-board image sequence inputs provided by a chaser. The results from SPEC (Kisantal et al., 2020) have shown promising results in the use of CNNs for the task; however, the current literature treats each incoming image as a separate input, thus ignoring the intrinsic temporal correlation between them. Therefore, the main focus is the investigation of the feasibility of a DRCNN for estimating the pose in rendezvous sequences. The problem has been previously studied by Kechagias-Stamatis et al. (2020) for lidar map inputs, but not for images. A second oversight of SPEC — and of SPEED in particular — is that the data does not feature reflective satellite materials (e.g. multi-layer insulation

[MLI]) or low lighting conditions, such as eclipse crossings; two conditions that can highly affect the performance of visible wavelength cameras. This work does not only discuss these factors, but additionally proposes its own multimodal advance to tackle them through the integration of LWIR images with the traditional RGB input. Recent contributions have also highlighted the benefit of transfer learning to initialise the deep CNN layers that are typically used as front-end feature extractors and benefit from being trained on large datasets. However, the habitual go-to means to achieve this, ImageNet, consists of RGB images, and as such is not extensible to multimodal problems. To bridge this gap, an initial domain-specific stage trained on an artificial oversampling of the dataset and a coarse form of the objective loss is proposed. Conversely, in an attempt to reap the benefits of keypoint-based estimation, a post-processing refinement stage trained on a reformulation of the loss in terms of feature reprojection error is implemented.

Figure 6.1 illustrates the approach at a high level. The resulting architecture is described in detail in Section 6.3.1. The proposed multistage optimisation strategy is presented in Section 6.3.2. Lastly, data augmentation techniques are illustrated in Section 6.3.3.

6.3.1 Architecture

Whereas previous CNN-based approaches focus on retrieving the spacecraft relative pose from a single image, ChiNet realigns the problem back into the classical formulation of sequence-based approaches while concurrently leveraging recent advances in deep learning to achieve simultaneous feature extraction and time series modelling. Mathematically, previous methods focus on maximising $p(\mathbf{T}^{(\kappa)} | \mathbf{x}^{(\kappa)})$, where $\mathbf{T}^{(\kappa)}$, $\mathbf{x}^{(\kappa)}$ are the relative pose and features extracted by a CNN, respectively, at time $\tau = \tau_\kappa$, and this work proposes instead to maximise the conditional probability of the current pose given features extracted from all previous inputs, i.e. $p(\mathbf{T}^{(\kappa)} | \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\kappa)})$.

The pipeline receives a monocular four-channel red-green-blue-thermal (RGBT) image sequence as input. At each time-step, the pixel intensity values of the multimodal frame are normalised per-channel to the interval $\{-1, 1\}$, and fed to a CNN that autonomously learns an optimal, reduced-dimension feature representation. These features are then passed to a RNN module for time series modelling: during training, temporally-ordered sequences of features are fed to the recurrent cells, and at inference time, each cell processes one frame at a time using the learned recurrent and non-recurrent weights to build and propagate an internal state representation that takes all previous inputs into account to return the pose. The details on the

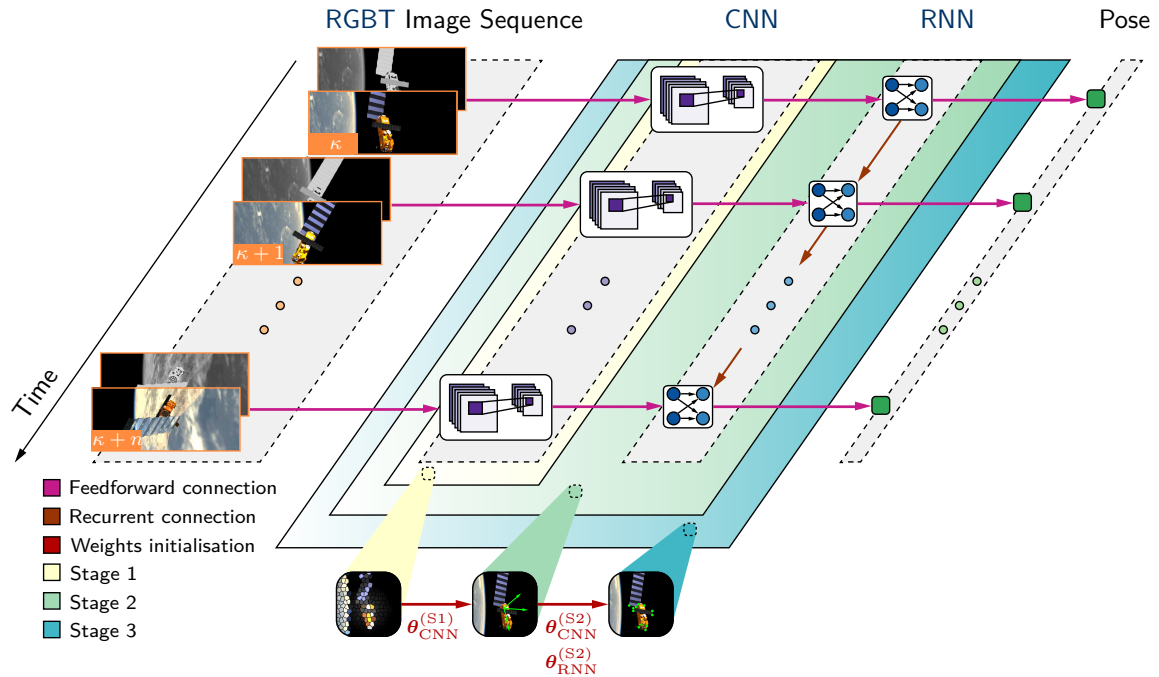


Figure 6.1: ChiNet system overview. The proposed deep recurrent convolutional neural network (DRCNN) architecture performs end-to-end spacecraft pose estimation from a sequence of multimodal red-green-blue-thermal (RGBT) image inputs. The training is structured into a funnelled coarse-to-fine three-stage procedure, where the convolutional neural network (CNN) front-end first learns singly a rough estimate of the pose. The remaining two phases, encompassing both the CNN and a recurrent neural network (RNN), focus on refining the previously learned objective to produce a more accurate solution.

CNN and RNN modules are described in this section.

6.3.1.1 Deep Feature Extraction with CNNs

CNN front-ends for feature extraction are typically chosen to be large but powerful architectures, such as ResNet (He et al., 2016) or Inception-v3 (Szegedy, Vanhoucke, et al., 2016) and the submissions to SPEC were no exception. On the other hand, these networks are also characterised by elevated processing times and are potentially prone to overfitting due to their high number of parameters. More recently, Redmon and Farhadi (2017, 2018) have proposed the YOLO object detector, which introduced a backbone termed Darknet, a CNN reportedly faster than ResNet while being on par with it for object classification tasks. The Darknet architecture is schematically illustrated in Figure 6.2. The Darknet-53 variant (here the suffix denotes the number of convolutional layers) in particular (represented on the left) is more efficient than ResNet-101 and ResNet-152 with similar classification performance to the latter. For reference, Proença and Gao (2019) use ResNet-50.

To further reduce the likelihood of overfitting, ChiNet adopts the Darknet-19

connections are introduced but only in the channel expansion-contraction layers (residual blocks in Fig. 6.2), thus avoiding the need to add 1×1 convolutions to keep the dimensions consistent. Lastly, the final convolution uses a 1×1 kernel (i.e. it behaves like a FC layer), so a dropout layer (Hinton et al., 2012) with probability $p = 0.5$ is added to further prevent overfitting.

Optimal Low-Level Sensor Fusion

Sensor data fusion for the processing of multimodal images can be tackled in multiple manners. A tracking solution (or high-level) fusion applies parallel relative navigation solutions on features extracted separately from each modality and the results are combined to form a new, fused solution. The outcome should aim to produce directly a lower pose estimation error than the solutions attained from each individual counterpart.

Another approach entails feature (or mid-level) fusion: features are detected separately in images produced in each modality and then combined to form new features, with which pose estimation is performed. The scheme should aim to produce new features with enhanced properties (e.g. extra repeatability, better matching scores, see Chap. 3) when compared to its individual counterparts.

ChiNet opts instead to adopt a third approach consisting in pixel (or low-level) fusion, where images are acquired separately by each camera and then combined to form a new, multimodal image, upon which feature extraction, target detection, and pose estimation are performed. The scheme aims to produce new images with enhanced properties (e.g. extra robustness to noise, less sensitive to lighting changes, etc.) when compared to its individual counterparts. This philosophy has been previously explored in VO applications using traditional IP techniques such as intensity level thresholding and discrete wavelet transforms, showing promising results (Poujol et al., 2015). For the CNN-based approach, the visible and LWIR images are concatenated along the channel dimension, forming a four-channel RGBT image which is then fed to the network. The first convolutional layer entails a weighted sum of the pixels in each channel, outputting new activation maps that effectively encompass the fused information. Furthermore, these weights are not predefined but learned in the context of the network training procedure, thus being optimal in the sense of minimising the objective loss. This approach therefore bypasses the need of manually developing a potentially sub-par weighing strategy to combine the multiple input modalities.

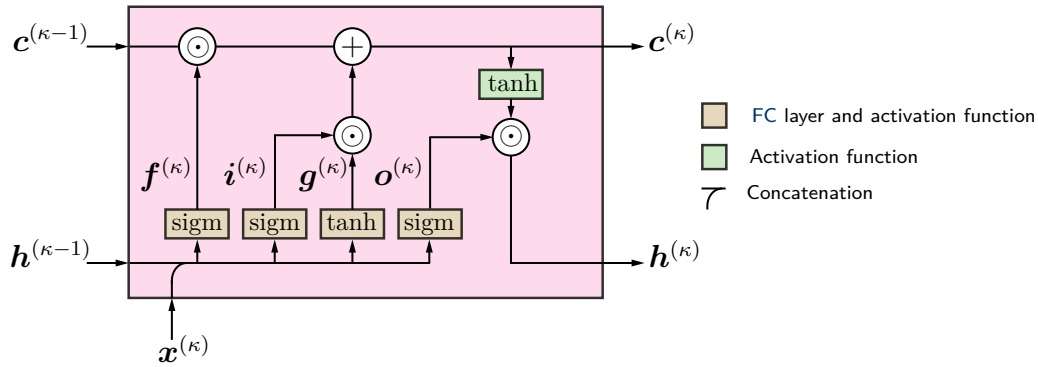


Figure 6.3: Block diagram of a long short-term memory (LSTM) recurrent memory unit. Units, or cells, are connected recurrently through their hidden state \mathbf{h} and cell state \mathbf{c} . Input features \mathbf{x} are combined with \mathbf{h} and subject to four different gates (linear layers followed by an activation): the forget gate \mathbf{f} controls the information that is kept in \mathbf{c} ; the input \mathbf{i} and modulation \mathbf{g} gates define the new information that will be added to \mathbf{c} ; the output gate \mathbf{o} filters the cell state information that will be transmitted.

6.3.1.2 Temporal Sequence Modelling with LSTMs

The features learned by the CNN are post-processed by a deep RNN module that models the intrinsic temporal correlations coming from an ordered sequence of image inputs. This addition is expected to be beneficial to the problem of spacecraft pose estimation due to the inherent relative motion dynamics entailed, and the estimate of the solution for the current frame can benefit from the knowledge of previous frames: even more than in ground-based applications, the perceived motion of a space target during rendezvous is not likely to change abruptly but is a smooth function of the previous states. An analogy could be traced in reference to the Kalman filter’s motion model and sensor update, which then persist onto the next time-steps via the Kalman gain (Chap. 5), except that a RNN’s hidden state is learned and not explicitly modelled.

As introduced in Chapter 2, LSTMs (Hochreiter and Schmidhuber, 1997) were designed in an attempt to combat vital flaws in the capability of vanilla RNNs to model long sequences, as they suffered from vanishing and exploding gradients. The LSTM’s ability to learn long-term dependencies is owed to its gated design that determines which sectors of the previous hidden state should be kept or discarded in the current iteration. This is achieved not only in combination with the current input, processed by four different units, but also by a cell state which acts as an “information motorway” that bypasses the cells. The LSTM structure is illustrated in Figure 6.3. The update equations can be compactly written as:

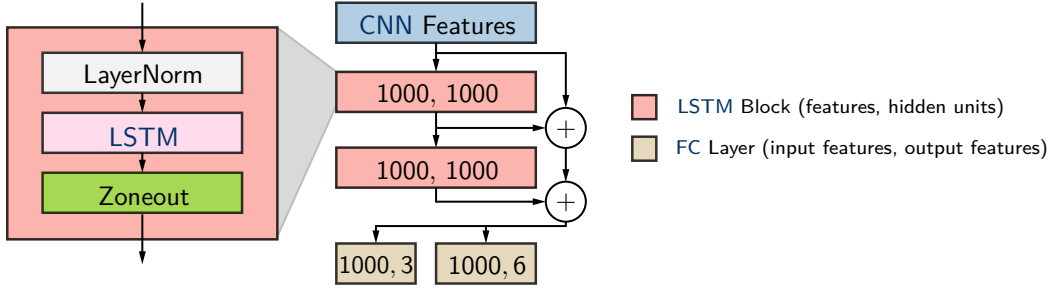


Figure 6.4: ChiNet recurrent module design. The features previously extracted by the convolutional neural network (CNN) front-end are processed by two long short-term memory (LSTM) stacks with 1000 hidden units each. The module is enhanced with residual connections, allowing gradients to flow through it directly, mitigating exploding or vanishing events. The recurrent layers are appended with two fully connected (FC) layers that produce an estimate of the relative position and a six-dimensional attitude representation.

$$\begin{bmatrix} \mathbf{f}^{(\kappa)} \\ \mathbf{i}^{(\kappa)} \\ \mathbf{o}^{(\kappa)} \\ \mathbf{g}^{(\kappa)} \end{bmatrix} = \mathbf{W}^h \mathbf{h}^{(\kappa-1)} + \mathbf{W}^x \mathbf{x}^{(\kappa)} + \mathbf{b}, \quad (6.1)$$

$$\mathbf{c}^{(\kappa)} = \mathbf{f}^{(\kappa)} \odot \mathbf{c}^{(\kappa-1)} + \mathbf{i}^{(\kappa)} \odot \mathbf{g}^{(\kappa)}, \quad (6.2)$$

$$\mathbf{h}^{(\kappa)} = \mathbf{o}^{(\kappa)} \odot \tanh(\mathbf{c}^{(\kappa)}), \quad (6.3)$$

where $\mathbf{f}, \mathbf{i}, \mathbf{o}, \mathbf{g}$ are the forget, input, output, and modulation gates, respectively, \mathbf{h} is the hidden state, \mathbf{x} is the input, $\mathbf{W}^h = [\mathbf{W}^{hf^\top} \ \mathbf{W}^{hi^\top} \ \mathbf{W}^{ho^\top} \ \mathbf{W}^{hg^\top}]$ is the recurrent weights matrix, $\mathbf{W}^x = [\mathbf{W}^{xf^\top} \ \mathbf{W}^{xi^\top} \ \mathbf{W}^{xo^\top} \ \mathbf{W}^{xg^\top}]$ is the input weights matrix, $\mathbf{b}^\top = [\mathbf{b}^{f^\top} \ \mathbf{b}^{i^\top} \ \mathbf{b}^{o^\top} \ \mathbf{b}^{g^\top}]$ is the bias vector, sigm is the sigmoid nonlinear activation function, \tanh is the hyperbolic tangent activation function, the superscript (κ) denotes a variable at time-step $\tau = \tau_\kappa$, and \odot denotes an element-wise product operation.

Recurrent Module Design

The design of the recurrent feature post-processing module is schematically depicted in Figure 6.4. The CNN features are fed to two stacked LSTM layers with 1000 hidden states each, which then branch off into two FC layers of output size equal to 3 and 6 that produce position and attitude estimates, respectively. Stacked LSTM layers have been previously adopted for architectures such as DeepVO (S. Wang et al., 2017) and DeepLO (Kechagias-Stamatis et al., 2020) and shown empirically

to help in modelling complex motion dynamics.

Due to the internal workings of the LSTM cell, pre- and post-processing operations common in FC or convolutional layers are not directly applicable. For instance, data normalisation must be performed inside the cell to establish coherence with respect to the gates' activation functions. Since the incoming data is sequential, ChiNet relies not on batch normalisation to centre the inputs of the LSTM, but on layer normalisation (J. L. Ba et al., 2016):

$$\text{LN}(\mathbf{a}^{(i)}) = \gamma_i \frac{\mathbf{a}^{(i)} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta_i, \quad (6.4)$$

where the mean μ_i and variance σ_i^2 are computed across all the features of the i -th layer rather than the batch dimension (cf. the batch normalisation Eq. 2.101 in Chap. 2, § 2.4.2.6), and ϵ is a small numerically stabilising term. Applying batch normalisation to a RNN would require fitting one layer per time-step and storing the statistics of each one during training, which would be impractical both in terms of time and memory consumption.

A second nuanced aspect pertains to dropout, typically applied as a binary mask to randomly nullify some of a layer's activations. In the case of LSTMs, however, stochasticity should be applied in the recurrent loop. More than that: rather than following a potentially naive dropout philosophy, ChiNet employs zoneout (Krueger et al., 2017), which was specifically designed for RNNs. In zoneout, the values of the hidden state $\mathbf{h}^{(\kappa)}$ and memory cell $\mathbf{c}^{(\kappa)}$ are randomly expected to either maintain their previous value or are updated in the usual manner according to Equations (6.2) and (6.3).

The modified LSTM equations thus become:

$$\begin{bmatrix} \mathbf{f}^{(\kappa)} \\ \mathbf{i}^{(\kappa)} \\ \mathbf{o}^{(\kappa)} \\ \mathbf{g}^{(\kappa)} \end{bmatrix} = \text{LN}(\mathbf{W}^h \mathbf{h}^{(\kappa-1)}; \gamma_1, \beta_1) + \text{LN}(\mathbf{W}^x \mathbf{x}^{(\kappa)}; \gamma_2, \beta_2), \quad (6.5)$$

$$\mathbf{c}^{(\kappa)} = \mathbf{d}^{c,(\kappa)} \odot \mathbf{c}^{(\kappa-1)} + (\mathbf{1} - \mathbf{d}^{c,(\kappa)}) \odot (\mathbf{f}^{(\kappa)} \odot \mathbf{c}^{(\kappa-1)} + \mathbf{i}^{(\kappa)} \odot \mathbf{g}^{(\kappa)}), \quad (6.6)$$

$$\mathbf{h}^{(\kappa)} = \mathbf{d}^{h,(\kappa)} \odot \mathbf{h}^{(\kappa-1)} + (\mathbf{1} - \mathbf{d}^{h,(\kappa)}) \odot (\mathbf{o}^{(\kappa)} \odot \tanh(\text{LN}(\mathbf{c}^{(\kappa)}; \gamma_3, \beta_3))). \quad (6.7)$$

where \mathbf{d}^c , \mathbf{d}^h are the binary cell and hidden state zoneout masks, respectively, and $\mathbf{1}$ is a vector of ones of appropriate length.

The final design choice for the recurrent module pertains to the inclusion of

residual connections, drawing inspiration from the CNN front-end itself (§ 6.3.1.1). Since these connections allow, by definition, gradients to flow directly through the network, bypassing the activation functions (which are contributors to the exploding or vanishing gradient problem), LSTMs and RNNs in general can presumably benefit from them. During preliminary experiments, it was found that the addition of residual connections to the LSTMs in ChiNet resulted in faster training convergence and overall lower pose estimation error.

6.3.2 Multistage Optimisation

Instead of pursuing an indirect approach in which a DNN is used to localise keypoints on the target which are then subject to a PnP procedure to extract the relative pose, ChiNet provides an end-to-end, direct method to retrieve it. The former, though, has been shown to produce the lowest error estimates in SPEC, suggesting that the latter may be harder to train. To mitigate this and lower the overall error in end-to-end approaches, a multistage, coarse-to-fine approach is proposed, taking inspiration from the classical ML strategies presented in the previous chapters. This section describes the optimisation strategy.

6.3.2.1 Stage 1

Stage 1 stems directly from the concept introduced in Chapter 4, and later consolidated in Chapter 5, in which the attitude space $SO(3)$ is divided into a spherical grid of discrete azimuth $\Delta_{\text{class}}^{\text{az}}$ and elevation $\Delta_{\text{class}}^{\text{el}}$ steps, centred on the target, which is imaged at a fixed distance r (i.e. a 2-sphere $S^2(r)$, or viewsphere). This effectively simplified the attitude estimation problem in the previous chapters by reducing the search to a discrete set $\mathbb{Y}_{S^2} = \{1, \dots, K\}$, with each K possible class corresponding to an ordered pair $\{\theta_{\text{az}}, \theta_{\text{el}}\}$ of azimuth and elevation angles in $S^2(r)$ that could then be used to provide a coarse solution and retrieve the closest viewpoint model image. The latter was then used to provide a rough value for the position as well.

ChiNet picks up on this approach and first trains the CNN on a simpler task to learn coarse features in terms of the discrete attitude $y_{S^2} \in \mathbb{Y}_{S^2}$ and the position depth $\|\mathbf{t}\|$ from a RGBT image \mathbf{I} . The RNN module is bypassed and the two FC layers are connected directly to the CNN's output (cf. Figure 6.4). Denoting $\mathbf{y}^{(S^2)}$ as the one-hot vector encoding of y_{S^2} , Stage 1 thus maximises the joint conditional probability:

$$\boldsymbol{\theta}^{(S1)*} = \arg \max_{\boldsymbol{\theta}^{(S1)}} p \left(\|\mathbf{t}^{(\kappa)}\|, \mathbf{y}^{(S^2, \kappa)} \mid \mathbf{I}^{(\kappa)}; \boldsymbol{\theta}^{(S1)} \right) \quad (6.8)$$

i.e. the training thus far depends only on each individual input at time $\tau = \tau_{\kappa}$, not

yet exploiting the temporal correlation in the data. The objective is to emulate the benefits of transfer learning (I. Goodfellow et al., 2016), in which the network is pre-trained on a set of tasks involving a large dataset — typically a subset of the 1000 ImageNet (Deng et al., 2009) object categories — and then used to initialise a same-sized network to solve the purported task that generally has fewer training examples. Transfer learning is advantageous for CNNs as these normally entail millions of parameters and thus may converge towards a suboptimal solution if the training data is not diverse enough.

However, ImageNet is only composed of RGB images and thus cannot be expanded for use with multimodal data. As such, a strategy to pre-train a CNN by artificially augmenting the number of samples based only on the nominal dataset is proposed, consisting of the following steps:

- (1) Discretise the attitude space into a set $\mathbb{Y}_{\mathbb{S}^2}$ of K possible viewsphere classes according to a defined mesh resolution $\{\Delta^{az}, \Delta^{el}\}$. Keep only the subset of classes $\mathbb{Y}'_{\mathbb{S}^2} = \{1, \dots, K'\} \subseteq \mathbb{Y}_{\mathbb{S}^2}$ that are represented in the dataset;
- (2) Define a number of desired observations per attitude class, $N_{\mathbb{S}^2}$;
- (3) Discretise the position into a set \mathbb{Y}_t comprised of M bins of depth values $\|\mathbf{t}\|$ of width Δ^t , selecting the edges according to the minimum and maximum observations in the dataset;
- (4) For each represented attitude class k in $\mathbb{Y}'_{\mathbb{S}^2}$:
 - (4-a) Identify the subset $\mathbb{Y}'_t \subseteq \mathbb{Y}_t$ of M' depth bins containing at least one observation;
 - (4-b) Randomly sample $N_{\mathbb{S}^2}/M'$ observations with attitude label $y_{\mathbb{S}^2} = k$ equally for each of the M' depth bins according to the position ground truth. Oversample if necessary.

The resulting Stage 1 dataset will have a total of $N_{\mathbb{S}^2}K'$ observations with equal representation according to the coarse pose bins $\{\Delta^t, \Delta^{az}, \Delta^{el}\}$. Preliminary analysis showed that having balanced attitude classes was paramount to prevent overfitting, even when resorting to oversampling, i.e. duplicating examples from minority classes to achieve a balanced number of observations. In this case, a more aggressive data augmentation procedure is employed to increase the variance of the inputs (see § 6.3.3). Both procedures are performed online and hence do not affect memory requirements.

