A New Passive 3-D Automatic Target Recognition Architecture for Aerial Platforms

Odysseas Kechagias-Stamatis and Nabil Aouf

***Abstract*— The 3-D automatic target recognition (ATR) has many advantages over its 2-D counterpart, but there are several constraints in the context of small low-cost unmanned aerial vehicles (UAVs). These limitations include the requirement for active rather than passive monitoring, high equipment costs, sensor packaging size, and processing burden. We, therefore, propose a new structure from motion (SfM) 3-D ATR architecture that exploits the UAV's onboard sensors, i.e., the visual band camera, gyroscope, and accelerometer, and meets the requirements of a small UAV system. We tested the proposed 3-D SfM ATR using simulated UAV reconnaissance scenarios and found that the performance was better than classic 3-D light detection and ranging (LIDAR) ATR, combining the advantages of 3-D LIDAR ATR and passive 2-D ATR. The main advantages of the proposed architecture include the rapid processing, target pose invariance, small template size, passive scene sensing, and inexpensive equipment. We implemented the SfM module under two keypoint detection, description and matching schemes, with the 3-D ATR module exploiting several current techniques. By comparing SfM 3-D ATR, 3-D LIDAR ATR, and 2-D ATR, we confirmed the superior performance of our new architecture.**

***Index Terms*— 3-D automatic target recognition (ATR), passive target recognition, structure from motion (SfM), unmanned aerial vehicles (UAVs).**

I. Introduction

AUTOMATIC target recognition (ATR) is an active research field for military applications because it can enhance the quality of reconnaissance in a hostile environment. Current research involves both 2-D and 3-D data, including solutions based on 2-D infrared (IR) [1], [2], 2-D synthetic aperture radar (SAR) [3], [4] or inverse SAR [5], 2-D hyperspectral imagery [6], and 3-D light detection and ranging (LIDAR) [7]–[11], the latter also including laser-induced fluorescence spectroscopy [12]. The military applications of ATR in several data domains have been reviewed [13].

LIDAR-based 3-D target recognition is superior to its 2-D counterpart because 3-D encoding can exploit the geometric properties and underlying structure of an object, offering more information than 2-D encoding. Indeed, features extracted from the 3-D domain are affected to a lesser extent by illumination variation and target pose changes [9], [14] and they can operate well in the context of a single 3-D model template [10], [11]. Despite these advantages, ATR based on 3-D LIDAR also has several drawbacks when used with small, low cost, time-critical unmanned aerial vehicles (UAVs) such as the RQ-11 Raven, including the disproportionate hardware cost of a LIDAR device, its large size and power requirements, the low data acquisition rate, and most importantly, the computational resources required to manipulate 3-D data. For military applications, LIDAR is an active device which, therefore, reveals the UAV's location. In contrast, the advantages of 2-D ATR include the small and inexpensive equipment, short processing times, and limited power requirements.

Here, we propose an architecture that combines the advantages of 3-D and 2-D ATR by exploiting a structure from motion (SfM) 3-D reconstruction concept that relies on a single visual band camera placed on a flying UAV platform. This is important because we demonstrate that SfM 3-D ATR preserves the capabilities of 3-D ATR, such as pose and illumination invariance, revealing the underlying structure of the target and relying on a single template. But SfM 3-D ATR also retains the benefits of 2-D ATR, such as the low processing burden, inexpensive hardware (camera *vs* LIDAR), faster data acquisition rate, and passive monitoring, the latter rendering it undetectable (Table I).

In the context of SfM-based 3-D ATR, this paper suggests a semantic SfM has been proposed, which simultaneously considers the geometric and semantic cues provided by 2-D images [15]. However, the processing burden is 20 min per scene, making it unsuitable for UAV applications that require near-real-time processing. Brostow *et al.* [16] have demonstrated the capabilities of object recognition using an SfM point cloud, albeit with simple objects involving nonreal-time 3-D reconstruction. Liebe *et al.* [17] propose SfM object recognition based on 2-D rather than 3-D data, thus preserving the constraints of 2-D ATR [17]. The usefulness of SfM has