The attitude estimation is formulated as a classification task and hence minimises the cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{\mathbb{S}^2}^{(\text{S1})} &= - \sum_i \log \text{softmax} \left(\mathbf{y}^{(\text{S}^2,i)}, \hat{\mathbf{y}}^{(\text{S}^2,i)} \right) \\ &= - \sum_i \sum_{k=1}^{K'} y_k^{(\text{S}^2,i)} \log \left(\hat{y}_k^{(\text{S}^2,i)} \right),\end{aligned}\tag{6.9}$$

where $\hat{\mathbf{y}}^{(\text{S}^2)}$ is the predicted attitude class. Similarly to Proença and Gao (2019), the position estimation is formulated as a regression task in terms of the relative depth error and minimises:

$$\mathcal{L}_t^{(\text{S1})} = \sum_i \frac{\|\mathbf{t}^{(i)} - \hat{\mathbf{t}}^{(i)}\|}{\|\mathbf{t}^{(i)}\|},\tag{6.10}$$

where $\hat{\mathbf{t}}$ is the predicted position. The complete Stage 1 loss will be a linear combination of $\mathcal{L}_{\mathbb{S}^2}^{(\text{S1})}$ and $\mathcal{L}_t^{(\text{S1})}$. The result is a multi-task learning problem that has traditionally involved the empirical tuning of the linear combination weights; this was the approach followed by S. Wang et al. (2017) for VO and by Proença and Gao (2019) for spacecraft pose estimation. This presents a difficult and expensive task, even when both objectives encompass a regression, since the position and attitude errors can be characterised by very disparate orders of magnitude. In contrast, ChiNet adopts Kendall, Gal, et al.’s (2018) approach which models each weight $\{\sigma_{\mathbb{S}^2}, \sigma_t\}$ as learnable task-specific variances of a Boltzmann distribution and a Gaussian² distribution, respectively, yielding the combined loss:

$$\mathcal{L}^{(\text{S1})} = \frac{1}{2} \mathcal{L}_t^{(\text{S1})} \sigma_t^{-2} + \mathcal{L}_{\mathbb{S}^2}^{(\text{S1})} \sigma_{\mathbb{S}^2}^{-2} + \log \sigma_t + \log \sigma_{\mathbb{S}^2}.\tag{6.11}$$

The additive terms act as regularisers to prevent the network from predicting infinite uncertainty and thus zero loss. In practice, as proposed in the original work, the indirect quantities $\log \sigma^2$ are learned instead in Equation (6.11) as the training is more numerically stable and robust to initialisation.

6.3.2.2 Stage 2

Stage 2 represents ChiNet’s nominal training phase on the normal, non-modified dataset. The full DRCNN pipeline is trained to maximise the conditional probability of a series of time-sequential poses $u^{(1)}, \dots, u^{(\kappa)}$, $u \in \mathcal{U} \cong \text{SE}(3)$ given a sequence of RGBT images, where the CNN weights are initialised with the results of Stage 1:

²Despite Equation (6.10) not strictly representing the L^2 component of a Gaussian PDF due to the division by $\|\mathbf{t}^{(i)}\|$, the formulation of Equation (6.11) yields good results in practice.

$$\boldsymbol{\theta}^{(S2)*} = \arg \max_{\boldsymbol{\theta}^{(S2)}} p \left(u^{(1)}, \dots, u^{(\kappa)} \mid \mathbf{l}^{(1)}, \dots, \mathbf{l}^{(\kappa)}; \boldsymbol{\theta}^{(S2)} \right) \quad (6.12)$$

$$\boldsymbol{\theta}^{(S2, \text{CNN})(0)} = \boldsymbol{\theta}^{(S1, \text{CNN})}. \quad (6.13)$$

Rather than estimating a joint representation of the pose, the problem is formulated again in terms of multi-task learning, maintaining the dual FC branches at the end of the network, where one regresses the position and the other the attitude. A common approach to regressing the attitude is to admit a unit quaternion representation $\mathbf{q} \in \text{SU}(2)$ (Kendall and Cipolla, 2017; Kendall, Grimes, et al., 2015; Proença and Gao, 2019). However, the quaternion representation is not necessarily continuous due to antipodal ambiguity, and the network is not capable of intrinsically learning the unit norm constraint, instead relying on brute-force normalisation to bring the output back to $\text{SU}(2)$. Both issues can prevent the network from properly fitting Equation (6.12). Kechagias-Stamatis et al. (2020) use a rotation matrix representation which does not suffer from this issue, but imposes instead an orthogonality constraint.

Instead, ChiNet learns a 6D continuous representation of the attitude proposed by Zhou et al. (2020) which has been shown to be more suitable for training, by making use of an orthogonalisation process in the representation itself to ensure that the network's output remains in $\text{SO}(3)$. The mapping $f_{6\text{D}}$ from the 6D representation, \mathbf{r} , to $\text{SO}(3)$ is given by:

$$f_{6\text{D}}: \mathbb{R}^{3 \times 2} \rightarrow \text{SO}(3)$$

$$\begin{bmatrix} | & | \\ \mathbf{r}_{1:3} & \mathbf{r}_{4:6} \\ | & | \end{bmatrix} \mapsto \begin{bmatrix} | & | & | \\ \mathbf{R}_{:,1} & \mathbf{R}_{:,2} & \mathbf{R}_{:,3} \\ | & | & | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \langle \mathbf{r}_{1:3} \rangle & \langle \mathbf{r}_{4:6} - (\mathbf{R}_{:,1}^\top \mathbf{r}_{4:6}) \mathbf{R}_{:,1} \rangle & \mathbf{R}_{:,1} \times \mathbf{R}_{:,2} \\ | & | & | \end{bmatrix}, \quad (6.14)$$

where $\langle \bullet \rangle$ denotes vector normalisation. The inverse mapping simply entails removing the last column of the rotation matrix \mathbf{R} :

$$f_{6\text{D}}^{-1}: \text{SO}(3) \rightarrow \mathbb{R}^{3 \times 2}$$

$$\mathbf{R} \mapsto \begin{bmatrix} | & | \\ \mathbf{R}_{:,1} & \mathbf{R}_{:,2} \\ | & | \end{bmatrix}. \quad (6.15)$$

The attitude is thus learned by minimising a regression loss based on the predicted

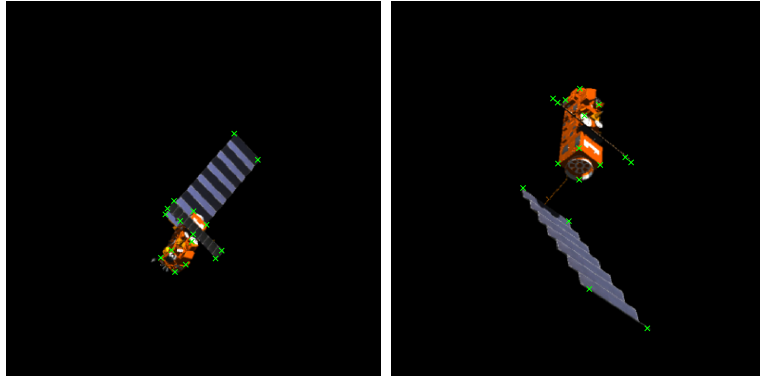


Figure 6.5: Model points \mathbb{P} of the ASTOS dataset for use with ChiNet’s refinement stage, reprojected onto two test images using the ground truth pose (green). $N = 16$ points corresponding to structural corners were manually selected.

values of \mathbf{r} :

$$\mathcal{L}_{\mathbf{r}}^{(S2)} = \sum_{\kappa=1}^T \|\hat{\mathbf{r}}^{(\kappa)} - \mathbf{r}^{(\kappa)}\|, \quad (6.16)$$

where the temporal component has been highlighted in terms of the training sequence length T . Similarly, for the position:

$$\mathcal{L}_{\mathbf{t}}^{(S2)} = \sum_{\kappa=1}^T \|\hat{\mathbf{t}}^{(\kappa)} - \mathbf{t}^{(\kappa)}\|, \quad (6.17)$$

The combined Stage 2 loss follows a similar rationale to Equation (6.11) in terms of Kendall, Gal, et al.’s (2018) formulation for two Gaussian distributions:

$$\mathcal{L}^{(S2)} = \mathcal{L}_{\mathbf{t}}^{(S2)} \sigma_{\mathbf{t}}^{-2} + \mathcal{L}_{\mathbf{r}}^{(S2)} \sigma_{\mathbf{r}}^{-2} + 2(\log \sigma_{\mathbf{t}} \sigma_{\mathbf{r}}). \quad (6.18)$$

Stage 2 (and 3) trains the complete pipeline using backpropagation through time (BPTT; Chap. 2). However, training very long sequences involves high memory requirements, so a truncated BPTT procedure is adopted instead. This entails unfolding the sequence for a predefined number of time-steps T smaller than the full sequence length, performing one training iteration, and then moving on to the next partition. In order to keep continuity while still allowing the network to learn long sequences, ChiNet follows Clark et al.’s (2017) approach whereby the training is carried out with a sliding window over the sequence, where consistency is established by appropriately initialising the LSTMs’s hidden states with those computed in the previous iteration.

6.3.2.3 Stage 3

The final training stage consists in a geometric refinement of the output from Stage 2, following the reprojection of 3D model points using the ground truth and predicted relative pose first proposed by Kendall and Cipolla (2017) for camera pose estimation in urban scenarios:

$$\boldsymbol{\theta}^{(\text{S3})*} = \arg \max_{\boldsymbol{\theta}^{(\text{S3})}} p \left(u^{(1)}, \dots, u^{(\kappa)} \mid \mathbf{I}^{(1)}, \dots, \mathbf{I}^{(\kappa)}, \mathbb{P}; \boldsymbol{\theta}^{(\text{S3})} \right), \quad (6.19)$$

$$\boldsymbol{\theta}^{(\text{S3})(0)} = \boldsymbol{\theta}^{(\text{S2})}, \quad (6.20)$$

where $\mathbb{P} = \{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(N)}\}$ is a manually selected set of N target model points expressed in \mathcal{F}_t . For the ASTOS dataset, $N = 16$ points were selected that corresponded to corners on the model of the target. Figure 6.5 illustrates the reprojection of the set \mathbb{P} onto the image plane. The loss is straightforwardly defined as:

$$\mathcal{L}^{(\text{S3})} = \sum_{\kappa=1}^T \sum_{i=1}^N \left\| \mathbf{z}^{(i,\kappa)} - \pi \left(\mathbf{K} \left(u^{(\kappa)} \oplus \mathbf{p}^{(i)} \right) \right) \right\|, \quad (6.21)$$

where $\mathbb{Z}_{\kappa} = \{\mathbf{z}^{(1,\kappa)}, \dots, \mathbf{z}^{(N,\kappa)}\}$ is the set of projected keypoints corresponding to \mathbb{P} at time $\tau = \tau_{\kappa}$, \mathbf{K} is the dataset camera intrinsic matrix, \oplus denotes pose-point composition, and $\pi(\bullet)$ is the projection operator as seen in Equation (2.7) (Chap. 2, § 2.7). Similarly to Stage 2, the 6D representation of the attitude is used. Equation (6.21) thus learns the pose implicitly via the minimisation of the reprojection error, which naturally balances the contributions of the position and attitude branches, and does not require defining explicit weights unlike Stages 1 and 2. This is advantageous for datasets in which the position depth has a high variance, since each contribution is weighed differently due to parallax, as reported by Kendall and Cipolla (2017). On the other hand, the loss formulation requires a good initialisation of the parameters $\boldsymbol{\theta}^{(\text{S3})}$ to converge, hence why it is used as a refinement stage.

6.3.3 Data Augmentation

Data augmentation is a form of pre-processing whereby the inputs are randomly enhanced to increase the training set variance and consequently the generalisation error of a DNN model. For CNNs specifically, this enhancement is applied directly to the images. Most data augmentation techniques focus on IP-based transforms, modifying the pixel values but leaving the labels intact. However, it is also possible to augment an image such that the labels are altered as well. Either way, both

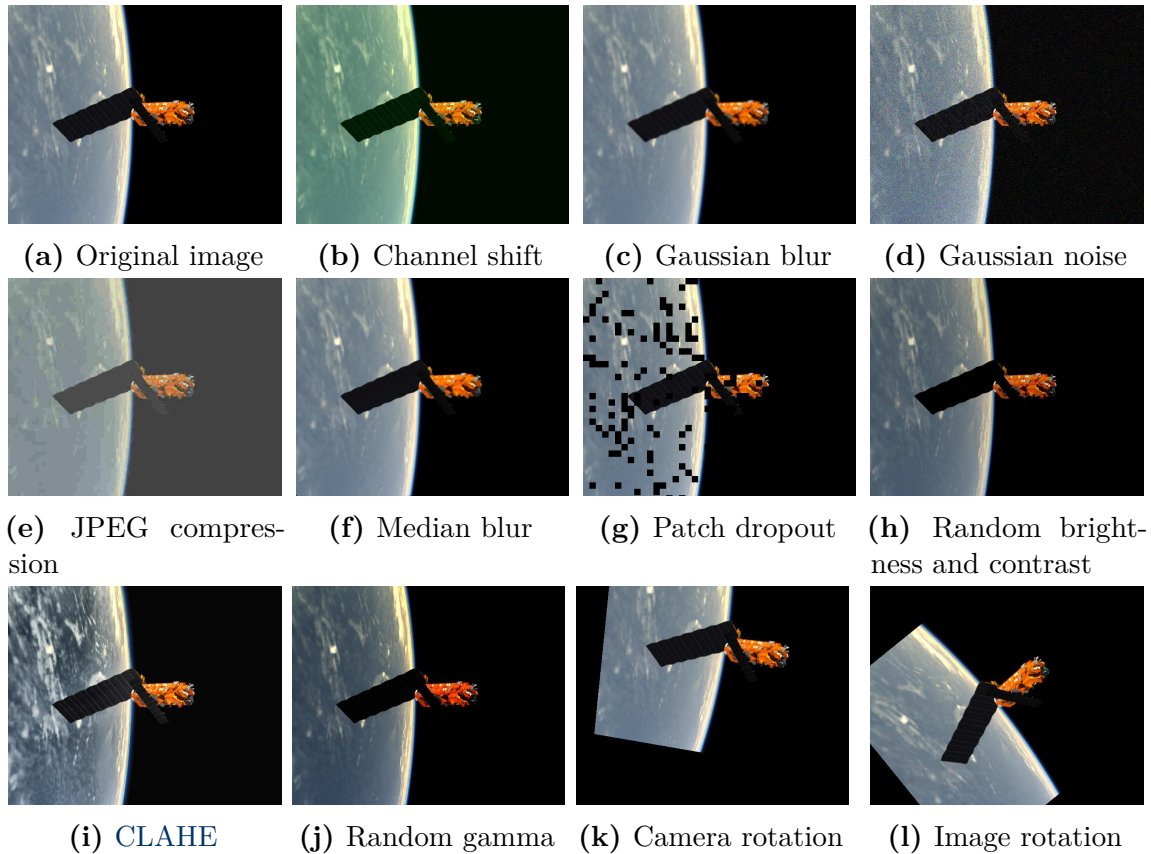


Figure 6.6: Image augmentation transform operations in use with ChiNet, exemplified on a red-green-blue (RGB) frame of the ASTOS dataset (a). A total of 11 transforms are employed: (b–j) image processing-based augmentations; (k–l) pose-based augmentations.

approaches result in the creation of “fake data” that increases the pool of available inputs for training.

Space rendezvous sequences, in particular, can benefit from image augmentation due to the slow relative motion between chaser and target and the periodicity intrinsic to target tumbling modes, which results in many frames looking comparatively similar and hence lower data variance. ChiNet makes use of 11 image augmentation transform operations; these are illustrated in Figure 6.6. Of these transforms, 9 are IP-based ones, which corrupt the aspect of the target to lead the network to focus on learning invariant features. This is particularly important to generalise for test sequences which have been imaged under different illumination conditions, or for real data when testing on synthetic sets, for example. Additionally, 2 pose-based augmentations are employed. In this case, the image is warped according to a homography \mathbf{H} induced by a pure rotation of the camera embodied in \mathbf{R} (Hartley and Zisserman, 2004):

$$\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}. \quad (6.22)$$

Table 6.1: Summary of experiments in Chapter 6.

Section	Description	Dataset
Section 6.4.4	Comparison of each stage's contribution in the multistage optimisation framework for the CNN with RGB inputs.	ASTOS
Section 6.4.6	Comparison of the CNN performance on RGB and RGBT inputs.	ASTOS
Section 6.4.5	Comparison of the performance between the vanilla CNN and the full DRCNN on RGB inputs.	ASTOS
Section 6.4.7	Summary of ChiNet's performance on the complete set of test data.	ASTOS
Section 6.4.8	Evaluation of ChiNet's performance on real data.	CITY

These introduce the network to random perturbations in the labels, injecting poses which might not usually be seen in a smooth, periodic rendezvous sequence.

Data augmentation is performed online for ChiNet, meaning that images are modified on-the-go and not prior to training, therefore not altering memory requirements for the training set. Multiple transforms can be applied to the same image and are controlled by a predefined probability of occurrence. These probabilities are tuned according to the current phase in the multistage optimisation pipeline: Stage 1 utilises oversampling to balance the attitude classes, hence augmentation is more frequent; in contrast, Stage 3 is a refinement stage, and therefore the image enhancement is lessened. Furthermore, transforms are applied consistency in between frames of the same training sequence.

6.4 Experiments

Experiments were conducted on both synthetic and laboratory datasets to validate each module of the proposed pipeline. Table 6.1 summarises the experiments conducted in this chapter.

6.4.1 Datasets

The datasets used to benchmark the performance of ChiNet are described below.

Astos Dataset The ASTOS dataset (Chap. 2) consists of 28 different rendezvous trajectories with the failed satellite Envisat, featuring three distinct guidance profiles, three tumbling modes, and two approach vectors, divided into sunlit and eclipsed sequences, imaging the target with both a visible camera and a thermal camera at a frequency of 10 Hz, thus making it ideal to benchmark the different contributions

that make up the ChiNet pipeline. The visible and LWIR images are aligned and resized to a resolution of $640 \text{ px} \times 512 \text{ px}$ for both training and testing.

City Dataset The CITY dataset consists of a collection of four rendezvous trajectories with a 1:4 scale mock-up of the National Aeronautics and Space Administration (NASA; United States) and National Centre for Space Studies (CNES; France) satellite Jason-1, acquired at City, University of London’s Autonomous Systems Laboratory. The mock-up rotates along its vertical axis at a constant rate of 6 deg s^{-1} . Despite having a different form factor, Jason-1 contains is similar to Envisat in terms of components (i.e. main bus coated in MLI, thermal radiators, solar array, radiometric instruments). In total, four trajectory types are considered:

- (1) CITY/FAR: The chaser observes the target at a fixed distance of 3.8 m. The target performs two revolutions during this period. The sequence lasts 2 min.
- (2) CITY/NEAR: The chaser observes the target at a fixed distance of 1.1 m. The target performs two revolutions during this period. The sequence lasts 2 min.
- (3) CITY/APPROACH-FAST: The chaser performs a translation along the line connection both centres of mass from an initial distance of 3.8 m to a final distance of 1.1 m, at a constant velocity of 9 cm s^{-1} . The target performs half a revolution during this period. The sequence lasts 30 s.
- (4) CITY/APPROACH-SLOW: The chaser performs a translation along the line connection both centres of mass from an initial distance of 3.8 m to a final distance of 2 m, at a constant velocity of 6 cm s^{-1} . The target performs half a revolution during this period. The sequence lasts 30 s.

Trajectories are acquired for simulation of both sunlight and eclipse conditions. On the visible spectrum, this is controlled respectively by aiming a floodlight directly at the target, or by aiming it at a nearby wall, creating a dimly lit environment. On the LWIR spectrum, the model’s temperature is controlled by internal resistor heaters in the main bus and by an external heater. The thermal signature of the model is made to coarsely match that of Envisat in both illumination conditions. Images are acquired at a resolution of $744 \text{ px} \times 490 \text{ px}$ and frequency of 10 Hz (software synchronised); the visible and thermal cameras are aligned and set up in a stereo configuration with a very short baseline to minimise disparity. The ground truth is recorded with an Optitrack motion capture system. Using the ground truth and the computer-aided design (CAD) model of the target, the background is digitally

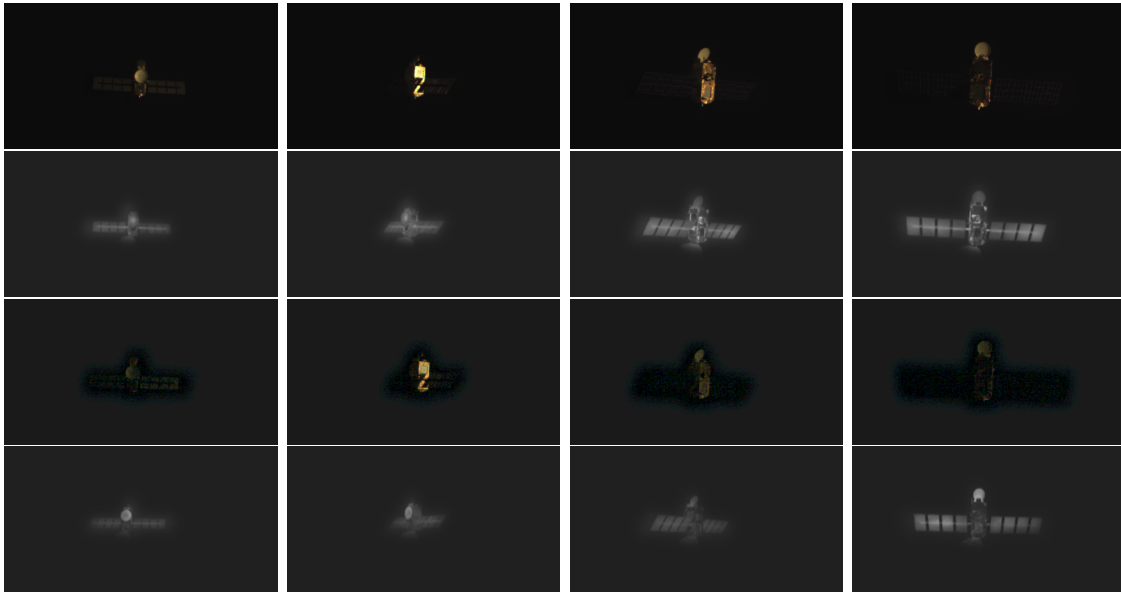


Figure 6.7: Sample images from the CITY/APPROACH-SLOW sequence imaged from the same pose under different modalities and illumination conditions. The images have been cropped for visualisation purposes. (*Top row*) Sunlight, visible modality. (*Second row*) Sunlight, long-wavelength infrared (LWIR) modality. (*Third row*) Eclipse, visible modality. The images have been exaggeratedly enhanced for better visualisation. (*Bottom row*) Eclipse, LWIR modality.

masked out to simulate a deep space background. Figure 6.7 depicts some sample frames of the dataset.

6.4.2 Training

In order to stress the pipeline, all models are trained exclusively on sunlit rendezvous sequences. Table 6.2 depicts the train/test data split for ASTOS, where seven sequences are used for training. This is equivalent to approximately 13 200 images. For the CITY dataset, CITY/FAR, CITY/NEAR, CITY/APPROACH-FAST are used for training, equivalent to 2760 images; CITY/APPROACH-SLOW is used for testing.

The performance of the training process is monitored by further extracting a validation dataset from the training sequences according to a 80–20 % partition. To accomplish this, all sequences of the training dataset are split into smaller sequences, where the length is randomly sampled from a range of powers of two. In the case of ASTOS, the possible subsequence lengths are $\{64, 128, 256, 512\}$; for CITY, these are $\{32, 64, 128, 256\}$, as the original sequences are shorter.

Clark et al.’s (2017) method is used to train the RNN module whereby each sequence is fed to the network according to a sliding window. In the present experiments, a window length of 8 frames with a stride of 4 was utilised.

Table 6.2: Train/test data split on the ASTOS dataset. All train sequences depict sunlit sequences, whereas the testing campaign includes both sunlit and eclipsed periods. Consult Chapter 2, Section 2.5.2, Figures (2.21) to (2.23) for the nomenclature.

Sequence		Train	Test
G1/	R1/ VBAR/	×	
	R1/ RBAR/		×
	R2/ VBAR/		×
	R2/ RBAR/	×	
	R3/ VBAR/	×	
	R3/ RBAR/		×
G2/	R1/ VBAR/		×
	R1/ RBAR/	×	
	R2/ VBAR/	×	
	R2/ RBAR/		×
	R3/ VBAR/		×
	R3/ RBAR/	×	
G3/	R1/	×	
	R2/		×

Table 6.3: Base learning rates used in the training of the complete multimodal deep recurrent convolutional neural network (DRCNN) pipeline.

Dataset	Stage 1	Stage 2		Stage 3	
	CNN	CNN	RNN	CNN	RNN
ASTOS	2.0×10^{-2}	3.0×10^{-4}	8.0×10^{-5}	4.1×10^{-4}	5.0×10^{-6}
CITY	2.0×10^{-2}	2.0×10^{-4}	8.0×10^{-5}	1.0×10^{-4}	1.0×10^{-4}

Image augmentation is performed online on the training data as depicted in Section 6.3.3 and according to a probability value. The probability is set to its highest during Stage 1 and successively lowered until Stage 3. When training the RNN module, the augmentations are applied consistently for each sequence.

Stages 1 and 2 are trained for 100 epochs with a cyclical learning rate decay of 5 cycles (Smith, 2017), whereas Stage 3 is trained for 66 epochs with early stopping and a step learning rate decay every 9 epochs. Stage 1 samples the dataset for a total of 10 000 images. The CNN and RNN modules are trained separately, but sequentially. The Adam optimiser (Kingma and J. Ba, 2014) is used. The final pipeline uses a dropout probability of 0.2, and hidden and cell states zoneout factors of 0.15 for both. The learning rates used in the training of the final pipeline are summarised in Table 6.3.

The DRCNN is implemented from the ground up on MATLAB version R2019b.

The pipeline is trained on Cranfield's high performance computing facility Delta using one NVIDIA® Turing® V100 Tensor Core graphics processing unit (GPU).

6.4.3 Testing

The test results are presented in terms of the position and attitude error metrics, respectively:

$$\delta\tilde{t} := \|\hat{\mathbf{t}} - \mathbf{t}\|, \quad (6.23)$$

$$\delta\tilde{q} := 2 \arccos \left(\hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \right)_4. \quad (6.24)$$

Additionally, the position error is also assessed in terms of the relative range:

$$\delta\tilde{t}_r := \frac{\delta\tilde{t}}{\|\mathbf{t}\|}. \quad (6.25)$$

6.4.4 Evaluation of Multistage Optimisation

To assess the contribution of each stage in the proposed multistage optimisation scheme, the CNN module is trained according to four different schemes: 1) Stage 2 only for 100 epochs [S2-100]; 2) Stage 2 only for 200 epochs [S2-200]; 3) Stages 1 and 2 [S1,S2]; and 4) Stages 1, 2, and 3 [S1,S2,S3]. The comparison tests are performed for two sample test sequences of the ASTOS dataset: G2/R1/VBAR and G3/R2. The former represents the baseline case, where the only relative motion is the tumbling of the target, whereas the latter adds complexity not only in terms of the additional rotation mode but due to the elliptical relative translation and the manifestation of Earth in the background.

Figure 6.8a depicts the results of the benchmark on ASTOS/G2/R1/VBAR. From the overall shape of the plot lines, the periodicity of the tumbling motion can be clearly discerned. An initial period approximately covering the interval $\tau \in [0 ; 60]$ s is first noted, during which the target performs slightly over half a revolution and the errors are overall higher, culminating in a local peak at which the solar array reflects Earth's rim. It is then followed by a second period covering $\tau = [60 ; 103]$ s where the main bus comes back into view and both shadows and reflections are minimised, hence driving down the errors. This pattern is repeated twice more throughout the plot as the target performs a total of three revolutions.

Regarding the position error, the S1,S2 strategy is essentially on par with S2-100 and S2-200 for the first period, and performs better than both on the second period. Notably, the benefit of the dual-stage training can be observed specifically at times

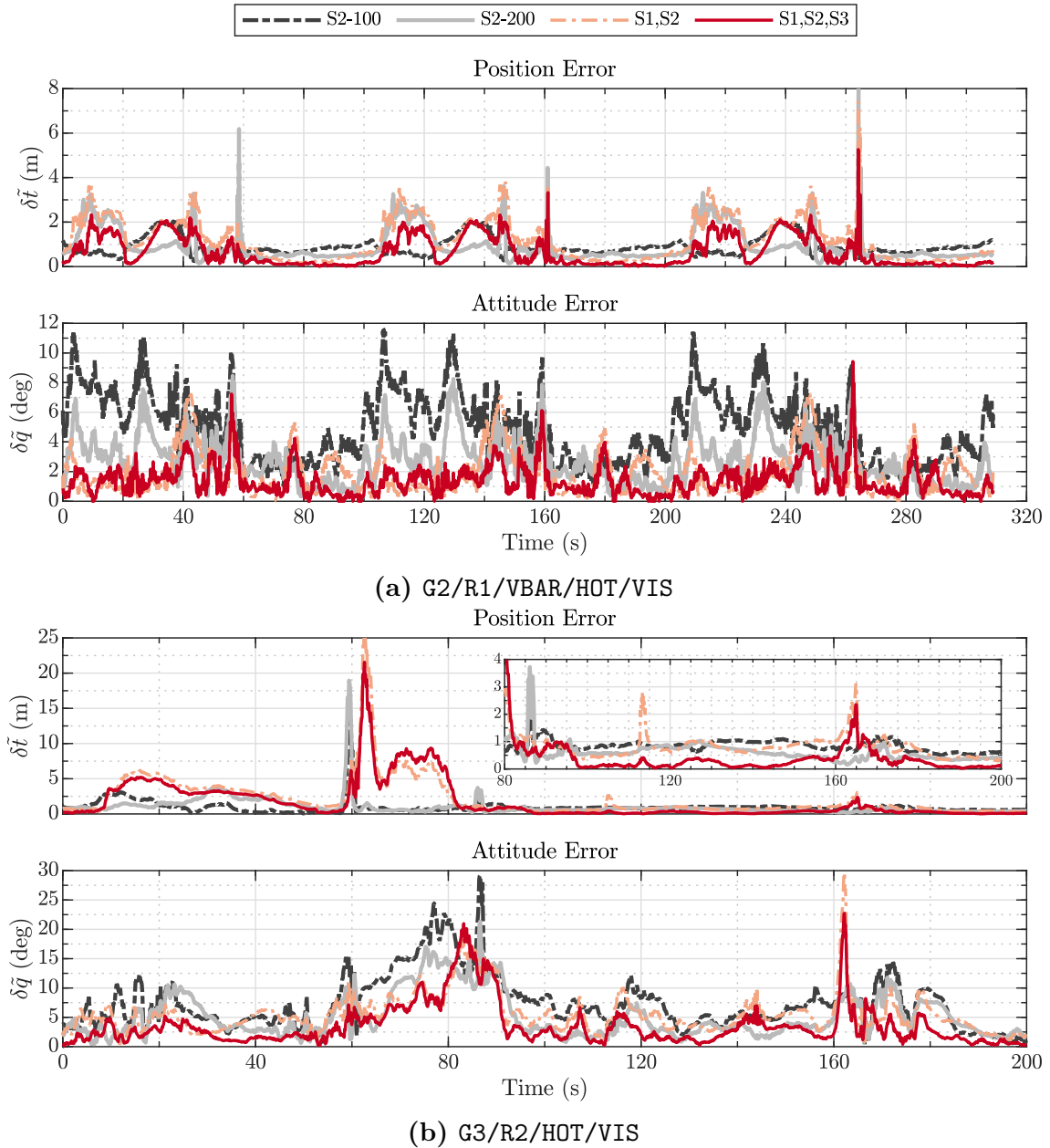


Figure 6.8: Comparison of estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences in terms of training stages used. All models are trained on a convolutional neural network (CNN) taking red-green-blue (RGB) inputs. (*S2-100*) Stage 2 trained for 100 epochs. (*S2-200*) Stage 2 trained for 200 epochs. (*S1,S2*) Stage 1 and Stage 2. (*S1,S2,S3*) Stage 1, Stage 2, and Stage 3.

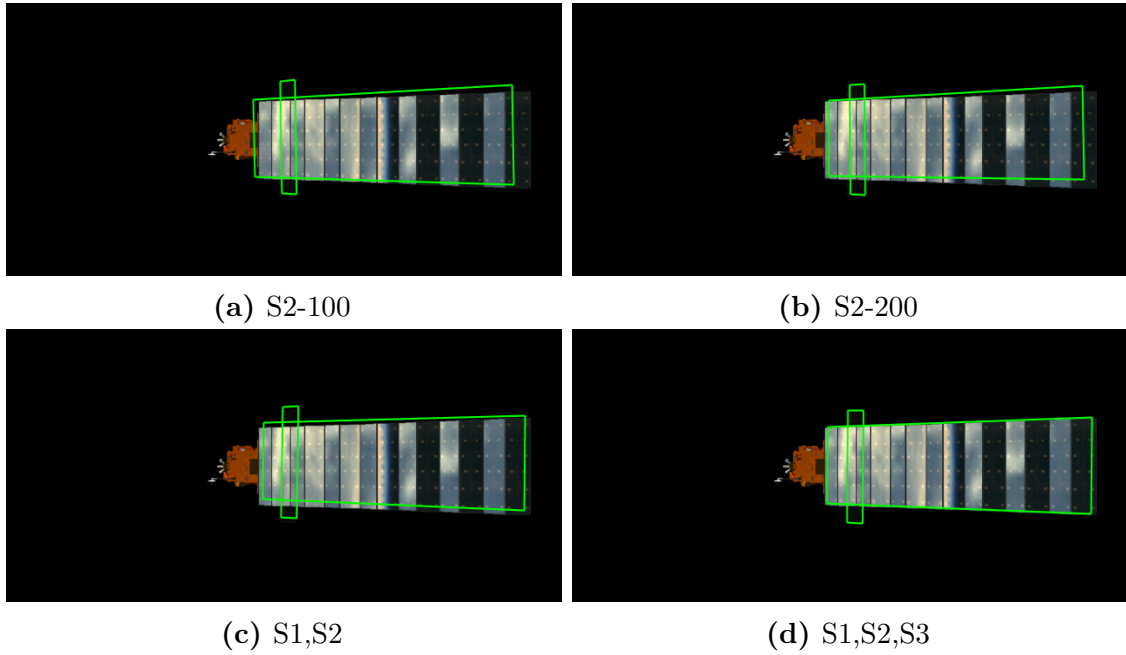


Figure 6.9: Effect of multistage optimisation. Illustrated on the ASTOS/G2/R1/VBAR/HOT/VIS sequence at time $\tau = 59.4$ s. Each stage progressively improves the pose estimate in the presence of spurious reflections, as shown by the model reprojection in green. (*S2-200*) Stage 2 trained for 200 epochs. (*S1,S2*) Stage 1 and Stage 2. (*S1,S2,S3*) Stage 1, Stage 2, and Stage 3.

$\tau = \{60, 160, 260\}$ s, where a mitigation of the error spikes is seen. Training on the three stages (S1,S2,S3) reduces these peaks even further. This is shown in Figure 6.9

The gains of adopting the proposed method become clearer looking at the attitude error plot. S2-100 exhibits the higher error throughout, followed by S2-200. The dual-stage S1,S2 approach further reduces the error, except for peaks at $\{45, 147, 250\}$ s, where it is comparable to the previous mode; this corresponds to the segments where the target nearly completes half a revolution and the solar array begins to cover the main bus. The triple-stage approach can be seen to provide the steadiest performance. It is also noted that the highest error peaks for the attitude correspond to those identified for the position, which S1,S2,S3 mitigates, but does not completely eliminate.

Table 6.4 summarises the errors numerically for each approach, validating the analysis done above. S1,S2 improves the attitude error on average relative to training directly through S2 at the cost of a slight increase in the mean and median position errors. S1,S2,S3 provides the best result overall, providing a mean sub-metre position accuracy and mean attitude error below 1.5 deg. The former is equivalent to a position error of 1.34% of range.

Figure 6.8b performs the same comparison, this time on the ASTOS/G3/R2 se-

Table 6.4: Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of training stages used. All models are trained on a convolutional neural network (CNN) taking red-green-blue (RGB) inputs. (*S2-100*) Stage 2 trained for 100 epochs. (*S2-200*) Stage 2 trained for 200 epochs. (*S1,S2*) Stage 1 and Stage 2. (*S1,S2,S3*) Stage 1, Stage 2, and Stage 3. The best results are highlighted in bold.

Model	$\delta\tilde{t}$ (m)		$\delta\tilde{t}_r$ (-)		$\delta\tilde{q}$ (deg)	
	Mean	Median	Mean	Median	Mean	Median
G2/R1/VBAR/HOT/VIS						
S2-100	0.90	0.78	0.0180	0.0156	4.90	4.61
S2-200	0.93	0.64	0.0186	0.0128	2.77	2.57
S1,S2	1.23	0.95	0.0245	0.0189	1.95	1.43
S1,S2,S3	0.67	0.30	0.0134	0.0059	1.40	1.17
G3/R2/HOT/VIS						
S2-100	0.95	0.81	0.0190	0.0162	7.34	6.40
S2-200	0.98	0.67	0.0196	0.0135	5.13	3.65
S1,S2	2.18	0.98	0.0436	0.0197	5.56	4.89
S1,S2,S3	1.75	0.41	0.0349	0.0082	3.55	2.45

quence. It can be observed that the single-stage training schemes actually perform better for the initial period of $[0 ; 80]$ s in terms of position error. An initial peak can be observed around 60s, which corresponds to the period in the trajectory where the solar array is angled such that the light reflected from the sun witnesses its maximal intensity. The dual- and triple-stage strategies are able to lessen the impact of this, but fail to avoid a spike shortly after, which corresponds to the period where the sunlight is still directly hitting both the solar array and the MLI facing the camera. During this period, most of Envisat’s main bus is covered by the solar array, which could explain the fact that S1,S2,S3 performs slightly worse than S1,S2 (i.e. there are fewer corner points visible, cf. Fig. 6.5).

The two approaches recover at 80s, after which the solar array ceases to occlude the spacecraft. Beyond this point, S1,S2,S3 becomes the best-performing strategy, save for a peak around 165s, where the panel once again dominates the FOV. Immediately after, the error is reduced but both S2-only strategies worsen, corresponding to a point where the array once again reflects Earth’s rim. The results for this trajectory therefore suggest that the proposed multistage optimisation robustifies the estimate of the position against aggressive illumination-induced artefacts but suffer when the target is self-occluded by the solar array. At this point, Earth is no longer present on the FOV, which could explain the fact that the spike is less intense

Table 6.5: Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of recurrence, benchmarking the plain convolutional neural network (CNN) against the complete deep recurrent convolutional neural network (DRCNN). All models are trained on Stages 1 and 2 and RGB inputs. The best results are highlighted in bold.

Model	$\delta\tilde{t}$ (m)		$\delta\tilde{t}_r$ (-)		$\delta\tilde{q}$ (deg)	
	Mean	Median	Mean	Median	Mean	Median
G2/R1/VBAR/HOT/VIS						
CNN	1.23	0.95	0.0245	0.0189	1.95	1.43
DRCNN	0.70	0.58	0.0140	0.0117	3.03	2.47
G3/R2/HOT/VIS						
CNN	2.18	0.98	0.0436	0.0197	5.56	4.89
DRCNN	1.00	0.78	0.0199	0.0156	6.09	5.72

compared to the one at 60 s.

The attitude estimation performance for ASTOS/G3/R2 largely follows the same trend, where once more S1,S2,S3 performs better overall. The initial increase in attitude errors, though, appears to persist for ten additional seconds, after which it decreases, corresponding to the point in the trajectory where Earth is mostly no longer present in the FOV, indicating that the attitude estimate is more sensitive to this factor.

Interestingly, despite having a higher mean position error relative to Stage 2 only, the complete multistage approach provides the best performance in terms of the median value (0.41 m against 0.81 m for S2-100, see Tab. 6.4).

6.4.5 Evaluation of Recurrent Module

Spacecraft pose estimation using DNNs has been exclusively tackled as a classification or regression task operating on each image individually. To study the benefit of modelling the problem as one dealing with a sequence of time-correlated images, two separate models are trained: one consisting solely in the plain CNN, and another consisting in the complete DRCNN pipeline using LSTMs. Both models are trained on Stages 1 and 2, and on RGB inputs.

Figure 6.10 plots the estimation results over time for the two sample trajectories, and Table 6.5 summarises them in terms of mean and median values. The DRCNN is successful in overwhelmingly mitigating the localised position error peaks for both trajectories, which correspond to points in the trajectory where the solar array reflections are most intense or it occludes the main bus, as mentioned in the previous

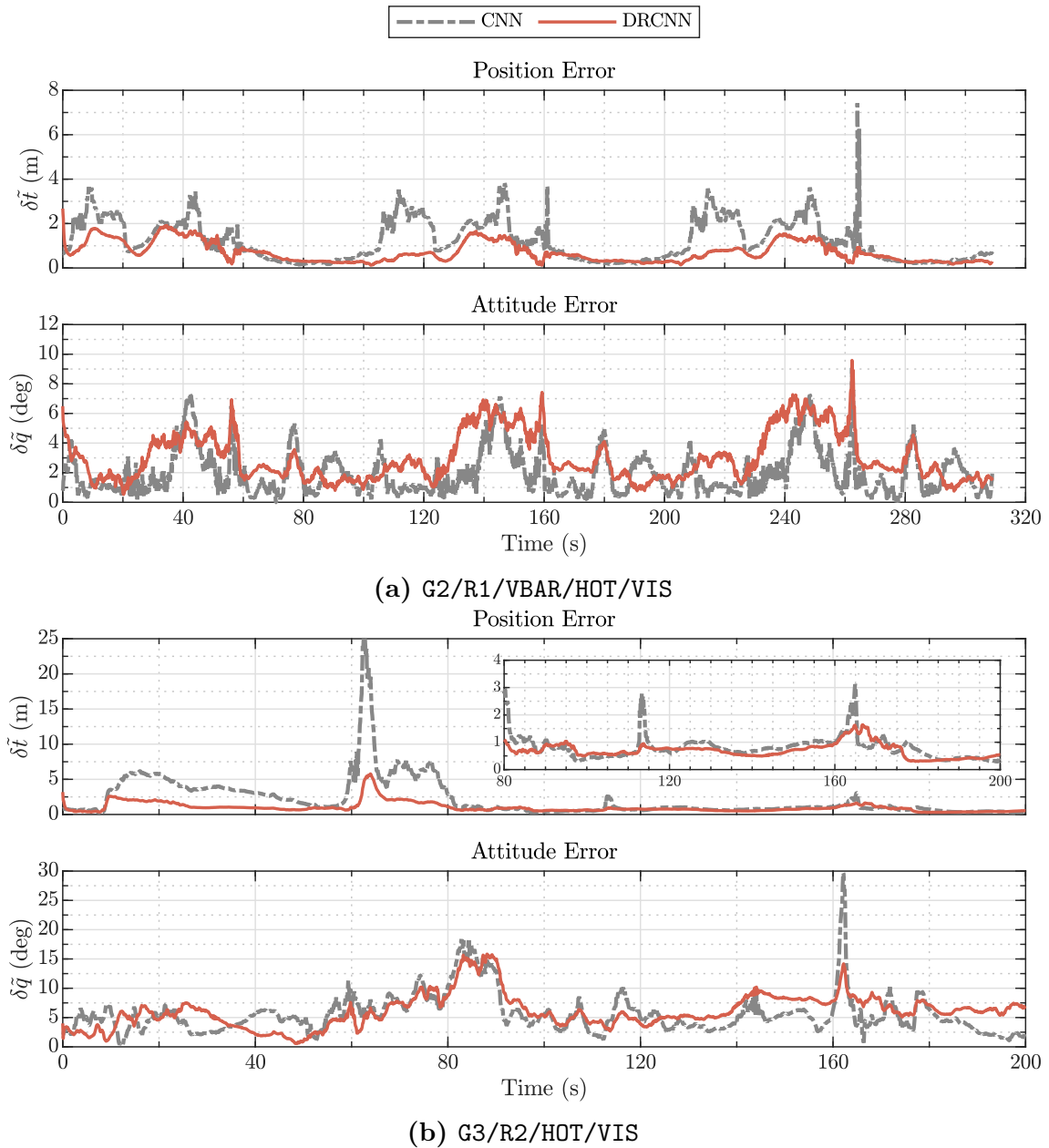


Figure 6.10: Comparison of estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences in terms of recurrence, benchmarking the plain convolutional neural network (CNN) against the complete deep recurrent convolutional neural network (DRCNN). All models are trained on Stages 1 and 2 and RGB inputs.

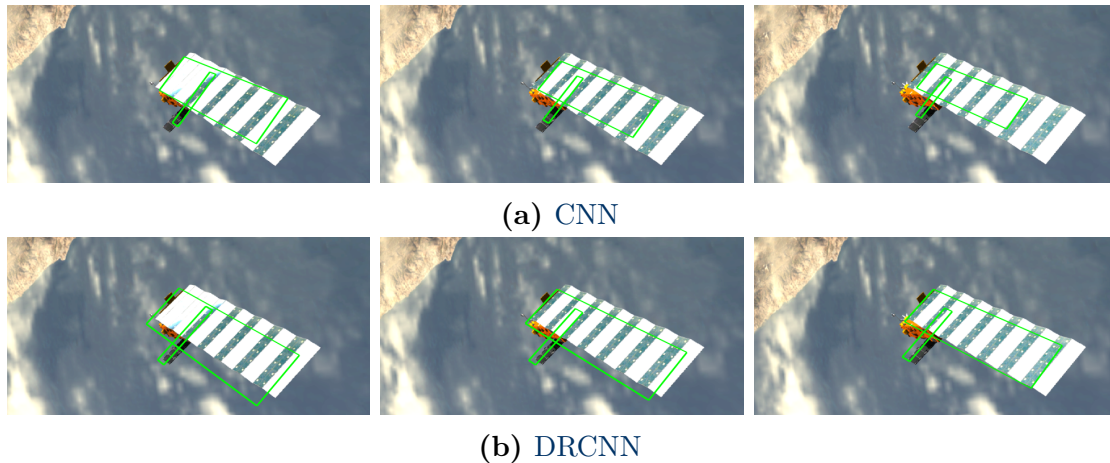


Figure 6.11: Effect of recurrent module illustrated on the ASTOS/G3/R2/HOT/VIS sequence at times $\tau = \{61, 62, 63\}$ s (left to right). The effect of the recurrent neural network (RNN) is mostly observed in the mitigation of position estimation errors, especially in terms of the range.

section. This is due to the LSTM states taking into account the preceding images, thus preventing sudden jumps in the solution. Figure 6.11 illustrates this for a triad of frames. The mean position error is reduced approximately by half, bringing the mean range-normalised error to approximately 1.40% and 2% for each trajectory, respectively.

The mean values for the attitude errors, however, are slightly worse for the RNN-based architecture. Overall, an increase of 0.5–1 deg in the mean error and 1 deg in the median error is observed. It can be argued that this is an acceptable loss in performance given the benefit seen for the position estimation. However, the pipeline can be easily modified to output an attitude estimate from the CNN alone while processing the position with the RNN; this is left as future work. Nonetheless, the perhaps more substantial trade-off to consider is whether peak mitigation (e.g. the one observed at 165 s on ASTOS/G3/R2/HOT) is preferred over average performance.

6.4.6 Evaluation of Multimodal Inputs

In this section, the influence of augmenting the RGB input produced by regular camera with an image in the LWIR, thus creating a four channel multimodal RGBT input, is evaluated. Two models are trained for comparison, one with inputs exclusively on the visible modality, and another with multimodal inputs. Both models are trained on Stages 1 and 2. The results are depicted in Figure 6.12 and Table 6.12.

The contribution of the multimodality can be seen immediately in Figure 6.12a, where the plots of both position and attitude errors in time for ASTOS/G2/R1/VBAR

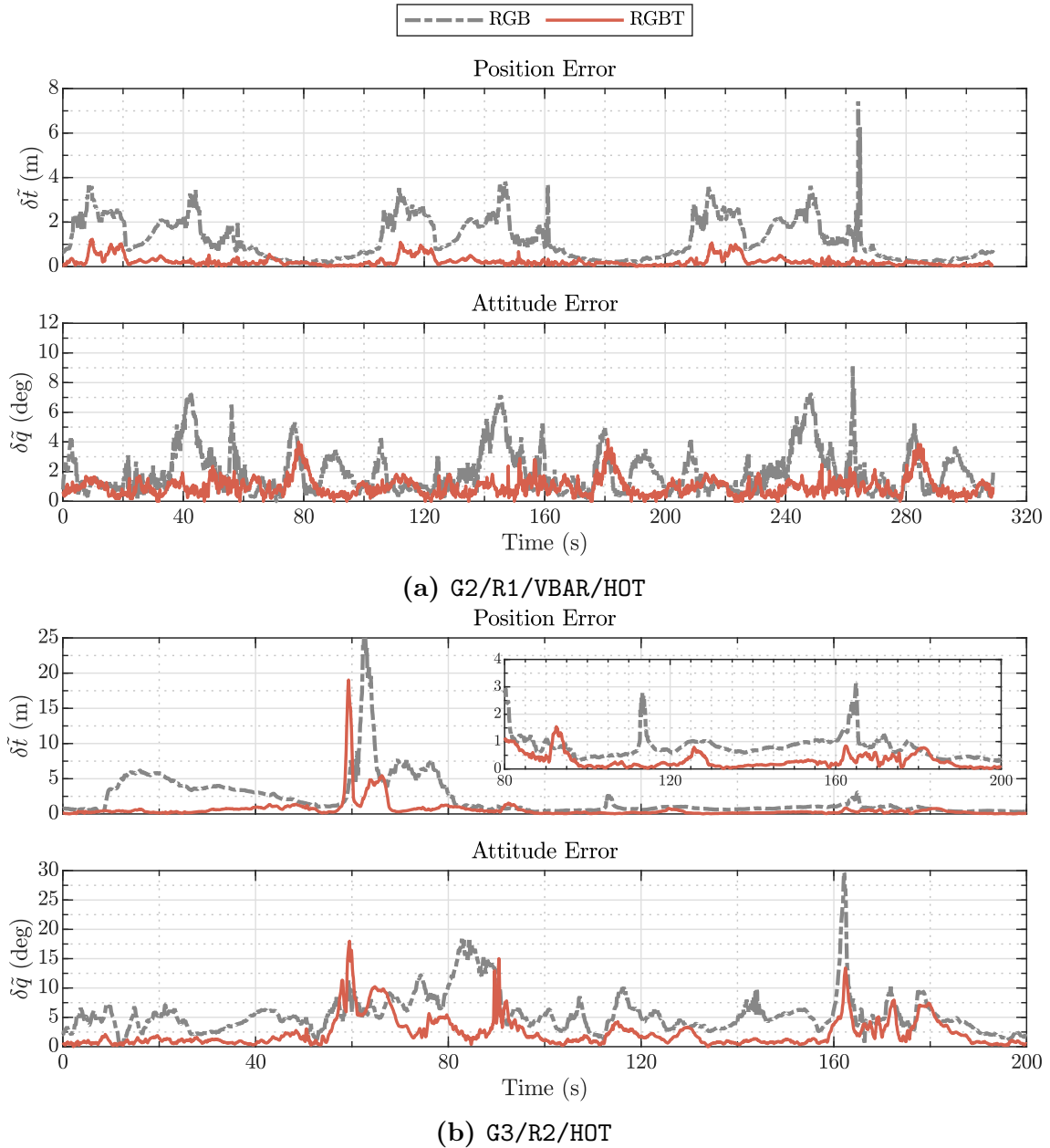


Figure 6.12: Comparison of estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences in terms of imaging modality, benchmarking red-green-blue (RGB) inputs against the multimodal red-green-blue-thermal (RGBT). All models are trained on a convolutional neural network (CNN) and Stages 1 and 2.

Table 6.6: Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of imaging modality, benchmarking red-green-blue (RGB) inputs against the multimodal red-green-blue-thermal (RGBT). All models are trained on a convolutional neural network (CNN) and Stages 1 and 2. The best results are highlighted in bold.

Model	$\delta\tilde{t}$ (m)		$\delta\tilde{t}_r$ (-)		$\delta\tilde{q}$ (deg)	
	Mean	Median	Mean	Median	Mean	Median
G2/R1/VBAR/HOT						
RGB	1.23	0.95	0.0245	0.0189	1.95	1.43
RGBT	0.23	0.17	0.0046	0.0034	0.98	0.88
G3/R2/HOT						
RGB	2.18	0.98	0.0436	0.0197	5.56	4.89
RGBT	0.61	0.31	0.0122	0.0063	2.37	1.40

exhibit more stability for RGBT inputs compared to RGB inputs. Notably, not only are the reflection-induced peaks mitigated, but the errors corresponding to the approximate first half of the tumbling period (refer again to § 6.4.4) are as well. Overall, the mean position error is reduced in almost 80% by using multimodal inputs, granting a mean range-normalised position error below 0.5%, compared to 2.5% for visible only. The mean attitude error is halved, becoming slightly lower than 1 deg.

Figure 6.12b plots the estimation errors over time for the more complex sequence ASTOS/G3/R2. The RGBT position error is lower than the baseline throughout the entire sequence, save for the peak centred at $\tau = 60$ s, which is not present in the baseline (cf. Fig. 6.8a). Because, at this point, the solar array is blocking the line-of-sight (LOS) to the bus, which appears essentially featureless hot body in the LWIR band, the network is in error. Nevertheless, immediately after, the error is brought down again; this is posited because as the main body becomes visible again the highly-contrasting cold radiators do too, generating features for the network, which are otherwise not intense enough in the RGB.

Despite the general degradation in performance for this trajectory, the position error metrics are better in all aspects for the multimodal approach, with a reduction in the mean error of more than 70%. The improvements in the attitude error are also notable (more than halved for the mean, reduced by more than two thirds for the median). Figure 6.13 qualitatively shows this effect.

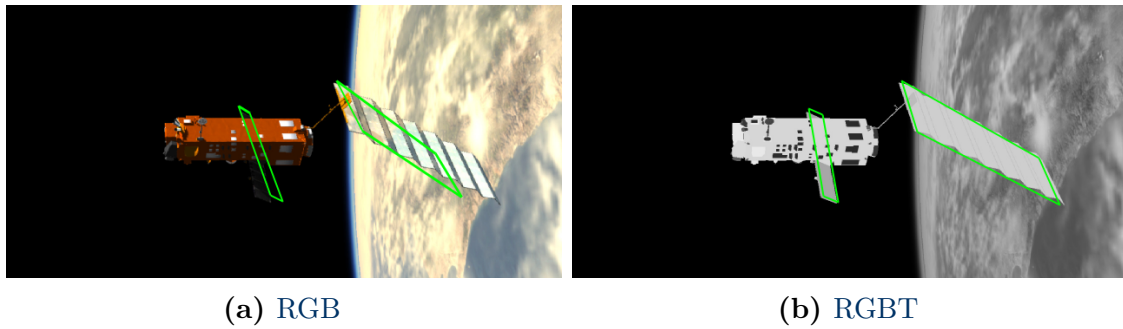


Figure 6.13: Effect of multimodality. Illustrated on the ASTOS/G3/R2/HOT trajectory at time $\tau = 88.5$ s. The RGBT-trained (red-green-blue-thermal) model (here represented as a thermal image) generally improves the pose with respect to the RGB-trained (red-green-blue) one.

6.4.6.1 Evaluation on Eclipse Sequences

The addition of the LWIR was shown above to greatly reduce the pose estimation errors under nominal conditions. Additionally, it would also be interesting to evaluate its performance under low illumination conditions: not being under direct sunlight means that a visible camera’s gain must be increased to boost the signal, which in turn also boosts noise, but despite lower temperatures in general the target is still emitting heat. As such, the benchmarks conducted in the beginning of Section 6.4.6 are now repeated for sample eclipse trajectories of the ASTOS dataset. This is an extremely complex test since the networks have only been trained on sunlit trajectories.

Despite the image augmentation module, it was found that testing the network directly on eclipse sequences directly did not produce a converging solution. As such, the images are adjusted manually in terms of gamma to make the darker areas lighter. The same adjustment is applied globally regardless of the sequence and the process involves very little supervision. Note that localised sections in the image are not being selectively manipulated, as the transformation is global. Furthermore, the thermal signature of the target in the LWIR is still different than the one seen during training. To compensate for a higher ISO setting on the visible camera associated to the lack of light, the pixels are corrupted with random values drawn from a zero-mean Gaussian distribution to emulate sensor noise. The noise is then also amplified by the gamma correction.

The tests are performed for two different noise levels: low-intensity noise (abbreviated “N-L”) with a standard deviation $\sigma = 4 \times 10^{-3}$ (equivalent to 1 on a pixel range of 0–255), and high-intensity noise (“N-H”) with a standard deviation $\sigma = 1 \times 10^{-2}$ (equivalent to 2.5).

Figure 6.14 illustrates the pose estimation errors over time for the ASTOS/G2/R1/-

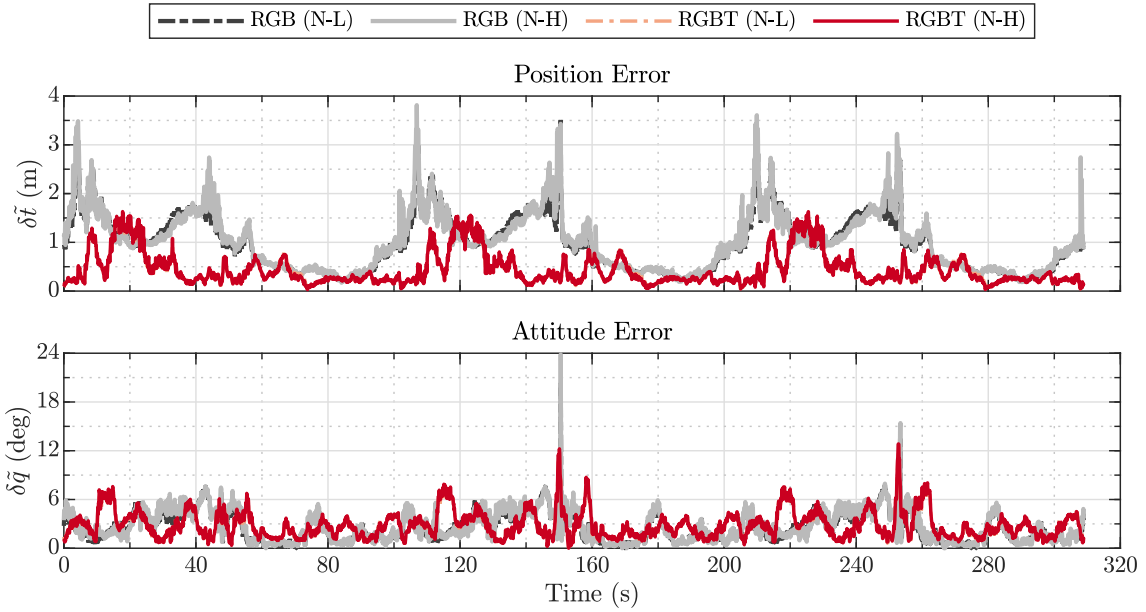


Figure 6.14: Comparison of estimated position and attitude errors over time on the G2/R1/VBAR/COLD rendezvous sequence from the ASTOS dataset in terms of imaging modality, benchmarking red-green-blue (RGB) inputs against the multimodal red-green-blue-thermal (RGBT). All models are trained on a convolutional neural network (CNN) and Stages 1 and 2. Note that the sequence depicts an eclipse period. (*N-L*) Noise, low intensity. (*N-H*) Noise, high intensity.

Table 6.7: Comparison of position and attitude error statistics on two sample ASTOS dataset rendezvous sequences in terms of imaging modality, benchmarking red-green-blue (RGB) inputs against the multimodal red-green-blue-thermal (RGBT). All models are trained on a convolutional neural network (CNN) and Stages 1 and 2. The best results are highlighted in bold. Note that all sequences depict an eclipse period. (*N-L*) Noise, low intensity. (*N-H*) Noise, high intensity. (*D*) Solution diverges.

Model	$\delta \tilde{t}$ (m)		$\delta \tilde{t}_r$ (-)		$\delta \tilde{q}$ (deg)	
	Mean	Median	Mean	Median	Mean	Median
G2/R1/VBAR/COLD (N-L)						
RGB	0.99	0.94	0.0198	0.0188	2.53	2.19
RGBT	0.44	0.31	0.0088	0.0063	2.79	2.48
G2/R1/VBAR/COLD (N-H)						
RGB	1.02	0.99	0.0205	0.0197	2.70	2.21
RGBT	0.45	0.30	0.0089	0.0060	2.88	2.56
G3/R2/COLD (N-L)						
RGB	1.42	1.06	0.0285	0.0212	20.56	16.55
RGBT	2.11	1.99	0.0423	0.0399	D	D

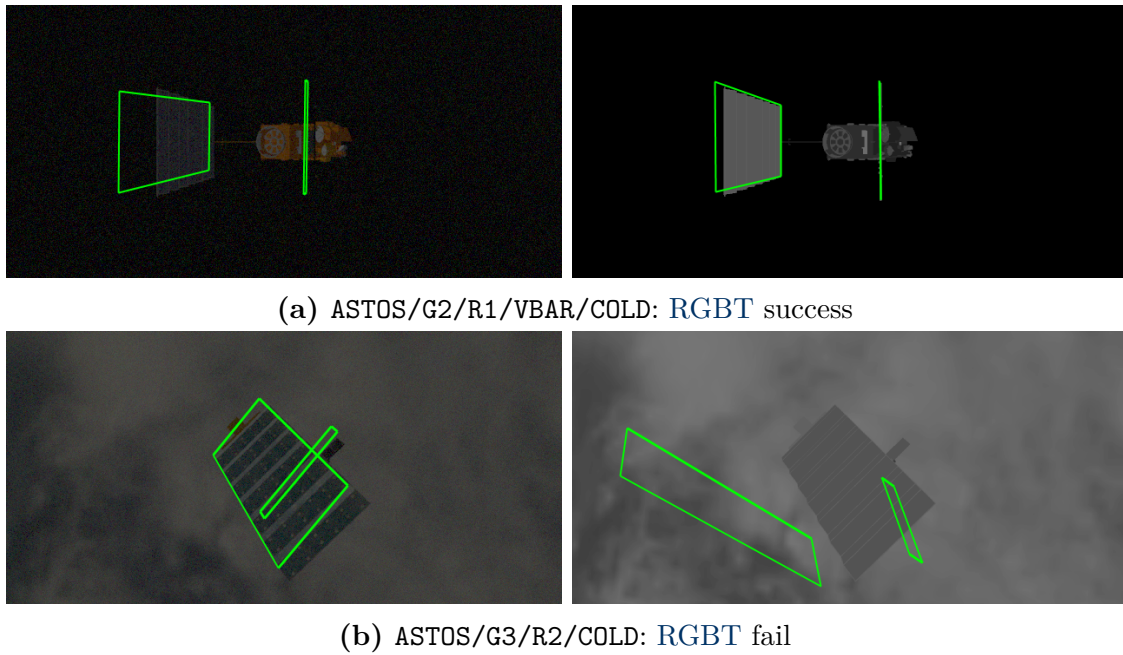


Figure 6.15: Effect of multimodality illustrated on eclipse sequences. (*Left Column*) red-green-blue (RGB) model. (*Right Column*) red-green-blue-thermal (RGBT) model.

VBAR/COLD sequence. Table 6.7 summarises the errors numerically. Figure 6.15a displays a qualitative version of these results. Both RGB and RGBT models are largely impervious to the sensor noise levels (a result of the image augmentation module), especially the latter. The visible band model actually produces a slightly improved mean position error since extreme lighting effects are mitigated; note the absence of the spike in position error at time $\tau = 165$ s. The attitude error, on the other hand, rises by 0.5–0.8 deg on average. The multimodal model provides a position accuracy with half the mean error relative to the visible, and twice as large as its hot case counterpart. Still, the range-normalised accuracy is kept below 1%. Surprisingly, the attitude estimate is not improved, and even marginally worsens.

The models were also tested on the ASTOS/G3/R2/COLD sequence. For the RGB model, whereas the position error is lower than its hot case counterpart in terms of the mean, at nearly 3% of the range (but worse in terms of median), the attitude estimation performance is over 3.5 worse, scoring an average error of approximately 20.5 deg. The RGBT model, while capable of providing an estimate of the position with a mean range-normalised error of around 4%, does not converge for the attitude. This represents a limitation of the thermal imaging when Earth is also present in the FOV. It is postulated that this is due to the lower intensity thermal signature of the target which is too similar to that of Earth’s. Since the network has only been trained on high-contrasting images for the LWIR band originating from the hot cases,

Table 6.8: Summary of position and attitude error statistics on all ASTOS dataset rendezvous test sequences for the complete deep recurrent convolutional neural network (DRCNN) pipeline, trained on Stages 1, 2, and 3. Only sunlit sequences have been used for training. All tests use multimodal red-green-blue-thermal (RGBT) inputs unless otherwise stated. (*N-L*) Noise, low intensity. (*N-H*) Noise, high intensity. (*D*) Solution diverges.

Sequence			$\delta \tilde{t}$ (m)		$\delta \tilde{t}_r$ (-)		$\delta \tilde{q}$ (deg)	
			Mean	Median	Mean	Median	Mean	Median
G1/	R1/	RBAR/ HOT	3.45	3.51	0.0468	0.0473	7.49	4.80
	R2/	VBAR/ HOT	4.05	4.23	0.0547	0.0581	8.67	4.53
	R3/	RBAR/ HOT	3.09	3.12	0.0437	0.0437	14.12	8.63
G2/	R1/	VBAR/ HOT	0.24	0.23	0.0049	0.0046	1.85	1.80
	R2/	RBAR/ HOT	0.33	0.24	0.0065	0.0048	2.09	1.26
	R3/	VBAR/ HOT	0.67	0.63	0.0134	0.0126	10.61	9.02
G3/	R2/	HOT	0.29	0.21	0.0058	0.0041	3.52	2.77
G1/	R1/	RBAR/ COLD (N-L)*	5.33	3.53	0.0676	0.0589	16.82	9.47
	R1/	RBAR/ COLD (N-H)*	5.85	4.72	0.0757	0.0729	17.14	7.84
	R2/	VBAR/ COLD (N-L)	4.00	3.80	0.0549	0.0534	23.26	10.98
	R2/	VBAR/ COLD (N-H)	3.95	3.72	0.0542	0.0519	23.66	10.80
	R3/	RBAR/ COLD (N-L)*	4.96	4.42	0.0693	0.0642	D	D
	R3/	RBAR/ COLD (N-H)*	5.16	4.43	0.0718	0.0637	D	D
G2/	R1/	VBAR/ COLD (N-L)	0.36	0.33	0.0071	0.0067	3.04	2.86
	R1/	VBAR/ COLD (N-H)	0.35	0.33	0.0070	0.0066	3.13	3.01
	R2/	RBAR/ COLD (N-L)*	0.95	0.83	0.0190	0.0167	22.38	20.93
	R2/	RBAR/ COLD (N-H)*	0.94	0.87	0.0188	0.0174	22.99	20.96
	R3/	VBAR/ COLD (N-L)	0.93	0.82	0.0186	0.0164	21.98	11.01
	R3/	VBAR/ COLD (N-H)	0.94	0.83	0.0188	0.0166	22.18	11.20
G3/	R2/	COLD (N-L)*	1.11	0.73	0.0221	0.0146	16.63	14.19
G3/	R2/	COLD (N-H)*	1.16	0.77	0.0232	0.0154	18.47	15.51

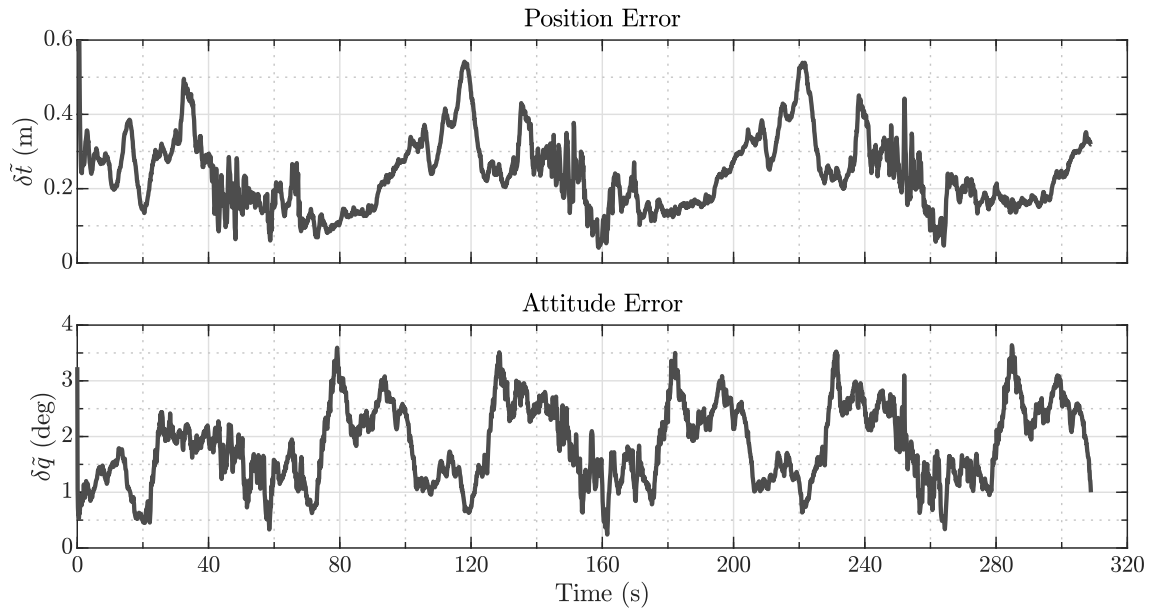
* Evaluated on red-green-blue (RGB) inputs.

it has difficulties identifying features on that band during eclipse. The comparative resilience on the visible band could in turn be explained by the fact that colour-based salient features can still be extracted. This is exemplified in Figure 6.15b.

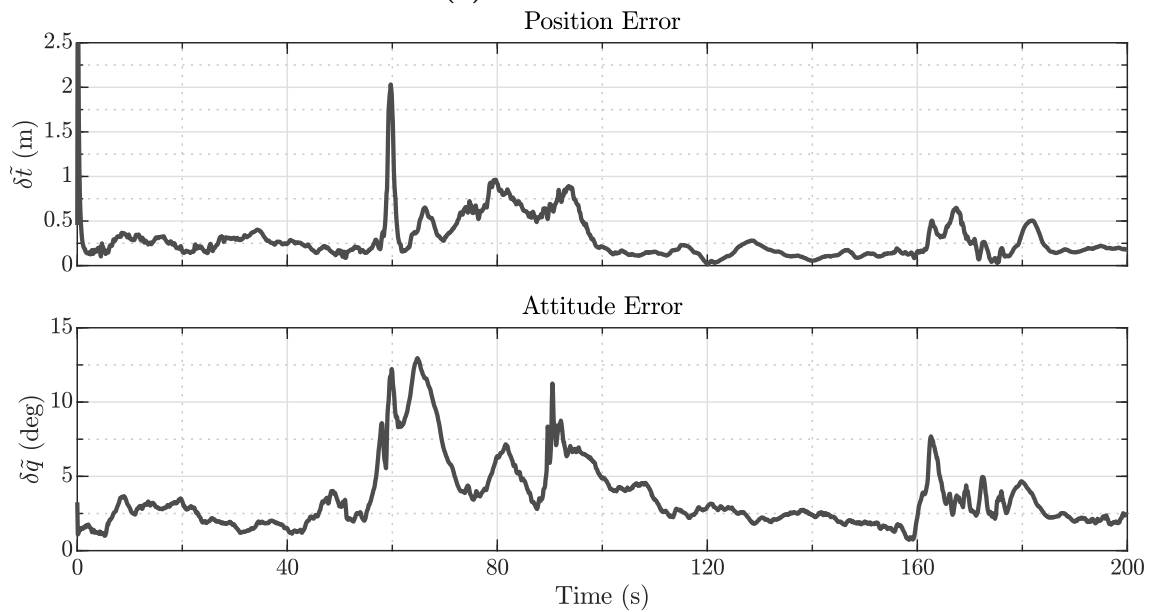
6.4.7 Summary of Performance on Astos Dataset

Table 6.8 compiles the error statistics for the performance of the complete multimodal DRCNN framework on the entire ASTOS dataset. For completeness, the performance on the nominal sample sequences is also benchmarked in Figure 6.16, and illustrated on sample frames in Figure 6.17.

Looking at the metrics for G2/R1/VBAR/HOT, the performance of ChiNet can be directly compared with the algorithm developed in Chapter 5 (herein referred to as “classical”). It can be seen that ChiNet provides an estimate of the position

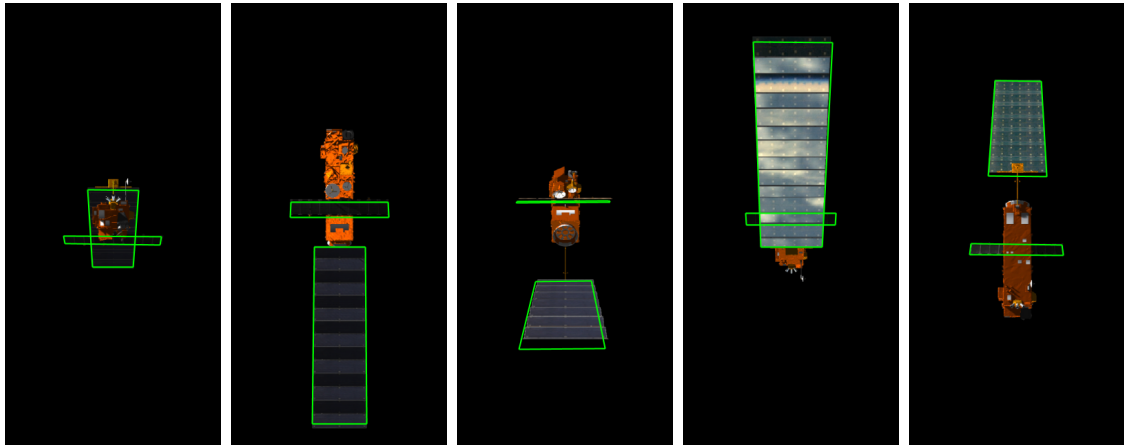


(a) G2/R1/VBAR/HOT

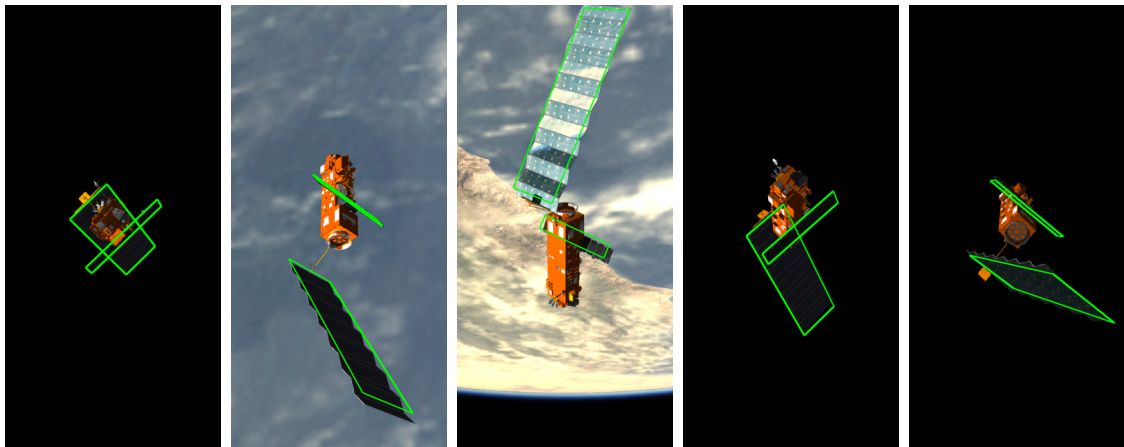


(b) G3/R2/HOT

Figure 6.16: Estimated position and attitude errors over time on two sample ASTOS dataset rendezvous sequences for the complete multimodal deep recurrent convolutional neural network (DRCNN).



(a) ASTOS/G2/R1/VBAR/HOT



(b) ASTOS/G3/R2/HOT

Figure 6.17: Qualitative pose estimation performance on frames of two sample ASTOS dataset rendezvous sequences for the complete multimodal deep recurrent convolutional neural network (DRCNN). Illustrated on red-green-blue (RGB) images, with the model reprojected in green.

with an error bound at 0.6 m, scoring on average a mean $\delta\tilde{t}_r = 0.49\%$; the classical solution, on the other hand, was seen to have reached maximum values of 2.5 m. For this trajectory, ChiNet presents an improvement of around 2.2 percentage points in terms of mean range-normalised position error. The classical solution performs better in terms of mean attitude error (0.78 deg). Still, ChiNet produces a solution not exceeding 2 deg in error.

Considering the remaining sequences within guidance profile G2 (fixed relative range), it can be seen that the quality of the solution degrades as more challenging rotation modes are considered. The estimation of the attitude appears to be more affected by this factor. For mode R2 (two-axis rotation), the pose errors are comparable to R1, even despite the benchmark of the former being performed on an RBAR approach vector (i.e. with Earth in the FOV). Mode R3 (precession) experiences by

far the largest degradation, with the mean attitude error exceeding 10.5 deg. On sequences featuring this rotation mode, the edge of the solar array leaves the FOV for a considerable amount of time, which could explain the higher error.

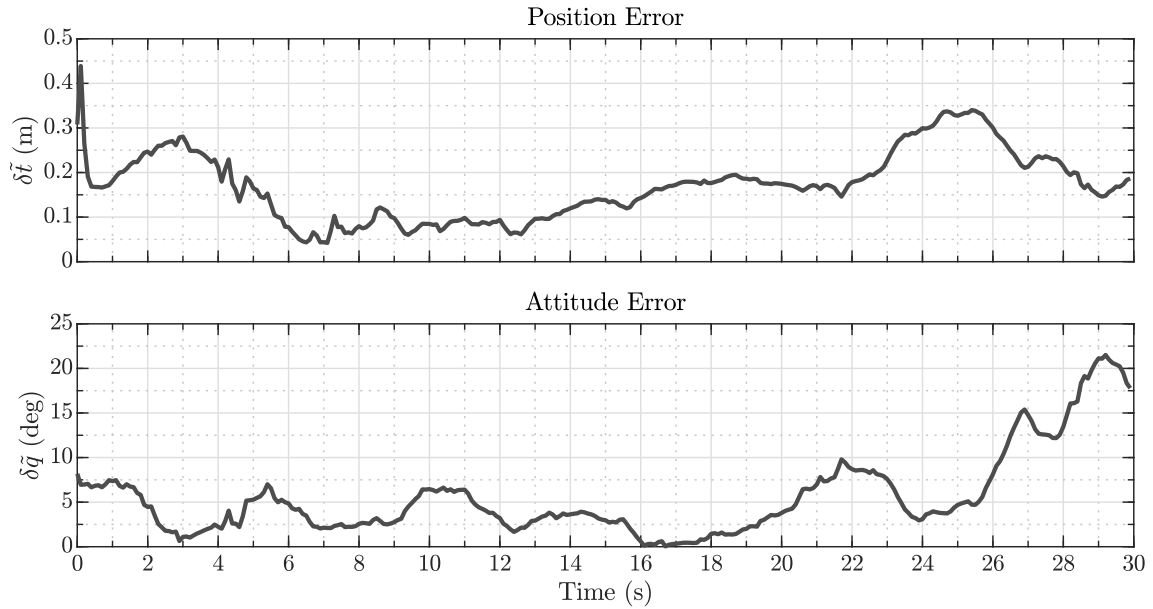
The G1 profile (forced translation) exhibits reduced performance when compared to G2. This was expected, however, since the network sees far more examples of the relative pose at a distance of 50 m than at larger distances. Nevertheless, for this profile ChiNet produces estimates of the position with mean $\delta\tilde{t}_r$ not exceeding 5.5%. The mean attitude error is less affected by the change in guidance profile, being 1.5–4 \times higher with respect to G2. Taking G1/R2/VBAR/HOT as an example, the mean $\delta\tilde{t}_r = 5.47\%$ is approximately 2.7 percent points higher than the output of the classical algorithm. The mean attitude error is also higher (3.4 \times).

Table 6.8 also illustrates the results obtained for the cold cases. RBAR trajectories are tested on RGB inputs to account for the limitations of the LWIR band in those conditions (see § 6.4.6.1). An overall increase in error is observed for all sequences when compared to the nominal hot cases. Comparing both illumination conditions for G2/R1/VBAR, it can be seen that, for the scenario where Earth is not present in the FOV and the tumbling is limited to one axis, the hot and cold solutions are comparable. Changing the approach vector (G2/R2/RBAR) or the tumbling mode (G2/R3/VBAR) widens the gap comparatively to the hot case. Both position and attitude are shown to be more affected by the eclipse for RBAR sequences.

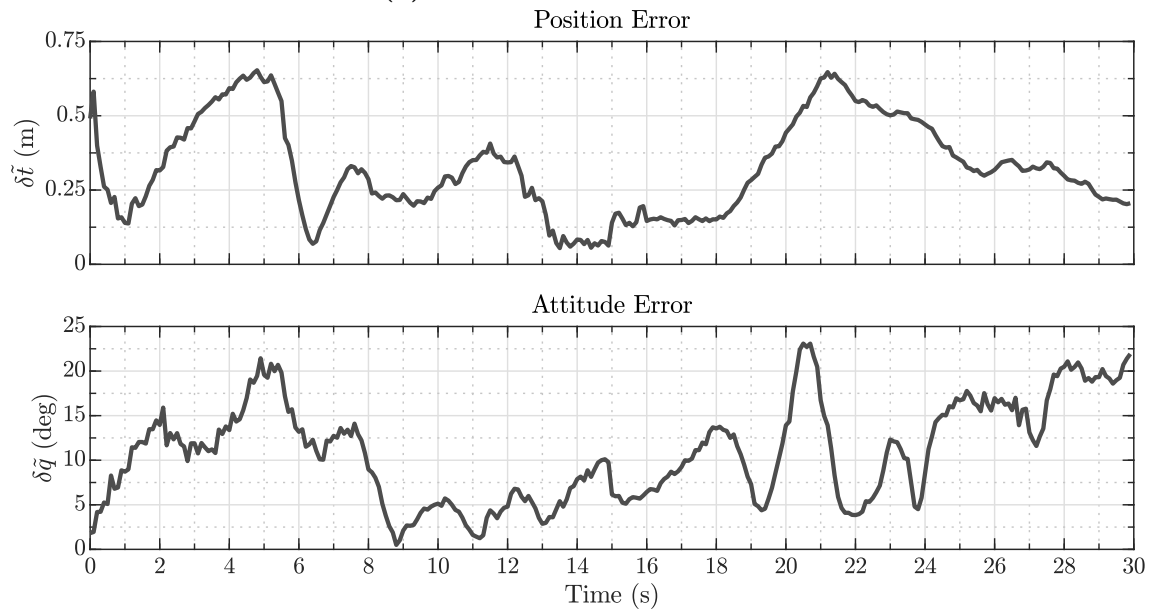
When relative translation changes are active (guidance profile G1), both position and attitude estimates increase in the cold case (with the latter more affected). ChiNet is able to produce a solution for all proposed sequences except for G1/R3/RBAR/COLD, where the attitude does not converge. This is arguably the hardest sequence, combining large variations in range, the precession tumbling mode, presence of Earth in the FOV, and low illumination. Nevertheless, the pipeline is still able to coarsely estimate the position with a mean range-normalised position error of approximately 7%.

6.4.8 Evaluation on Laboratory Data

Lastly, the performance of the complete ChiNet pipeline is assessed on data from the CITY dataset. This test provides insight on how well the deep learning framework can adapt to data captured by actual sensors, and to the sources of error a laboratory setup brings, namely: 1) camera calibration; 2) ground truth measurement; 3) camera misalignments; 4) camera synchronisation; and 5) sensor noise. It also evaluates how the network fares against previously unseen motion when trained on reduced amounts of data.



(a) CITY/APPROACH-SLOW/HOT



(b) CITY/APPROACH-SLOW/COLD

Figure 6.18: Estimated position and attitude errors over time on the CITY dataset laboratory test rendezvous sequences. Both models are trained on the full deep recurrent convolutional neural network (DRCNN) pipeline with multimodal red-green-blue-thermal (RGBT) inputs and on Stages 1, 2, and 3.

Table 6.9: Summary of position and attitude error statistics on the CITY dataset laboratory test rendezvous sequences. Both models are trained on the full deep recurrent convolutional neural network (DRCNN) pipeline with multimodal red-green-blue-thermal (RGBT) inputs and on Stages 1, 2, and 3.

Sequence	$\delta \tilde{t}$ (m)		$\delta \tilde{t}_r$ (-)		$\delta \tilde{q}$ (deg)	
	Mean	Median	Mean	Median	Mean	Median
CITY/APPROACH-SLOW/HOT	0.17	0.17	0.0634	0.0630	5.52	3.97
CITY/APPROACH-SLOW/COLD	0.32	0.31	0.1144	0.1084	10.86	11.13

Figure 6.18a illustrates the evolution in time of the position and attitude estimation errors for the test sequence CITY/APPROACH-SLOW/HOT under simulated sunlight conditions. Table 6.9 summarises these results numerically, whereas Figure 6.19a does it qualitatively. It can be observed that the position error is bounded at 35 cm throughout the trajectory, save for the initial transient period. The mean error is shown to be approximately half of that, which corresponds to a figure below 6.5% of range. The attitude error is kept below 10 deg for the first 85% of the sequence, demonstrating that the network is mostly able to separate the translational motion from the rotational one; a degradation of the estimate is observed during the last 4 s, when the target reaches a rotation of 180 deg around the spin axis and the error peaks at about 20 deg, which can be explained by the fact that the training data is biased towards an observation of that specific attitude for larger relative distances. The mean error is approximately 5.5 deg.

Figure 6.18b portrays the attained results for the same trajectory when under simulated eclipse conditions. Similarly to the synthetic tests on ASTOS, the input corresponding to the RGB image is adjusted to increase the visibility of the target. As expected, a general degradation of the solution is observed, but ChiNet is able to keep the errors bounded. As in the previous case, the attitude estimation error increases towards the end of the sequence; however, this is also observed in this case for approximately the first 8 s. It is reminded, though, that the pipeline has not been trained with eclipsed data. Both position and attitude errors are $2 \times$ higher compared to the hot case both in terms of the mean and median (the latter slightly more so in the case of the attitude). Figure 6.19b qualitatively shows the obtained solution.

6.5 Conclusions and Future Work

This chapter presented ChiNet: this thesis' contribution towards deep learning-based, end-to-end spacecraft pose estimation. The proposed method employs a CNN as a

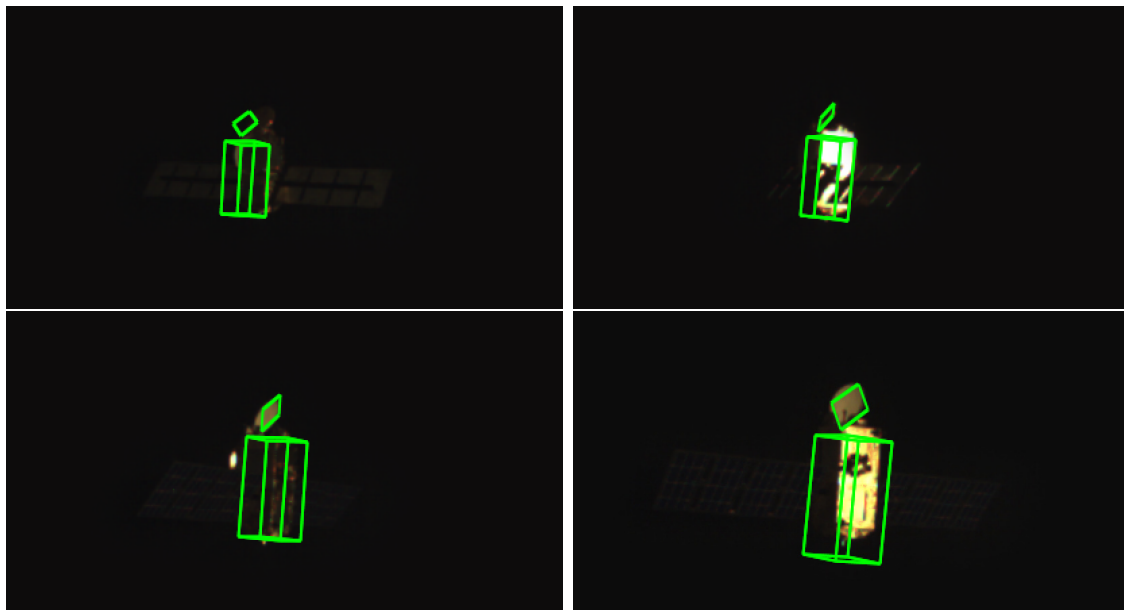
front-end feature extractor and applies an LSTM-based RNN back-end to model the temporal relationship between incoming frames from an optical camera. Furthermore, images on the visible spectrum are augmented with those captured in the LWIR band, granting a feature-rich multimodal input. The full pipeline is trained according to a multistage optimisation scheme that categorises the learning process in a coarse to fine fashion.

Each of the proposed contributions was individually tested on realistic synthetic data. The addition of the coarse training stage was demonstrated to mitigate spikes in the pose estimation errors originating from sharp reflections of both Earth and sunlight on the solar array, particularly for the estimation of the relative position. Including the keypoint-based refinement stage was shown to improve the median position error, as well as the mean and median attitude error, at some cost of the mean position performance in the case where Earth is present in the background. The recurrent module eliminated sharp jumps in the estimate of the position, reducing the mean error by half. The attitude estimate did also become more stable, at a slight cost in the mean and median error values. The inclusion of multimodal RGBT image inputs was shown to improve the mean position error in 70–80% and to reduce the mean attitude error in half on nominal cases. Some limitations of the LWIR band were identified, however, namely when a module largely uniform in terms of temperature such as the solar array dominates the FOV. Overall, ChiNet was shown to generalise well to unseen trajectories, benchmarking a mean range-normalised position error of 2.5% per average trajectory and a mean attitude estimation error of 6.9 deg per average trajectory on the sequences of the ASTOS dataset under nominal illumination conditions. The simplest case was shown to be comparable to the classical solution developed in Chapter 5, even surpassing it in terms of position estimation performance. The pipeline required no localisation or segmentation preprocessing to produce an accurate solution.

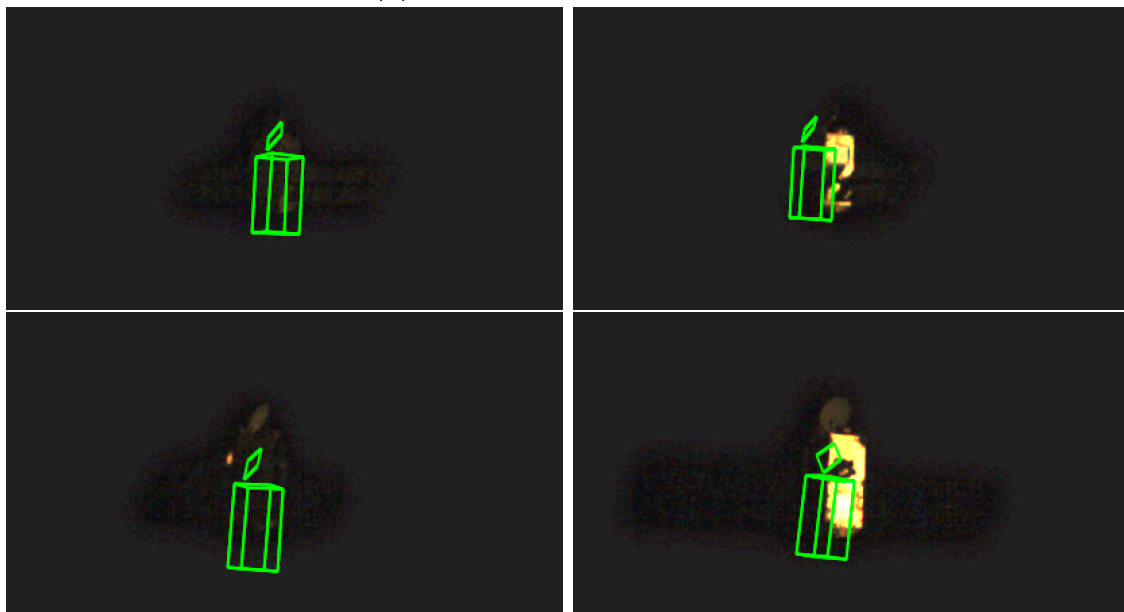
The network was further stressed by subjecting it to tests on eclipse sequences of the ASTOS dataset which had not been seen during training, including a different target thermal signature. This highlighted some limitations of the method in the sense that the image intensity values of the test images had to be adjusted to produce a working solution; the use of RGBT inputs was also shown to fail in the presence of Earth as the thermal signature of the target in eclipse made it too difficult to distinguish it from the background in the LWIR band. Despite generally lower performance, ChiNet still produced feasible pose estimate solutions that could be complemented by additional on-board sensors, save for a single case where multiple hampering factors were at play. Lastly, the proposed work was benchmarked on

a laboratory dataset, demonstrating the capability of the network to learn novel situations under a reduced training regime.

Future work might tackle additional image augmentation strategies, such as localised *IP*-based manipulation of the target in the *FOV*, or background replacement, to robustify the pipeline towards the highlighted cases where the presence of Earth affects the perception of the target, particularly in the *LWIR* modality during eclipse. Another potential avenue to investigate could be the contribution of domain adaptation (Csurka, 2017) in the context of spacecraft pose estimation, whereby a deep network is trained with synthetic images and tested on real data, as the latter are typically scarce prior to the actual mission, but the former can be generated in large quantities.



(a) CITY/APPROACH-SLOW/HOT



(b) CITY/APPROACH-SLOW/COLD

Figure 6.19: Qualitative pose estimation performance on frames of the CITY dataset laboratory test rendezvous sequences for the complete multimodal deep recurrent convolutional neural network (DRCNN). Illustrated on red-green-blue (RGB) images, with the model reprojected in green. The eclipse images have been enhanced for visibility.

CHAPTER 7

Conclusion

7.1 Overview

Vision-based relative navigation has been used for over four decades to guide man-made vessels throughout deep space towards the exploration of other worlds in our solar system. Due to their cheapness, small form factor, and capability of on-board processing, cameras are nowadays almost always considered a necessity when performing an autonomous far-range rendezvous (RV) with a celestial body, be it a planet, a moon, or an asteroid. In addition, cameras can serve a dual purpose as sensor for both navigation and scientific objectives.

However, for close range RVs, expensive active sensors such as lidar are still the norm, and vision is traditionally reserved for supporting functions regarding cooperative targets. Nonetheless, the ongoing democratisation of space might soon shift the current paradigm towards generalised, but inexpensive, autonomy as new solutions are required to manage both new and current satellite missions.

This dissertation recognised this as a motivator and sought to advance the state-of-the-art in vision-based non-cooperative rendezvous (NCRV), with a particular focus on active debris removal (ADR) applications. This task was endeavoured by focusing on narrowing the gap between the computer vision and space domains, while acknowledging the challenging and specific conditions of an NCRV operation. Such challenges were tackled in a structured approach through the investigation of model-based methods for the estimation of the six degrees-of-freedom (6-DOF) pose of an artificial, known target relative to a camera mounted on a chaser spacecraft, and by looking not only at the visible modality, but beyond it, incorporating long-wavelength infrared (LWIR) measurements into the solution.

7.2 Summary and Discussion

In Chapter 1, the objectives of this thesis were formulated in the form of three research questions, which are now revisited as the work closes with a summary of novel contributions brought by each chapter and a discussion of envisaged future work.

The first contribution of this thesis was done in the form of data generation. Unlike for ground or aerial applications, image-based datasets for space *RV* are limited or non-existent, and real images are very expensive to obtain. As such, the Astos dataset, presented at the end of Chapter 2, was created, consisting in a series of simulated trajectories with Envisat under various configurations. In total, 14 different trajectories were devised, where the target was imaged during sunlit and eclipsed periods with a visible and thermal camera, generating 56 different sequences. The Astos dataset was fundamental for appropriately characterising the task at hand, and was used for validation in all subsequent experiments.

Chapter 3 presented, for the first time, a multimodal analysis of state-of-the-art keypoint detectors and descriptors in the context of an *NCRV* with an artificial body. Several key findings were identified that demonstrated the benefits of the *LWIR* modality relative to the visible, namely a generalised increase in feature repeatability, and a better matching score during eclipse sequences. The analysis also showed some limitations in terms of using keypoints in model-based strategies, as the overall performance decreased on tests of wide baseline transformations. Observed results motivated the combination of certain detectors with non-native descriptors. The algorithms were benchmarked on an embedded board with low processing power to emulate the limited resources of an on-board guidance, navigation and control (*GNC*) system, where good trade-offs were identified while leaving 80% of the computational budget for other tasks. Given the swath of the conducted analysis, unprecedented for its domain, the first proposed research question [RQ1], which read:

How do low-level image processing algorithms behave on images acquired during a space rendezvous?

is considered to be answered.

The first contribution towards the task of pose estimation in the visible band was offered in Chapter 4, where the problem was decomposed into two parts: an offline training phase, where images of the target were rendered using a computer-aided design (*CAD*) model and annotated with local 3D information; and an online estimation phase, where point and edge contour features of the target were detected during the rendezvous sequence and matched to the training images. The resulting

2D-3D constraints were then jointly minimised under an M-estimation framework to simultaneously reject outlying correspondences and estimate the pose. The importance of the findings stands on the fact that it was shown that a solution could be obtained for a tumbling target based on a limited number of keyframes and CPU-bound (central processing unit) processing, in opposition to alternative methods which rely on GPU-based (graphics processing unit) hardware-accelerated renderings of the target model in real-time to deal with complex motions.

In Chapter 5, two main improvements to the earlier algorithm were proposed. The first one tackled the problem of keyframe initialisation through the development of a coarse viewpoint classification method based on global Zernike moment (ZM) features of the target’s shape. The second one employed an extended Kalman filter (EKF) to improve the nominal solution by fusing the individual M-estimates generated by each feature type. Both the M-estimation and filtering schemes were linearised on the tangent space of SE(3), which served two predominant purposes: 1) the internal update equations were formulated in terms of the 6-DOF error state and then converted to the nominal state via the exponential map, naturally adhering to the constraints of higher-dimensional representations of the attitude such as the unit quaternion; and 2) because the error states followed the minimal representation of the attitude, the covariance obtained as a by-product of the M-estimation solution could be directly input as the measurement noise to the EKF, providing a natural way to balance the contributions of each feature type. The filter prediction was in turn used to reduce the search space of features for the following time-step, further improving the solution.

This revamped method was validated on synthetic and laboratory-generated NCRV sequences with Envisat, featuring different tumbling modes, guidance profiles, and detrimental illumination effects, where average error values of 2.5 % of the range for the position and 1 deg for the attitude were obtained. These metrics, attained with only a digital camera as a sensor, are extremely close to Fehse’s (2003) “1 % of range” rule of thumb, and are well within the accuracy requirements of established cooperative systems (cf. Tab. 1.2, Chap. 1). The proposed approach was compared to existing alternatives (including model-free ones) and shown to perform better, but dependant on the segmentation between target and background; in particular, it was benchmarked on the Spacecraft PosE Estimation Dataset (SPEED), where it was demonstrated to be competitive with state-of-the-art CNN-based (convolutional neural network) methods on laboratory data — even better than most of the Satellite Pose Estimation Challenge (SPEC) entrants. In this light, the second research question [RQ2], originally stated as:

Can a contribution be made towards model-based spacecraft relative pose estimation in the visible wavelength?

is taken to be positively answered.

The final novelty proposed by this dissertation explored the realm of deep learning through the introduction of ChiNet in Chapter 2 as an end-to-end data-driven spacecraft pose estimation approach. The architecture combined the power of CNNs to process raw images with an LSTM-based (long short-term memory) back-end to model the temporal relations arising from sequential inputs, thus introducing the first deep recurrent convolutional neural network (DRCNN) for vision-based target-centred navigation in space. Taking inspiration from the work done in previous chapters, an optimisation framework comprising a coarse to fine multiple stage scheme was used to train the network. Each of the proposed contributions was demonstrated individually on models trained on a dataset split of the Astos collection.

Primarily, the framework was used to evaluate the contribution of the LWIR band by training two models: one with red-green-blue (RGB) inputs and a second with red-green-blue-thermal (RGBT) inputs. The multimodal solution showed a generalised improvement in the estimated pose relative to the visible alone. In average terms, the performance of ChiNet was comparable to the earlier presented classical-based approach in terms of position estimation, with an inferior but acceptable benchmark in attitude estimation. ChiNet was also able to natively perform in cases where Earth was present in the camera's field of view (FOV), which the previous method could not do. Potential limitations of using RGBT inputs were clearly identified, namely when sections of the target appearing featureless in the thermal spectrum occlude the rest of the body. An additional test was made by subjecting the network to eclipse sequences, which it had not seen during training. Despite an expected degradation in the overall performance, the method was demonstrated to work on a previously unseen thermal signature of the target; it did not produce, however, a valid solution when Earth was visible due to comparable intensity values, though an RGB-based estimate was still accomplishable. Lastly, ChiNet's resilience towards training with limited available data was validated on laboratory sequences of the Jason-1 satellite. The conducted testing campaign exhaustively studied the contribution of LWIR features towards the estimation of the full 6-DOF pose, thus providing an answer to [RQ3]:

Can the long-wavelength infrared modality improve vision-based six degrees-of-freedom relative navigation? If so, how?

which was considered to be the final research question.

Through a structured approach and rigorous experimentation, this dissertation has illustrated the capability of vision-based sensing for close-range rendezvous with non-cooperative targets, both in the visible domain alone and following a multimodal approach with support of the thermal band, publishing several papers in the process with the aspiration of pushing orbital vehicles in the direction of robust, long-term autonomous navigation adequate for today's modern Space Age.

7.3 Future Work

This thesis proposed several model-based insights and solutions towards the advancement of the vision-based spacecraft relative pose estimation problem. However, research remains to be done in order to thoroughly cover all expected hurdles. In this section, some potential future avenues for investigation are briefly covered.

Foreground-Background Segmentation

One of the key limitations of the method proposed in Chapters 4 and 5 was the reliance on a correct extraction of the target's shape. While mostly simple in the case of a black, deep space background, this becomes a non-trivial task when Earth is in the background. The addition of a dedicated foreground-background segmentation module could therefore allow the extension of the presented method to such challenging cases. Gaussian mixture modelling (GMM), which was used in this thesis, has been previously employed in segmentation problems (Azzam et al., 2016), but is normally reserved for cases where the camera is static. In the case of space RVs, a more adequate approach could involve explicitly accounting for the camera's egomotion (Peleg and Rom, 1990).

Generative Adversarial Networks

Generative adversarial networks (GANs; I. J. Goodfellow et al., 2014) are a framework for estimating models — typically implemented using deep neural networks (DNN) — for synthesising images. It involves the training of two subnetworks: a generator, which creates samples by inputting random noise through the pipeline; and a discriminator, which detects whether the sample came from the model distribution or the data distribution. Recently, Khan et al. (2018) have proposed the use of GANs to artificially boost the number of channels in an input image, thus improving the diversity representation of the data. This could be seen as an extension of the multimodal imaging concept.

A particular type of generative architecture is the CycleGAN (Zhu et al., 2017), which allows for the unsupervised training of image-to-image translation models

without paired examples. Yun et al. (2019) have recently investigated the use of CycleGANs for visible-IR (infrared) image translation. Such a model could potentially be used to synthesize new views of a RV sequence under different wavelengths to serve as training samples for a multimodal framework, such as the one proposed in this thesis, when available data is limited.

Domain Adaptation

Chapter 5 explored the use of synthetic images to derive a model which was then tested on real data according to a knowledge-driven logic. However, this remains a challenging feat for data-driven methods (i.e. deep learning), as synthetic images are only similar to real images up to a certain point. DNN models for space RV applications would greatly benefit from solving this problem of domain adaptation since real training images are quite burdensome and expensive to obtain, whereas synthetic images have the potential to yield essentially unlimited training samples. Domain adaptation has been investigated for deep learning-based object pose estimation by learning mappings from the synthetic feature space to the real feature space (Rad et al., 2018). The author is currently exploring the extent of the domain gap specifically for spacecraft pose estimation, as noted in Chapter 1 under paper [C3] (Hogan et al., 2021).

Bibliography

This document contains 261 references.

(Abderrahim et al., 2005)

Abderrahim, M., Diaz, J., Rossi, C., and Salichs, M. (2005). “Experimental Simulation of Satellite Relative Navigation Using Computer Vision”. In: *Proceedings of 2nd International Conference on Recent Advances in Space Technologies (RAST)*. IEEE. DOI: 10.1109/rast.2005.1512596.

(Absil et al., 2008)

Absil, P.-A., Mahony, R., and Sepulchre, R. (Dec. 2008). *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, pp. 54–56. DOI: 10.1515/9781400830244.

(Agarwal et al., 2020)

Agarwal, S. et al. (2020). *Ford Multi-AV Seasonal Dataset*. arXiv: 2003.07969 [cs.R0]. URL: <https://avdata.ford.com>.

(Aglietti et al., 2020)

Aglietti, G. S. et al. (2020). “The active space debris removal mission RemoveDebris. Part 2: In orbit operations”. In: *Acta Astronautica* 168, pp. 310–322. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2019.09.001>.

(Agrawal et al., 2008)

Agrawal, M., Konolige, K., and Blas, M. R. (2008). “CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching”. In: *Computer Vision – ECCV 2008*. Ed. by D. Forsyth, P. Torr, and A. Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 102–115. DOI: 10.1007/978-3-540-88693-8_8.

(Akinlar and Topal, 2011)

Akinlar, C. and Topal, C. (Oct. 2011). “EDLines: A Real-Time Line Segment Detector with a False Detection Control”. In: *Pat-*

tern Recognition Letters 32.13, pp. 1633–1642. DOI: 10.1016/j.patrec.2011.06.001.

(Alahi et al., 2012)

Alahi, A., Ortiz, R., and Vandergheynst, P. (June 2012). “FREAK: Fast Retina Keypoint”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. DOI: 10.1109/cvpr.2012.6247715.

(Alcantarilla et al., 2013)

Alcantarilla, P., Nuevo, J., and Bartoli, A. (2013). “Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces”. In: *Proceedings of the British Machine Vision Conference (BMVC) 2013*. British Machine Vision Association. DOI: 10.5244/c.27.13.

(Andrenucci et al., 2011)

Andrenucci, M., Pergola, P., and Ruggiero, A. (2011). *Active Removal of Space Debris - Expanding foam application for active debris removal*. Tech. rep. 10-6411. Available on line at www.esa.int/act. Noordwijk, NL: European Space Agency. URL: http://www.esa.int/gsp/ACT/doc/ARI/ARI%20Study%20Report/ACT-RPT-MAD-ARI-10-6411-Pisa-Active_Removal_of_Space_Debris-Foam.pdf (visited on 02/01/2020).

(Augenstein and Rock, 2009)

Augenstein, S. and Rock, S. (Aug. 2009). “Simultaneous Estimation of Target Pose and 3-D Shape Using the FastSLAM Algorithm”. In: *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2009-5782.

(Azzam et al., 2016)

Azzam, R., Kemouche, M., Aouf, N., and Richardson, M. (2016). “Efficient visual object detection with spatially global Gaussian mixture models and uncertainties”. In: *Journal of Visual Communication and Image Representation* 36, pp. 90–106. ISSN: 1047-3203. DOI: <https://doi.org/10.1016/j.jvcir.2015.11.009>.

(J. L. Ba et al., 2016)

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). *Layer Normalization*. arXiv: 1607.06450 [stat.ML].

- (Barfoot, 2017) Barfoot, T. D. (2017). *State Estimation for Robotics*. 1st ed. New York, NY: Cambridge University Press, pp. 173–174, 265, 359. ISBN: 1107159393, 9781107159396. DOI: 10.1017/97811316671528.
- (Bay et al., 2006) Bay, H., Tuytelaars, T., and Van Gool, L. (2006). “SURF: Speeded Up Robust Features”. In: *European Conference on Computer Vision – ECCV 2006*. Springer Berlin Heidelberg, pp. 404–417. DOI: 10.1007/11744023_32.
- (Beaton and Tukey, 1974) Beaton, A. E. and Tukey, J. W. (May 1974). “The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data”. In: *Technometrics* 16.2, pp. 147–185. DOI: 10.1080/00401706.1974.10489171.
- (Bennett, 1970) Bennett, F. (Jan. 1970). “Lunar Descent and Ascent Trajectories”. In: *8th Aerospace Sciences Meeting*. West Germany: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.1970-25.
- (Bhaskaran et al., 1998) Bhaskaran, S. et al. (1998). “Orbit Determination Performance Evaluation of the Deep Space 1 Autonomous Navigation System”. In: *AAS/AIAA Space Flight Mechanics Meeting*. AAS 98-193. AAS Spaceflight Mechanics Technical Committee and AIAA Astrodynamics Technical Committee. Monterey, CA.
- (Biesbroek, Innocenti, et al., 2017) Biesbroek, R., Innocenti, L., Wolahan, A., and Serrano, S. M. (2017). “e.Deorbit – ESA’s Active Debris Removal Mission”. In: *7th European Conference on Space Debris*. Ed. by T. Flohrer and F. Schmitz. Vol. 7. ESA Space Debris Office. URL: <https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/1053> (visited on 02/01/2020).
- (Biesbroek, Wolahan, et al., 2017) Biesbroek, R., Wolahan, A., and Serrano, S. M. (Oct. 2017). *e.Inspector*. ESA Clean Space Industrial Days. Noordwijk, The Netherlands. URL: https://indico.esa.int/event/181/contributions/1378/attachments/1305/1530/e.Inspector_SARA.pdf (visited on 02/19/2020).

- (Bishop, 2006) Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. 1st ed. Berlin, Heidelberg: Springer-Verlag, pp. 227–228. DOI: 10.5555/1162264.
- (Blanco, 2019) Blanco, J.-L. (Mar. 2019). *A Tutorial on SE(3) Transformation Parameterizations and On-Manifold Optimization*. Tech. rep. University of Málaga. URL: <https://w3.ual.es/~jlblanco/publications/#publications>.
- (Blanco et al., 2010) Blanco, J.-L., Gonzalez, J., and Fernández-Madrigal, J. A. (2010). *An Experimental Comparison of Image Feature Detectors and Descriptors Applied to Grid Map Matching*. Tech. rep. University of Malaga, Spain.
- (Boden, 1999) Boden, D. G. (1999). “Introduction to Astrodynamics”. In: *Space Mission Analysis and Design*. Ed. by J. R. Wertz and W. J. Larson. 3rd edition. Space Technology Series. El Segundo, CA: Microcosm Press. Chap. 6, pp. 132, 153.
- (Bodin et al., 2012) Bodin, P. et al. (Feb. 2012). “PRISMA Formation Flying Demonstrator: Overview and Conclusions from the Nominal Mission”. In: *35th Annual AAS Guidance and Control Conference* (Feb. 3, 2012). AAS 12-072. Breckenridge, CO. URL: <https://elib.dlr.de/80862/>.
- (Bonnal et al., 2013) Bonnal, C., Ruault, J.-M., and Desjean, M.-C. (2013). “Active Debris Removal: Recent Progress and Current Trends”. In: *Acta Astronautica* 85, pp. 51–60. DOI: 10.1016/j.actaastro.2012.11.009.
- (Boulekhour and Aouf, 2014) Boulekhour, M. and Aouf, N. (June 2014). “Robust Motion Estimation Using Covariance Intersection”. In: *22nd Mediterranean Conference on Control and Automation*. IEEE. DOI: 10.1109/med.2014.6961507.
- (Breuers, 1999) Breuers, M. G. J. (Aug. 1999). “Image-based Aircraft Pose Estimation using Moment Invariants”. In: *Automatic Target Recognition IX*. Ed. by F. A. Sadjadi. SPIE. DOI: 10.1117/12.359963.
- (Cai et al., 2015) Cai, J., Huang, P., Zhang, B., and Wang, D. (Dec. 2015). “A TSR Visual Servoing System Based on a Novel Dynamic Template

- Matching Method”. In: *Sensors* 15.12, pp. 32152–32167. DOI: 10.3390/s151229884.
- (Calonder et al., 2010)
Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). “BRIEF: Binary Robust Independent Elementary Features”. In: *Computer Vision – ECCV 2010*. Springer Berlin Heidelberg, pp. 778–792. DOI: 10.1007/978-3-642-15561-1_56.
- (Campbell et al., 2017)
Campbell, T., Furfaro, R., Linares, R., and Gaylor, D. (2017). “A Deep Learning Approach for Optical Autonomous Planetary Relative Terrain Navigation”. In: *27th AAS/AIAA Space Flight Mechanics Meeting, 2017*. Univelt Inc., pp. 3293–3302.
- (Canny, 1987)
Canny, J. (1987). “A Computational Approach to Edge Detection”. In: *Readings in Computer Vision*. Elsevier, pp. 184–203. DOI: 10.1016/b978-0-08-051581-6.50024-6.
- (Cassinis et al., 2020)
Cassinis, L. P., Fonod, R., Gill, E., Ahrns, I., and Fernandez, J. G. (Jan. 2020). “CNN-Based Pose Estimation System for Close-Proximity Operations Around Uncooperative Spacecraft”. In: *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2020-1457.
- (Castellini et al., 2015)
Castellini, F., Antal-Wokes, D., Santayana, R. P. de, and Vantournhout, K. (2015). “Far Approach Optical Navigation and Comet Photometry for the Rosetta Mission”. In: *Proceedings of the 25th International Symposium on Space Flight Dynamics*. DLR German Space Operations Center (GSOC) and ESA European Space Operations Centre (ESOC). Munich, Germany.
- (Cavrois et al., 2015)
Cavrois, B., Vergnol, A., Donnard, A., Casiez, P., and Mongrard, O. (Jan. 2015). “LIRIS demonstrator on ATV5: a step beyond for European non cooperative navigation system”. In: *2015 AIAA Guidance, Navigation, and Control Conference*. Kissimmee, FL: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2015-0336.
- (Chen et al., 2019)
Chen, B., Cao, J., Parra, A., and Chin, T.-J. (Oct. 2019). “Satel-

- lite Pose Estimation with Deep Landmark Regression and Non-linear Pose Refinement”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. DOI: 10.1109/iccvw.2019.00343.
- (Chesley et al., 1999)
Chesley, B., Lutz, R., and Brodsky, R. F. (1999). “Space Payload Design and Sizing”. In: *Space Mission Analysis and Design*. Ed. by J. R. Wertz and W. J. Larson. 3rd edition. Space Technology Series. El Segundo, CA: Microcosm Press. Chap. 9, p. 243.
- (Chien, 1992)
Chien, C.-H. (1992). “Multiview-based pose estimation from range images”. In: *Cooperative Intelligent Robotics in Space III*. Ed. by J. D. Erickson. Vol. 1829. International Society for Optics and Photonics. SPIE, pp. 421–432. DOI: 10.1117/12.131719.
- (Chong and Zak, 2013)
Chong, E. K. and Zak, S. H. (2013). *An Introduction to Optimization*. 4th ed. Hoboken, NJ: John Wiley & Sons, pp. 81–93. ISBN: 978-1-118-27901-4.
- (J. Christian et al., 2011)
Christian, J., Hinkel, H., Maguire, S., D’Souza, C., and Patangan, M. (Aug. 2011). “The Sensor Test for Orion RelNav Risk Mitigation (STORRM) Development Test Objective”. In: *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, p. 6260. DOI: 10.2514/6.2011-6260.
- (J. A. Christian and Cryan, 2013)
Christian, J. A. and Cryan, S. (Aug. 2013). “A Survey of LIDAR Technology and its Use in Spacecraft Relative Navigation”. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2013-4641. URL: <https://doi.org/10.2514%2F6.2013-4641>.
- (Clark et al., 2017)
Clark, R., Wang, S., Wen, H., Markham, A., and Trigoni, N. (2017). *VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem*. arXiv: 1701.08376 [cs.CV].
- (Comport et al., 2006)
Comport, A. I., Marchand, E., Pressigout, M., and Chaumette, F.

- (2006). “Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.4, pp. 615–628. DOI: 10.1109/tvcg.2006.78.
- (Courtois and Aouf, 2017)
Courtois, H. and Aouf, N. (Oct. 2017). “Fusion of Stereo and Lidar Data for Dense Depth Map Computation”. In: *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE. DOI: 10.1109/red-uas.2017.8101664.
- (Cowan et al., 2016)
Cowan, B., Imanberdiyev, N., Fu, C., Dong, Y., and Kayacan, E. (Nov. 2016). “A Performance Evaluation of Detectors and Descriptors for UAV Visual Tracking”. In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE. DOI: 10.1109/icarcv.2016.7838649.
- (Cropp, 2001)
Cropp, A. (2001). “Pose Estimation and Relative Orbit Determination of a Nearby Target Microsatellite using Passive Imagery”. PhD thesis. United Kingdom: University of Surrey. URL: <http://epubs.surrey.ac.uk/843875/>.
- (Csurka, 2017)
Csurka, G. (2017). *Domain Adaptation for Visual Applications: A Comprehensive Survey*. arXiv: 1702.05374 [cs.CV].
- (Csurka et al., 2004)
Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). “Visual Categorization with Bags of Keypoints”. In: *8th European Conference on Computer Vision (ECCV)*. Vol. 1. 1-22. Prague, Czech Republic, pp. 1–2.
- (David et al., 2004)
David, P., DeMenthon, D., Duraiswami, R., and Sament, H. (Sept. 2004). “SoftPOSIT: Simultaneous Pose and Correspondence Determination”. In: *International Journal of Computer Vision* 59.3, pp. 259–284. DOI: 10.1023/b:visi.0000025800.10423.1f.
- (Davison, 2003)
Davison, A. J. (2003). “Real-Time Simultaneous Localisation and Mapping with a Single Camera”. In: *Proceedings Ninth IEEE International Conference on Computer Vision*. Nice, France: IEEE. DOI: 10.1109/iccv.2003.1238654.

(Davison et al., 2007)

Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (June 2007). “MonoSLAM: Real-Time Single Camera SLAM”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6, pp. 1052–1067. DOI: 10.1109/tpami.2007.1049.

(Deng et al., 2009)

Deng, J. et al. (2009). “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

(Dhome et al., 1989)

Dhome, M., Richetin, M., Lapreste, J.-T., and Rives, G. (1989). “Determination of the Attitude of 3D Objects from a Single Perspective View”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.12, pp. 1265–1278. DOI: 10.1109/34.41365.

(Diebel, 2006)

Diebel, J. (2006). *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*. Tech. rep. Stanford, CA: Stanford University.

(Dor and Tsiotras, 2018)

Dor, M. and Tsiotras, P. (Jan. 2018). “ORB-SLAM Applied to Spacecraft Non-Cooperative Rendezvous”. In: *2018 Space Flight Mechanics Meeting*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2018-1963.

(Drummond and Cipolla, 2002)

Drummond, T. and Cipolla, R. (2002). “Real-time visual tracking of complex structures”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7, pp. 932–946. DOI: 10.1109/TPAMI.2002.1017620.

(Dubanchet, 2017)

Dubanchet, V. (2017). Computer Software. Space rendezvous simulator for the generation of relative motion trajectories between two space bodies, provided as part of the Ph.D. programme sponsorship. Cannes, France: Thales Alenia Space France.

(Dubois-Matra, 2016)

Dubois-Matra, O. (2016). *Image Processing for Navigation Us-*

- ing Low Performance Computing*. Statement of Work TEC-ECN/156.16/ODM. European Space Agency.
- (Duda, Hart, and Stork, 2012)
Duda, R. O., Hart, P. E., and Stork, D. G. (2012). “Pattern Classification”. In: 2nd. New York, NY: John Wiley & Sons, pp. 21, 124–125.
- (Duda and Hart, 1972)
Duda, R. O. and Hart, P. E. (Jan. 1972). “Use of the Hough transformation to detect lines and curves in pictures”. In: *Communications of the ACM* 15.1, pp. 11–15. DOI: 10.1145/361237.361242.
- (Dudani et al., 1977)
Dudani, S. A., Breeding, K. J., and McGhee, R. B. (Jan. 1977). “Aircraft Identification by Moment Invariants”. In: *IEEE Transactions on Computers* C-26.1, pp. 39–46. DOI: 10.1109/tc.1977.5009272.
- (Dupré, 2008)
Dupré, S. (2008). “Inside the Camera Obscura: Kepler’s Experiment and Theory of Optical Imagery”. In: *Early Science and Medicine* 13.3, pp. 219–244. DOI: <https://doi.org/10.1163/157338208X285026>.
- (Durrant-Whyte, 2001)
Durrant-Whyte, H. (Jan. 2001). *Multi Sensor Data Fusion*. Preprint. Version 1.2. pp. 75–87. New South Wales, Australia: Australian Centre for Field Robotics, The University of Sydney.
- (Duxbury and Callahan, 1988)
Duxbury, T. C. and Callahan, J. D. (July 1988). “PHOBOS and Deimos astrometric observations from Viking”. In: *Astronomy and Astrophysics* 201.1. Provided by the SAO/NASA Astrophysics Data System, pp. 169–176. URL: <https://ui.adsabs.harvard.edu/abs/1988A&A...201..169D>.
- (Engel, Usenko, et al., 2016)
Engel, J., Usenko, V., and Cremers, D. (July 2016). *A Photometrically Calibrated Benchmark For Monocular Visual Odometry*. arXiv: 1607.02555 [cs.CV].
- (Engel, Schöps, et al., 2014)
Engel, J., Schöps, T., and Cremers, D. (2014). “LSD-SLAM: Large-Scale Direct Monocular SLAM”. In: *European Conference*

- on Computer Vision – ECCV 2014*. Springer International Publishing, pp. 834–849. DOI: 10.1007/978-3-319-10605-2_54.
- (Evangelidis and Psarakis, 2008)
Evangelidis, G. D. and Psarakis, E. Z. (Oct. 2008). “Parametric Image Alignment using Enhanced Correlation Coefficient Maximization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10, pp. 1858–1865. DOI: 10.1109/tpami.2008.113.
- (Fehse, 2003)
Fehse, W. (2003). *Automated Rendezvous and Docking of Spacecraft*. 1st edition. Cambridge, UK: Cambridge University Press, pp. 1, 3, 8, 32–33, 114, 272–277. DOI: 10.1017/cbo9780511543388.
- (Ferraz et al., 2014)
Ferraz, L., Binefa, X., and Moreno-Noguer, F. (June 2014). “Very Fast Solution to the PnP Problem with Algebraic Outlier Rejection”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. DOI: 10.1109/cvpr.2014.71.
- (Figueiredo and Jain, 2002)
Figueiredo, M. and Jain, A. (Mar. 2002). “Unsupervised Learning of Finite Mixture Models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.3, pp. 381–396. DOI: 10.1109/34.990138.
- (Fischler and Bolles, 1981)
Fischler, M. A. and Bolles, R. C. (June 1981). “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* 24.6, pp. 381–395. DOI: 10.1145/358669.358692.
- (Flusser et al., 2016)
Flusser, J., Suk, T., and Zitová, B. (Dec. 2016). *2D and 3D Image Analysis by Moments*. 1st ed. John Wiley & Sons, Ltd, pp. 47–48, 352–356. DOI: 10.1002/9781119039402.
- (Forshaw et al., 2016)
Forshaw, J. L. et al. (2016). “RemoveDEBRIS: An in-orbit active debris removal demonstration mission”. In: *Acta Astronautica* 127, pp. 448–463. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2016.06.018>.

- (Furfaro et al., 2018) Furfaro, R. et al. (2018). “Deep Learning for Autonomous Lunar Landing”. In: *2018 AAS/AIAA Astrodynamics Specialist Conference*, pp. 1–22.
- (Gallier, 2011) Gallier, J. (2011). “Geometric Methods and Applications: For Computer Science and Engineering”. In: 2nd. New York, NY: Springer, pp. 468, 504–505. DOI: 10.1007/978-1-4419-9961-0.
- (Gallier and Quaintance, 2019) Gallier, J. and Quaintance, J. (Aug. 2019). *Differential Geometry and Lie Groups I: A Computational Perspective*. University of Pennsylvania (book in progress). p. 591. URL: <http://www.cis.upenn.edu/~jean/gbooks/manif.html>.
- (Gansmann et al., 2017) Gansmann, M., Mongrard, O., and Ankersen, F. (2017). “3D Model-Based Relative Pose Estimation for Rendezvous and Docking Using Edge Features”. In: *10th International ESA Conference on Guidance, Navigation and Control Systems*. Salzburg, Austria: ESA.
- (Garber, 2012) Garber, S. (Aug. 2012). *Style Guide for NASA History Authors and Editors*. URL: <https://history.nasa.gov/styleguide.html> (visited on 05/06/2021).
- (Geiger et al., 2013) Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (Aug. 2013). “Vision Meets Robotics: The KITTI Dataset”. In: *The International Journal of Robotics Research* 32.11, pp. 1231–1237. DOI: 10.1177/0278364913491297.
- (Geyer et al., 2020) Geyer, J. et al. (2020). *A2D2: Audi Autonomous Driving Dataset*. arXiv: 2004.06320 [cs.CV]. URL: <https://www.a2d2.audi>.
- (Gifford, 2014) Gifford, H. (2014). *Hierarchical k-Means for Unsupervised Learning*. Tech. rep. Carnegie Mellon University.
- (Gil et al., 2009) Gil, A., Mozos, O. M., Ballesta, M., and Reinoso, O. (Apr. 2009). “A Comparative Evaluation of Interest Point Detectors and Local Descriptors for Visual SLAM”. In: *Machine Vision and Applications* 21.6, pp. 905–920. DOI: 10.1007/s00138-009-0195-x.
- (Glasner et al., 2011) Glasner, D., Galun, M., Alpert, S., Basri, R., and Shakhnarovich,

- G. (Nov. 2011). “Viewpoint-Aware Object Detection and Pose Estimation”. In: *2011 International Conference on Computer Vision*. New York, NY: IEEE, pp. 923–933. DOI: 10.1109/iccv.2011.6126379.
- (I. Goodfellow et al., 2016)
Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, pp. 12–14, 78, 185–191, 286–291, 298–302, 526–531. DOI: 10.5555/308695.
- (I. J. Goodfellow et al., 2014)
Goodfellow, I. J. et al. (2014). *Generative Adversarial Networks*. arXiv: 1406.2661 [stat.ML].
- (Grewal and Andrews, 2015)
Grewal, M. and Andrews, A. (2015). “Kalman Filtering: Theory and Practice Using MATLAB”. In: 4th ed. Hoboken, NJ: Wiley, pp. 135–139, 231. ISBN: 9781118984987.
- (Grimm et al., 1992)
Grimm, K. A. et al. (1992). “Experiment in vision-based autonomous grasping within a reduced gravity environment”. In: *Cooperative Intelligent Robotics in Space III*. Ed. by J. D. Erickson. Vol. 1829. International Society for Optics and Photonics. SPIE, pp. 410–420. DOI: 10.1117/12.131718.
- (Hall, 2015)
Hall, B. C. (2015). *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. 2nd ed. Springer International Publishing, E1–E1. DOI: 10.1007/978-3-319-13467-3.
- (Harris and Stephens, 1988)
Harris, C. and Stephens, M. (1988). “A Combined Corner and Edge Detector”. In: *Proceedings of the Alvey Vision Conference 1988*. Alvey Vision Club. DOI: 10.5244/c.2.23.
- (Hartley and Zisserman, 2004)
Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge, UK: Cambridge University Press, pp. 32–33, 102–104, 135, 141–142, 180–181, 204–205, 257–258 434–435. DOI: 10.1017/cbo9780511811685.
- (Harvard et al., 2020)
Harvard, A., Capuano, V., Shao, E. Y., and Chung, S.-J. (Jan. 2020). “Spacecraft Pose Estimation from Monocular Images Using Neural Network Based Keypoints and Visibility Maps”. In:

- AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2020-1874.
- (He et al., 2015) He, K., Zhang, X., Ren, S., and Sun, J. (Oct. 2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE. DOI: 10.1109/iccv.2015.123.
- (He et al., 2016) He, K., Zhang, X., Ren, S., and Sun, J. (June 2016). “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2016.90.
- (Hertzberg, 2008) Hertzberg, C. (2008). “A Framework for Sparse, Non-Linear Least Squares Problems on Manifolds”. MA thesis. Bremen, Germany: Universität Bremen.
- (Hinton et al., 2012) Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv: 1207.0580 [cs.NE].
- (Ho and Newman, 2006) Ho, K. L. and Newman, P. (Sept. 2006). “Loop closure detection in SLAM by combining visual and spatial appearance”. In: *Robotics and Autonomous Systems* 54.9, pp. 740–749. DOI: 10.1016/j.robot.2006.04.016.
- (Hochreiter and Schmidhuber, 1997) Hochreiter, S. and Schmidhuber, J. (1997). “Long Short-term Memory”. In: *Neural Computation* 9.8, pp. 1735–1780.
- (Hogan et al., 2021) Hogan, M., Rondao, D., Aouf, N., and Dubois-Matra, O. (June 2021). “Using Convolutional Neural Networks for Relative Pose Estimation of a Non-Cooperative Spacecraft with Thermal Infrared Imagery”. In: *11th International ESA Conference on Guidance, Navigation and Control Systems*. Accepted manuscript. Virtual conference: ESA.
- (Holland and Welsch, 1977) Holland, P. W. and Welsch, R. E. (Jan. 1977). “Robust Regression Using Iteratively Reweighted Least-Squares”. In: *Communica-*

- tions in Statistics - Theory and Methods* 6.9, pp. 813–827. DOI: 10.1080/03610927708827533.
- (Holst, 1995) Holst, G. C. (1995). “Solid-State Cameras”. In: *Handbook of Optics*. Ed. by M. Bass. 2nd ed. Vol. III: Classical Optics, Vision Optics, X-Ray Optics. McGraw-Hill. Chap. 4, pp. 4.2, 4.8.
- (Howard et al., 1999) Howard, R. T., Bryan, T. C., and Book, M. L. (1999). “On-orbit testing of the video guidance sensor”. In: *Laser Radar Technology and Applications IV*. Ed. by G. W. Kamerman and C. Werner. Vol. 3707. International Society for Optics and Photonics. SPIE, pp. 290–300. DOI: 10.1117/12.351352.
- (P. Huber, 1977) Huber, P. (Jan. 1977). “Robust Methods of Estimation of Regression Coefficients”. In: *Series Statistics* 8.1, pp. 41–53. DOI: 10.1080/02331887708801356.
- (P. J. Huber, 2009) Huber, P. J. (2009). “Robust Statistics”. In: 2nd ed. John Wiley & Sons, Inc., pp. 175–186. DOI: 10.1002/0471725250.
- (Ioffe and Szegedy, 2015) Ioffe, S. and Szegedy, C. (July 2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 448–456. URL: <http://proceedings.mlr.press/v37/loffe15.html>.
- (Irani and Anandan, 2000) Irani, M. and Anandan, P. (2000). “About Direct Methods”. In: *Vision Algorithms: Theory and Practice*. Ed. by B. Triggs, A. Zisserman, and R. Szeliski. Vol. 1883. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer-Verlag, pp. 267–277.
- (Jena-Optronik, 2015) Jena-Optronik (Apr. 2015). *Rendezvous- and Docking Sensor RVS Datasheet*. URL: <https://www.jena-optronik.de/products/rendezvous-sensors/rvs.html> (visited on 04/30/2021).
- (Johansson et al., 2016) Johansson, J., Solli, M., and Maki, A. (2016). “An Evaluation of Local Feature Detectors and Descriptors for Infrared Images”. In:

Computer Vision – ECCV 2016 Workshops. Ed. by G. Hua and H. Jégou. Springer International Publishing, pp. 711–723. DOI: 10.1007/978-3-319-49409-8_59.

(Johnson et al., 2008)

Johnson, N. L., Stansbery, E., Whitlock, D. O., Abercromby, K. J., and Shoots, D. (2008). *History of On-orbit Satellite Fragmentations*. Orbital Debris Program Office. Tech. rep. NASA/TM-2008-214779. Version 14. Houston, TX: National Aeronautics and Space Administration. URL: <http://orbitaldebris.jsc.nasa.gov/library/SatelliteFragHistory/TM-2008-214779.pdf> (visited on 08/30/2021).

(Junkins et al., 1999)

Junkins, J. L., Hughes, D. C., Wazni, K. P., and Pariyapong, V. (1999). “Vision-Based Navigation for Rendezvous, Docking and Proximity Operations”. In: *22nd Annual AAS Guidance and Control Conference*. Vol. 99. Breckenridge, CO, p. 21.

(Kanani et al., 2012)

Kanani, K., Petit, A., Marchand, E., Chabot, T., and Gerber, B. (2012). “Vision Based Navigation for Debris Removal Missions”. In: *63rd International Astronautical Congress*. Paper IAC-12,A6,5,9,x14900. Naples, Italy: International Astronautical Federation (IAF).

(Kanatani, 1996)

Kanatani, K. (1996). “Statistical Optimization for Geometric Computation: Theory and Practice”. In: New York, NY: Elsevier Science Inc., p. 67. DOI: 10.1016/s0923-0459(96)x8019-4.

(Karlsson et al., 2005)

Karlsson, N. et al. (2005). “The vSLAM Algorithm for Robust Localization and Mapping”. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 24–29. DOI: 10.1109/robot.2005.1570091.

(Kawano et al., 2001)

Kawano, I., Mokuno, M., Kasai, T., and Suzuki, T. (2001). “Result of Autonomous Rendezvous Docking Experiment of Engineering Test Satellite-VII”. In: *Journal of Spacecraft and Rockets* 38.1, pp. 105–111. DOI: 10.2514/2.3661.

(Kechagias-Stamatis et al., 2020)

Kechagias-Stamatis, O., Aouf, N., Dubanchet, V., and Richardson,

- M. A. (2020). “DeepLO: Multi Projection Deep LIDAR Odometry for Space Orbital Robotics Rendezvous Relative Navigation”. In: *Acta Astronautica*. Accepted for publication.
- (Kelsey et al., 2006)
- Kelsey, J., Byrne, J., Cosgrove, M., Seereeram, S., and Mehra, R. (2006). “Vision-Based Relative Pose Estimation for Autonomous Rendezvous And Docking”. In: *2006 IEEE Aerospace Conference*. IEEE. DOI: 10.1109/aero.2006.1655916.
- (Kendall and Cipolla, 2017)
- Kendall, A. and Cipolla, R. (2017). “Geometric Loss Functions for Camera Pose Regression with Deep Learning”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6555–6564. DOI: 10.1109/CVPR.2017.694.
- (Kendall, Gal, et al., 2018)
- Kendall, A., Gal, Y., and Cipolla, R. (2018). “Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491. DOI: 10.1109/CVPR.2018.00781.
- (Kendall, Grimes, et al., 2015)
- Kendall, A., Grimes, M., and Cipolla, R. (2015). “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2938–2946. DOI: 10.1109/ICCV.2015.336.
- (Kessler and Cour-Palais, 1978)
- Kessler, D. J. and Cour-Palais, B. G. (1978). “Collision Frequency of Artificial Satellites: The Creation of a Debris Belt”. In: *Journal of Geophysical Research: Space Physics* 83.A6, pp. 2637–2646. DOI: 10.1029/ja083ia06p02637.
- (Khan et al., 2018)
- Khan, A., Sohail, A., and Ali, A. (2018). *A New Channel Boosted Convolutional Neural Network using Transfer Learning*. arXiv: 1804.08528 [cs.CV].
- (Kingma and J. Ba, 2014)
- Kingma, D. P. and Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].

- (Kintner, 1976) Kintner, E. C. (1976). “On the Mathematical Properties of the Zernike Polynomials”. In: *Optica Acta: International Journal of Optics* 23.8, pp. 679–680.
- (Kisantal et al., 2020) Kisantal, M. et al. (2020). “Satellite Pose Estimation Challenge: Dataset, Competition Design and Results”. In: *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1. DOI: 10.1109/taes.2020.2989063.
- (Klein and D. Murray, 2007) Klein, G. and Murray, D. (Nov. 2007). “Parallel Tracking and Mapping for Small AR Workspaces”. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE. DOI: 10.1109/ismar.2007.4538852.
- (Kneip et al., 2014) Kneip, L., Li, H., and Seo, Y. (2014). “UPnP: An Optimal O(n) Solution to the Absolute Pose Problem with Universal Applicability”. In: *European Conference on Computer Vision – ECCV 2014*. Springer International Publishing, pp. 127–142. DOI: 10.1007/978-3-319-10590-1_9.
- (Kozlowski and Kosonocky, 1995) Kozlowski, L. J. . and Kosonocky, W. F. . (1995). “Infrared Detector Arrays”. In: *Handbook of Optics*. Ed. by M. Bass. 2nd ed. Vol. I: Fundamentals, Techniques, and Design. McGraw-Hill. Chap. 23, pp. 23.4–23.10.
- (Krizhevsky et al., 2012) Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25. Curran Associates, Inc., pp. 1097–1105.
- (Krueger et al., 2017) Krueger, D. et al. (2017). *Zoneout: Regularizing RNNs by Randomly Preserving Hidden Activations*. arXiv: 1606.01305 [cs.NE].
- (LeCun et al., 1989) LeCun, Y. et al. (Nov. 1989). “Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning”.

- In: *IEEE Communications Magazine* 27.11, pp. 41–46. DOI: 10.1109/35.41400.
- (Lee et al., 2014) Lee, J. H. et al. (May 2014). “Outdoor Place Recognition in Urban Environments using Straight Lines”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. DOI: 10.1109/icra.2014.6907675.
- (Leinz et al., 2008)
- Leinz, M. R. et al. (2008). “Modeling, simulation, testing, and verification of the Orbital Express Autonomous Rendezvous and Capture Sensor System (ARCSS)”. In: *Sensors and Systems for Space Applications II*. Ed. by R. T. Howard and P. Motaghedi. Vol. 6958. International Society for Optics and Photonics. SPIE, pp. 75–87. DOI: 10.1117/12.779599.
- (Lepetit and Fua, 2005)
- Lepetit, V. and Fua, P. (2005). “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey”. In: *Foundations and Trends in Computer Graphics and Vision* 1.1, pp. 1–89. DOI: 10.1561/0600000001.
- (Lepetit, Moreno-Noguer, et al., 2008)
- Lepetit, V., Moreno-Noguer, F., and Fua, P. (July 2008). “EPnP: An Accurate $O(n)$ Solution to the PnP Problem”. In: *International Journal of Computer Vision* 81.2, pp. 155–166. DOI: 10.1007/s11263-008-0152-6.
- (Leutenegger et al., 2011)
- Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). “BRISK: Binary Robust Invariant Scalable Keypoints”. In: *2011 International Conference on Computer Vision*. IEEE, pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542.
- (S. Li et al., 2009)
- Li, S., Lee, M.-C., and Pun, C.-M. (Jan. 2009). “Complex Zernike Moments Features for Shape-Based Image Retrieval”. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 39.1, pp. 227–237. DOI: 10.1109/tsmca.2008.2007988.
- (Liebelt et al., 2008)
- Liebelt, J., Schmid, C., and Schertler, K. (June 2008). “Viewpoint-independent object class detection using 3D Feature Maps”. In:

- 2008 *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. DOI: 10.1109/cvpr.2008.4587614.
- (Lindeberg, 1994) Lindeberg, T. (Jan. 1994). “Scale-Space Theory: A Basic Tool for Analyzing Structures at Different Scales”. In: *Journal of Applied Statistics* 21.1-2, pp. 225–270. DOI: 10.1080/757582976.
- (C. Liu and Hu, 2014)
Liu, C. and Hu, W. (2014). “Relative Pose Estimation for Cylinder-Shaped Spacecrafts using Single Image”. In: *IEEE Transactions on Aerospace and Electronic Systems* 50.4, pp. 3036–3056. DOI: 10.1109/taes.2014.120757.
- (López et al., 1999)
López, A. M., Lumbreras, F., Serrat, J., and Villanueva, J. J. (1999). “Evaluation of Methods for Ridge and Valley Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.4, pp. 327–335. DOI: 10.1109/34.761263.
- (Louet and Bruzzi, 1999)
Louet, J. and Bruzzi, S. (1999). “ENVISAT mission and system”. In: *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS’99*. Vol. 3, pp. 1680–1682. DOI: 10.1109/IGARSS.1999.772059.
- (Lowe, 1991)
Lowe, D. G. (May 1991). “Fitting Parameterized Three-Dimensional Models to Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.5, pp. 441–450. DOI: 10.1109/34.134043.
- (Lowe, 2004)
Lowe, D. G. (Nov. 2004). “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2, pp. 91–110. DOI: 10.1023/b:visi.0000029664.99615.94.
- (Lunghi et al., 2016)
Lunghi, P., Ciarambino, M., and Lavagna, M. (July 2016). “A multilayer perceptron hazard detector for vision-based autonomous planetary landing”. In: *Advances in Space Research* 58.1, pp. 131–144. DOI: 10.1016/j.asr.2016.04.012.
- (Maas et al., 2013)
Maas, A., Hannun, A., and Ng, A. (2013). “Rectifier Nonlinearities Improve Neural Network Acoustic Models”. In: *Proceedings of the International Conference on Machine Learning*. Atlanta, Georgia.

(Maimone et al., 2007)

Maimone, M., Cheng, Y., and Matthies, L. (2007). “Two Years of Visual Odometry on the Mars Exploration Rovers”. In: *Journal of Field Robotics* 24.3, pp. 169–186. DOI: 10.1002/rob.20184.

(Markley and Crassidis, 2014)

Markley, F. L. and Crassidis, J. L. (2014). *Fundamentals of Spacecraft Attitude Determination and Control*. 1st edition. Springer New York, pp. 31–37. DOI: 10.1007/978-1-4939-0802-8.

(Markovsky and Mahmoodi, 2009)

Markovsky, I. and Mahmoodi, S. (Jan. 2009). “Least-Squares Contour Alignment”. In: *IEEE Signal Processing Letters* 16.1, pp. 41–44. DOI: 10.1109/lsp.2008.2008588.

(Mastrodemos et al., 2005)

Mastrodemos, N., Kubitschek, D. G., and Synnott, S. P. (2005). “Autonomous Navigation for the Deep Impact Mission Encounter with Comet Tempel 1”. In: *Space Science Reviews* 117.1-2, pp. 95–121. DOI: 10.1007/s11214-005-3394-4.

(Maybeck, 1979)

Maybeck, P. S. (1979). “Stochastic Models, Estimation, and Control”. In: vol. 1. New York, NY: Academic Press, pp. 238–241. DOI: 10.1109/tsmc.1980.4308494.

(McKnight, 2010)

McKnight, D. (Sept. 2010). “Pay Me Now or Pay Me More Later: Start the Development of Active Orbital Debris Removal Now”. In: *Advanced Maui Optical and Space Surveillance Technologies Conference*. Ed. by S. Ryan. Provided by the SAO/NASA Astrophysics Data System, E63. URL: <https://ui.adsabs.harvard.edu/abs/2010amos.confE..63M>.

(Mei et al., 2009)

Mei, L., Sun, M., Carter, K. M., III, A. O. H., and Savarese, S. (2009). “Unsupervised Object Pose Classification from Short Video Sequences”. In: *Proceedings of the British Machine Vision Conference 2009*. Guildford, United Kingdom: British Machine Vision Association, pp. 89.1–89.12. DOI: 10.5244/c.23.89.

(Meseguer et al., 2014)

Meseguer, J., Pérez-Grande, I., Sanz-Andrés, A., and Alonso, G. (2014). “Thermal Systems”. In: *The International Handbook of Space Technology*. Ed. by M. Macdonald and V. Badescu. Berlin, Heidelberg: Springer Praxis Books. Chap. 13, pp. 380–382. DOI: 10.1007/978-3-642-41101-4_13.

(Micron Technology, 2006)

Micron Technology (2006). *MT9D131 1/3.2-Inch System-On-A-Chip (SOC) CMOS Digital Image Sensor Datasheet*. URL: <https://planetary.s3.amazonaws.com/assets/pdfs/CMOS%20Camera%20System.pdf> (visited on 04/30/2021).

(Mikolajczyk and Schmid, 2005)

Mikolajczyk, K. and Schmid, C. (Oct. 2005). “A Performance Evaluation of Local Descriptors”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10, pp. 1615–1630. DOI: 10.1109/tpami.2005.188.

(Mikolajczyk, Tuytelaars, et al., 2005)

Mikolajczyk, K., Tuytelaars, T., et al. (Oct. 2005). “A Comparison of Affine Region Detectors”. In: *International Journal of Computer Vision* 65.1-2, pp. 43–72. DOI: 10.1007/s11263-005-3848-x.

(Miksik and Mikolajczyk, 2012)

Miksik, O. and Mikolajczyk, K. (2012). “Evaluation of Local Detectors and Descriptors for Fast Feature Matching”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, pp. 2681–2684.

(Möller and Trumbore, 1997)

Möller, T. and Trumbore, B. (Jan. 1997). “Fast, Minimum Storage Ray-Triangle Intersection”. In: *Journal of Graphics Tools* 2.1, pp. 21–28. DOI: 10.1080/10867651.1997.10487468.

(Montemerlo et al., 2002)

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem”. In: *Eighteenth National Conference on Artificial Intelligence*. Edmonton, Alberta, Canada: American Association for Artificial Intelligence, pp. 593–598.

(Moravec, 1980)

Moravec, H. P. (1980). “Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover”. AAI8024717. PhD thesis. Stanford, CA, USA.

(Moreno-Noguer et al., 2007)

Moreno-Noguer, F., Lepetit, V., and Fua, P. (2007). “Accurate Non-Iterative O(n) Solution to the PnP Problem”. In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. DOI:

10.1109/iccv.2007.4409116. URL: <https://doi.org/10.1109%2Ficcv.2007.4409116>.

(Mouats et al., 2018)

Mouats, T., Aouf, N., Nam, D., and Vidas, S. (Feb. 2018). “Performance Evaluation of Feature Detectors and Descriptors Beyond the Visible”. In: *Journal of Intelligent & Robotic Systems* 92.1, pp. 33–63. DOI: 10.1007/s10846-017-0762-8.

(Muja and Lowe, 2009)

Muja, M. and Lowe, D. G. (2009). “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration”. In: *Proceedings of the 4th International Conference on Computer Vision Theory and Applications (VISAPP)*. SciTePress - Science. DOI: 10.5220/0001787803310340.

(Mur-Artal et al., 2015)

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (Oct. 2015). “ORB-SLAM: A Versatile and Accurate Monocular SLAM System”. In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163. DOI: 10.1109/tro.2015.2463671.

(Murphy, 2012)

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press, pp. 1–3, 9–10. DOI: 10.5555/2380985.

(R. M. Murray et al., 1994)

Murray, R. M., Sastry, S. S., and Zexiang, L. (1994). *A Mathematical Introduction to Robotic Manipulation*. 1st ed. Boca Raton, FL, USA: CRC Press, Inc., pp. 34–39, 41–42, 53–54. ISBN: 0849379814. DOI: 10.1201/9781315136370.

(Naasz et al., 2010)

Naasz, B., Van Eepoel, J., Queen, S., Southward, C. M., and Hannah, J. (Feb. 2010). “Flight Results of the HST SM4 Relative Navigation Sensor System”. In: *33rd Annual AAS Guidance and Control Conference* (Feb. 6, 2010). AAS 10-086. Breckenridge, CO. URL: <https://core.ac.uk/reader/10553280> (visited on 05/01/2021).

(Nister et al., 2004)

Nister, D., Naroditsky, O., and Bergen, J. (2004). “Visual Odometry”. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR*

2004. Vol. 1. 1063-6919/04, pp. I–I. DOI: 10.1109/CVPR.2004.1315094.
- (Oestreich et al., 2020) Oestreich, C., Lim, T. W., and Broussard, R. (Jan. 2020). “On-Orbit Relative Pose Initialization via Convolutional Neural Networks”. In: *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2020-0457.
- (Oumer, 2014) Oumer, N. W. (2014). “Monocular 3D Pose Tracking of a Specular Object”. In: *Proceedings of the 9th International Conference on Computer Vision Theory and Applications*. SCITEPRESS - Science. DOI: 10.5220/0004667304580465.
- (Ozuysal et al., 2009) Ozuysal, M., Lepetit, V., and Fua, P. (June 2009). “Pose Estimation for Category Specific Multiview Object Localization”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. New York, NY: IEEE, pp. 778–785. DOI: 10.1109/cvpr.2009.5206633.
- (Painter et al., 1994) Painter, J. R., Mair, G. M., and Clarkson, T. G. (1994). “Computer-Integrated Engineering Systems”. In: *Mechanical Engineer’s Reference Book*. Ed. by E. H. Smith. 12th ed. Oxford, UK: Butterworth-Heinemann. Chap. 5, pp. 5/26–5/27. DOI: 10.1016/B978-0-7506-1195-4.50009-9.
- (Peleg and Rom, 1990) Peleg, S. and Rom, H. (1990). “Motion based segmentation”. In: *10th International Conference on Pattern Recognition*. IEEE Comput. Soc. Press. DOI: 10.1109/icpr.1990.118074.
- (Petit et al., 2013) Petit, A., Marchand, E., and Kanani, K. (Nov. 2013). “A robust model-based tracker combining geometrical and color edge information”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. DOI: 10.1109/iros.2013.6696887.
- (Petit et al., 2014) Petit, A., Marchand, E., and Kanani, K. (May 2014). “Combining complementary edge, keypoint and color features in model-based tracking for highly dynamic scenes”. In: *2014 IEEE International*

- Conference on Robotics and Automation (ICRA)*. IEEE. DOI: 10.1109/icra.2014.6907457.
- (Post and J. Li, 2018)
Post, M. and Li, J. (Jan. 2018). “Visual Monocular 3D Reconstruction and Component Identification for Small Spacecraft”. In: Multidisciplinary Digital Publishing Institute (preprint). DOI: 10.20944/preprints201801.0195.v1.
- (Poujol et al., 2015)
Poujol, J. et al. (Dec. 2015). “A Visible-Thermal Fusion Based Monocular Visual Odometry”. In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, pp. 517–528. DOI: 10.1007/978-3-319-27146-0_40.
- (Proença and Gao, 2019)
Proença, P. F. and Gao, Y. (2019). *Deep Learning for Spacecraft Pose Estimation from Photorealistic Rendering*. arXiv: 1907.04298 [cs.CV].
- (Rad et al., 2018) Rad, M., Oberweger, M., and Lepetit, V. (June 2018). “Feature Mapping for Learning Fast and Accurate 3D Pose Inference from Synthetic Images”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. DOI: 10.1109/cvpr.2018.00490.
- (Redmon and Farhadi, 2017)
Redmon, J. and Farhadi, A. (July 2017). “YOLO9000: Better, Faster, Stronger”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2017.690.
- (Redmon and Farhadi, 2018)
Redmon, J. and Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv: 1804.02767 [cs.CV].
- (Reeves et al., 1988)
Reeves, A., Prokop, R., Andrews, S., and Kuhl, F. (1988). “Three-Dimensional Shape Analysis Using Moments and Fourier Descriptors”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.6, pp. 937–943. DOI: 10.1109/34.9115.
- (Ren et al., 2017) Ren, S., He, K., Girshick, R., and Sun, J. (June 2017). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and*

- Machine Intelligence* 39.6, pp. 1137–1149. DOI: 10.1109/tpami.2016.2577031.
- (Ricaurte et al., 2014)
Ricaurte, P., Chilán, C., Aguilera-Carrasco, C., Vintimilla, B., and Sappa, A. (Feb. 2014). “Feature Point Descriptors: Infrared and Visible Spectra”. In: *Sensors* 14.2, pp. 3690–3701. DOI: 10.3390/s140203690.
- (Rondao, 2016)
Rondao, D. (Apr. 2016). “Modeling and Simulation of the ECOSat-III Attitude Determination and Control System”. MA thesis. Lisbon, Portugal: Instituto Superior Técnico, University of Lisbon.
- (Rondao and Aouf, 2018)
Rondao, D. and Aouf, N. (Jan. 2018). “Multi-View Monocular Pose Estimation for Spacecraft Relative Navigation”. In: *2018 AIAA Guidance, Navigation, and Control Conference*. Kissimmee, FL: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2018-2100.
- (Rondao, Aouf, and Dubois-Matra, 2018)
Rondao, D., Aouf, N., and Dubois-Matra, O. (Oct. 2018). “Multi-spectral Image Processing for Navigation Using Low Performance Computing”. In: *69th International Astronautical Congress (IAC) 2018*. Bremen, Germany: IAF. URL: <https://dspace.lib.cranfield.ac.uk/handle/1826/13558>.
- (Rondao, Aouf, and M. A. Richardson, 2021)
Rondao, D., Aouf, N., and Richardson, M. A. (2021). “ChiNet: Deep Recurrent Convolutional Learning for Multimodal Spacecraft Pose Estimation”. In: *IEEE Transactions on Aerospace and Electronic Systems*. Manuscript in submission.
- (Rondao, Aouf, M. A. Richardson, and Dubanchet, 2021)
Rondao, D., Aouf, N., Richardson, M. A., and Dubanchet, V. (2021). “Robust On-Manifold Optimization for Uncooperative Space Relative Navigation with a Single Camera”. In: *Journal of Guidance, Control, and Dynamics*. Article in advance, pp. 1–26. DOI: 10.2514/1.G004794.
- (Rondao, Aouf, M. A. Richardson, and Dubois-Matra, 2020)
Rondao, D., Aouf, N., Richardson, M. A., and Dubois-Matra, O. (July 2020). “Benchmarking of local feature detectors and descriptors for multispectral relative navigation in space”. In:

- Acta Astronautica* 172, pp. 100–122. DOI: 10.1016/j.actaastro.2020.03.049.
- (Rosten and Drummond, 2006)
 Rosten, E. and Drummond, T. (2006). “Machine Learning for High-Speed Corner Detection”. In: *Computer Vision – ECCV 2006*. Springer Berlin Heidelberg, pp. 430–443. DOI: 10.1007/11744023_34.
- (Rousseeuw and Leroy, 1987)
 Rousseeuw, P. J. and Leroy, A. M. (Oct. 1987). “Robust Regression and Outlier Detection”. In: New York, NY: John Wiley & Sons, Inc., pp. 1–4, 12. DOI: 10.1002/0471725382.
- (Roux and da Cunha, 2004)
 Roux, Y. and da Cunha, P. (Oct. 2004). “The GNC Measurement System for the Automated Transfer Vehicle”. In: *18th International Symposium on Space Flight Dynamics (ISSFD)*. Jointly organised by the German Space Operations Center of DLR and the European Space Operations Centre of ESA. Munich, Germany, pp. 11–15.
- (Rublee et al., 2011)
 Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). “ORB: An Efficient Alternative to SIFT or SURF”. In: *2011 International Conference on Computer Vision*. IEEE, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.
- (Ruel et al., 2012) Ruel, S., Luu, T., and Berube, A. (2012). “Space shuttle testing of the TriDAR 3D rendezvous and docking sensor”. In: *Journal of Field Robotics* 29.4, pp. 535–553. DOI: <https://doi.org/10.1002/rob.20420>.
- (Rumelhart et al., 1985)
 Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (Sept. 1985). *Learning Internal Representations by Error Propagation*. Tech. rep. DOI: 10.21236/ada164453.
- (Rumelhart et al., 1986)
 Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (Oct. 1986). “Learning Representations by Back-Propagating Errors”. In: *Nature* 323.6088, pp. 533–536. DOI: 10.1038/323533a0.
- (Russell and Norvig, 2013)
 Russell, S. and Norvig, P. (2013). *Artificial Intelligence: A Modern*

- Approach*. 3rd ed. Pearson Education Limited, pp. 1–3. DOI: 10.5555/1671238.
- (Schmid et al., 2000)
Schmid, C., Mohr, R., and Bauckhage, C. (2000). “Evaluation of Interest Point Detectors”. In: *International Journal of Computer Vision* 37.2, pp. 151–172. DOI: 10.1023/A:1008199403446.
- (Schönemann, 1966)
Schönemann, P. H. (Mar. 1966). “A generalized solution of the orthogonal procrustes problem”. In: *Psychometrika* 31.1, pp. 1–10. DOI: 10.1007/bf02289451.
- (Selig, 2004)
Selig, J. M. (2004). *Geometric Fundamentals of Robotics*. 1st ed. New York, NY: Springer Science & Business Media, pp. 54–57. DOI: 10.1007/b138859.
- (Sharma, Beierle, et al., 2018)
Sharma, S., Beierle, C., and D’Amico, S. (Mar. 2018). “Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Convolutional Neural Networks”. In: *2018 IEEE Aerospace Conference*. IEEE. DOI: 10.1109/aero.2018.8396425.
- (Sharma and D’Amico, 2019)
Sharma, S. and D’Amico, S. (2019). *Pose Estimation for Non-Cooperative Rendezvous Using Neural Networks*. arXiv: 1906.09868 [cs.CV].
- (Sharma, Ventura, et al., 2018)
Sharma, S., Ventura, J., and D’Amico, S. (Nov. 2018). “Robust Model-Based Monocular Pose Initialization for Noncooperative Spacecraft Rendezvous”. In: *Journal of Spacecraft and Rockets* 55.6, pp. 1414–1429. DOI: 10.2514/1.a34124.
- (J.-F. Shi, Ulrich, Ruel, and Anctil, 2015)
Shi, J.-F., Ulrich, S., Ruel, S., and Anctil, M. (Aug. 2015). “Uncooperative Spacecraft Pose Estimation Using an Infrared Camera During Proximity Operations”. In: *AIAA SPACE 2015 Conference and Exposition*. Paper AIAA 2015-4429. Pasadena, CA: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2015-4429.
- (J.-F. Shi and Ulrich, 2016)
Shi, J.-F. and Ulrich, S. (Aug. 2016). “SoftPOSIT Enhancements for Monocular Camera Spacecraft Pose Estimation”. In: *2016 21st*

- International Conference on Methods and Models in Automation and Robotics (MMAR)*. New York, NY: IEEE. DOI: 10.1109/mmar.2016.7575083.
- (J.-F. Shi, Ulrich, and Ruel, 2016)
Shi, J.-F., Ulrich, S., and Ruel, S. (2016). “Spacecraft Pose Estimation Using a Monocular Camera”. In: *67th International Astronautical Congress*. Paper IAC-16-C1.3.4. Guadalajara, Mexico: International Astronautical Federation (IAF).
- (J.-F. Shi, Ulrich, and Ruel, 2017)
Shi, J.-F., Ulrich, S., and Ruel, S. (Jan. 2017). “Spacecraft Pose Estimation using Principal Component Analysis and a Monocular Camera”. In: *AIAA Guidance, Navigation, and Control Conference*. Paper 2017-1034. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2017-1034.
- (J. Shi and Tomasi, 1994)
Shi, J. and Tomasi, C. (1994). “Good Features To Track”. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*. IEEE Comput. Soc. Press. DOI: 10.1109/cvpr.1994.323794.
- (Shuster, 1993)
Shuster, M. D. (Oct. 1993). “A Survey of Attitude Representations”. In: *Journal of the Astronautical Sciences* 41.4, pp. 439–517.
- (Silburt et al., 2018)
Silburt, A., Zhu, C., Ali-Dib, M., Menou, K., and Jackson, A. (2018). “DeepMoon: Convolutional Neural Network Trainer to Identify Moon Craters”. In: *Astrophysics Source Code Library*.
- (Smith, 2017)
Smith, L. N. (2017). “Cyclical Learning Rates for Training Neural Networks”. In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472. DOI: 10.1109/WACV.2017.58.
- (Spencer et al., 2021)
Spencer, D. A. et al. (2021). “The LightSail 2 solar sailing technology demonstration”. In: *Advances in Space Research* 67.9. Solar Sailing: Concepts, Technology, and Missions II, pp. 2878–2889. DOI: <https://doi.org/10.1016/j.asr.2020.06.029>.
- (Stanković and Falkowski, 2003)
Stanković, R. S. and Falkowski, B. J. (Jan. 2003). “The Haar

- Wavelet Transform: its Status and Achievements”. In: *Computers & Electrical Engineering* 29.1, pp. 25–44. DOI: 10.1016/s0045-7906(01)00011-8.
- (Stewart, 1999) Stewart, C. V. (Jan. 1999). “Robust Parameter Estimation in Computer Vision”. In: *SIAM Review* 41.3, pp. 513–537. DOI: 10.1137/s0036144598345802.
- (Stillwell, 2008) Stillwell, J. (2008). *Naive Lie Theory*. New York, NY: Springer, pp. 32–37, 82, 98. DOI: 10.1007/978-0-387-78214-0.
- (Strube et al., 2015) Strube, M. et al. (Jan. 2015). “Raven: An On-Orbit Relative Navigation Demonstration Using International Space Station Visiting Vehicles”. In: *Guidance, Navigation, and Control 2015*. Vol. 154. Advances in the Astronautical Sciences. American Astronautical Society. URL: <https://ntrs.nasa.gov/citations/20150002731> (visited on 05/01/2021).
- (Su et al., 2015) Su, H., Qi, C. R., Li, Y., and Guibas, L. J. (Dec. 2015). “Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. New York, NY: IEEE, pp. 2686–2694. DOI: 10.1109/iccv.2015.308.
- (C. Sun et al., 2017) Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (Oct. 2017). “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE. DOI: 10.1109/iccv.2017.97.
- (K. Sun et al., 2019) Sun, K., Xiao, B., Liu, D., and Wang, J. (June 2019). “Deep High-Resolution Representation Learning for Human Pose Estimation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2019.00584.
- (Synnott et al., 1986) Synnott, S., Donegan, A., Riedel, J., and Stuve, J. (1986). “Interplanetary optical navigation - Voyager Uranus encounter”. In: *Astrodynamics Conference*. Williamsburg, VA: AIAA. DOI: 10.2514/6.1986-2113.
- (Szegedy, W. Liu, et al., 2015) Szegedy, C., Liu, W., et al. (June 2015). “Going Deeper with

- Convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2015.7298594.
- (Szegedy, Vanhoucke, et al., 2016)
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (June 2016). “Rethinking the Inception Architecture for Computer Vision”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2016.308.
- (Szeliski, 2011)
Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. 1st edition. London, UK: Springer-Verlag, pp. 45–49, 65–68, 184, 202, 337–338. ISBN: 1848829345, 9781848829343. DOI: 10.1007/978-1-84882-935-0.
- (Takeishi et al., 2015)
Takeishi, N. et al. (2015). “Evaluation of Interest-region Detectors and Descriptors for Automatic Landmark Tracking on Asteroids”. In: *Transactions of the Japan Society for Aeronautical and Space Sciences* 58.1, pp. 45–53. DOI: 10.2322/tjsass.58.45.
- (Taketomi et al., 2017)
Taketomi, T., Uchiyama, H., and Ikeda, S. (June 2017). “Visual SLAM algorithms: a survey from 2010 to 2016”. In: *IPSSJ Transactions on Computer Vision and Applications* 9.1. DOI: 10.1186/s41074-017-0027-2.
- (Tang et al., 2008)
Tang, J., Chen, W., and Wang, J. (2008). “A Study on the P3P Problem”. In: *Advanced Intelligent Computing Theories and Applications: With Aspects of Theoretical and Methodological Issues*. Ed. by D.-S. Huang, D. C. Wunsch, D. S. Levine, and K.-H. Jo. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 422–429. DOI: 10.1007/978-3-540-87442-3_53, .
- (Tateno et al., 2017)
Tateno, K., Tombari, F., Laina, I., and Navab, N. (July 2017). “CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr.2017.695.

(Thomas et al., 2006)

Thomas, A. et al. (2006). “Towards Multi-View Object Class Detection”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*. IEEE. DOI: 10.1109/cvpr.2006.311.

(Thrun et al., 2005)

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. Cambridge, MA and London, UK: The MIT Press.

(Tietz and T. E. Richardson, 1983)

Tietz, J. C. and Richardson, T. E. (June 1983). *Development of an Autonomous Video Rendezvous and Docking System*. Phase 2. Tech. rep. MCR-83-584/NASA-CR-170794. Martin Marietta Aerospace. URL: <https://ntrs.nasa.gov/citations/19830019794> (visited on 05/01/2021).

(Torr and Zisserman, 2000)

Torr, P. and Zisserman, A. (2000). “Feature Based Methods for Structure and Motion Estimation”. In: *Vision Algorithms: Theory and Practice*. Ed. by B. Triggs, A. Zisserman, and R. Szeliski. Vol. 1883. Lecture Notes in Computer Science. Springer-Verlag, pp. 278–295.

(Tredwell, 1995)

Tredwell, T. J. (1995). “Visible Array Detectors”. In: *Handbook of Optics*. Ed. by M. Bass. 2nd ed. Vol. I: Fundamentals, Techniques, and Design. McGraw-Hill. Chap. 22, p. 22.2.

(Triggs et al., 2000)

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). “Bundle Adjustment — A Modern Synthesis”. In: *Vision Algorithms: Theory and Practice*. Springer Berlin Heidelberg, pp. 298–372. DOI: 10.1007/3-540-44480-7_21.

(Tulsiani and Malik, 2015)

Tulsiani, S. and Malik, J. (June 2015). “Viewpoints and Keypoints”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York, NY: IEEE, pp. 1510–1519. DOI: 10.1109/cvpr.2015.7298758.

(Turing, 1950)

Turing, A. (1950). “Computing Machinery and Intelligence”. In: *Mind* 59.236, pp. 433–460.

(Vacchetti et al., 2003)

Vacchetti, L., Lepetit, V., and Fua, P. (2003). “Fusing Online and

- Offline Information for Stable 3D Tracking in Real-Time”. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* IEEE Comput. Soc, pp. II–241. DOI: 10.1109/cvpr.2003.1211476.
- (Vacchetti et al., 2004) Vacchetti, L., Lepetit, V., and Fua, P. (Oct. 2004). “Stable Real-Time 3D Tracking Using Online and Offline Information”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.10, pp. 1385–1391. DOI: 10.1109/tpami.2004.92.
- (Wagner, 2016) Wagner, L. (Feb. 2016). *Jena-Optronik Relative Navigation Sensor Activities*. ESA Asteroid Impact Mission Industry Days. Noordwijk, The Netherlands. URL: https://indico.esa.int/event/133/contributions/743/attachments/834/1011/06_Navigation_camera_Lidars.pdf (visited on 04/30/2021).
- (Wahba, 1965) Wahba, G. (July 1965). “A Least Squares Estimate of Satellite Attitude”. In: *SIAM Review* 7.3, pp. 409–409. DOI: 10.1137/1007077.
- (S. Wang et al., 2017) Wang, S., Clark, R., Wen, H., and Trigoni, N. (May 2017). “DeepVO: Towards End-to-end Visual Odometry with Deep Recurrent Convolutional Neural Networks”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. DOI: 10.1109/icra.2017.7989236.
- (Z. Wang et al., 2011) Wang, Z., Fan, B., and Wu, F. (2011). “Local Intensity Order Pattern for Feature Description”. In: *2011 International Conference on Computer Vision*. IEEE, pp. 603–610. DOI: 10.1109/iccv.2011.6126294.
- (Wertz, 1999) Wertz, J. R. (1999). “Guidance and Navigation”. In: *Space Mission Analysis and Design*. Ed. by J. R. Wertz and W. J. Larson. 3rd edition. Space Technology Series. El Segundo, CA: Microcosm Press. Chap. 11, pp. 497, 498.
- (Wertz and Bell, 2003) Wertz, J. R. and Bell, R. (Aug. 2003). “Autonomous Rendezvous and Docking Technologies — Status and Prospects”. In: *Space Systems Technology and Operations*. AeroSense 2003. Ed. by J.

- Peter Tchoryk and J. Shoemaker. Vol. 5088. Orlando, FL: SPIE. DOI: 10.1117/12.498121.
- (Wertz, 2001) Wertz, J. R. (2001). *Mission Geometry: Orbit & Constellation Design & Management*. 1st edition. Space Technology Library. With contributions by Meissinger, H. F., Newman, L. K., Smit, G. N. Hawthorne, CA and New York, NY: Microcosm Press and Springer, pp. 38–52, 507–508.
- (Widger and Woodall, 1976)
- Widger, W. K. and Woodall, M. P. (1976). “Integration of the Planck Blackbody Radiation Function”. In: *Bulletin of the American Meteorological Society* 57.10, pp. 1217–1219. DOI: 10.1175/1520-0477(1976)057<1217: IOTPBR>2.0.CO;2.
- (Wie et al., 2014) Wie, B., Lappas, V., and Gil-Fernández, J. (2014). “Attitude and Orbit Control Systems”. In: *The International Handbook of Space Technology*. Ed. by M. Macdonald and V. Badescu. Berlin, Heidelberg: Springer Praxis Books. Chap. 12, pp. 338–344, 352, 361–363, 365. DOI: 10.1007/978-3-642-41101-4.
- (Wieszok et al., 2017)
- Wieszok, Z., Aouf, N., Kechagias-Stamatis, O., and Chermak, L. (May 2017). “Stixel Based Scene Understanding for Autonomous Vehicles”. In: *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE. DOI: 10.1109/icnsc.2017.8000065.
- (Williams et al., 2007)
- Williams, B., Klein, G., and Reid, I. (2007). “Real-Time SLAM Relocalisation”. In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. DOI: 10.1109/iccv.2007.4409115.
- (Yilmaz, Aouf, Majewski, et al., 2017)
- Yilmaz, Ö., Aouf, N., Majewski, L., Sanchez-Gestido, M., and Ortega, G. (2017). “Using Infrared Based Relative Navigation for Active Debris Removal”. In: *10th International ESA Conference on Guidance, Navigation and Control Systems*. Salzburg, Austria: ESA, pp. 1–16.
- (Yilmaz, 2017) Yilmaz, Ö. (2017). Private Communication. Provision of Envisat satellite mock-up steady-state temperature and emissivity profiles acquired under testing campaign for the purpose of generating

- a synthetic image-based dataset. Shrivenham, United Kingdom: Cranfield University.
- (Yılmaz, 2018) Yılmaz, Ö. (2018). “Infrared Based Monocular Relative Navigation for Active Debris Removal”. PhD thesis. Shrivenham, United Kingdom: Cranfield University.
- (Yılmaz, Aouf, Checa, et al., 2017) Yılmaz, Ö., Aouf, N., Checa, E., Majewski, L., and Sanchez-Gestido, M. (Nov. 2017). “Thermal Analysis of Space Debris for Infrared-Based Active Debris Removal”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 233.3, pp. 811–822. DOI: 10.1177/0954410017740917.
- (Yun et al., 2019) Yun, K. et al. (2019). “Improved visible to IR image transformation using synthetic data augmentation with cycle-consistent adversarial networks”. In: *Pattern Recognition and Tracking XXX*. Ed. by M. S. Alam. Vol. 10995. International Society for Optics and Photonics. SPIE, pp. 1–8. DOI: 10.1117/12.2519121.
- (Zeiler and Fergus, 2013) Zeiler, M. D. and Fergus, R. (2013). *Visualizing and Understanding Convolutional Networks*. arXiv: 1311.2901 [cs.CV].
- (G. Zhang et al., 2016) Zhang, G., Kontitsis, M., Filipe, N., Tsiotras, P., and Vela, P. A. (2016). “Cooperative Relative Navigation for Space Rendezvous and Proximity Operations using Controlled Active Vision”. In: *Journal of Field Robotics* 33.2, pp. 205–228. DOI: 10.1002/rob.21575.
- (J. Zhang et al., 2006) Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2006). “Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study”. In: *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’06)*. IEEE. DOI: 10.1109/cvprw.2006.121.
- (L. Zhang and Koch, 2013) Zhang, L. and Koch, R. (2013). “An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency”. In: *Journal of Visual Communication and Image Representation* 24.7, pp. 794–805. ISSN: 1047-3203. DOI: <https://doi.org/10.1016/j.jvcir.2013.05.006>.

(Y. Zhang et al., 2015)

Zhang, Y., Liu, H., and Shang, Y. (Oct. 2015). “3D Model-based Detection and Tracking for Space Autonomous and Uncooperative Rendezvous”. In: *AOPC 2015: Optical Design and Manufacturing Technologies*. Ed. by L. Li, L. Zheng, and K. P. Thompson. Bellingham, WA: SPIE. DOI: 10.1117/12.2224964.

(Z. Zhang, 1997)

Zhang, Z. (Jan. 1997). “Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting”. In: *Image and Vision Computing* 15.1, pp. 59–76. DOI: 10.1016/S0262-8856(96)01112-2.

(Zhou et al., 2020)

Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. (2020). *On the Continuity of Rotation Representations in Neural Networks*. arXiv: 1812.07035 [cs.LG].

(Zhu et al., 2017)

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.

(Zou et al., 2016)

Zou, Y., Wang, X., Zhang, T., and Song, J. (June 2016). “Combining Point and Edge for Satellite Pose Tracking Under Illumination Varying”. In: *2016 12th World Congress on Intelligent Control and Automation (WCICA)*. IEEE. DOI: 10.1109/wcica.2016.7578814.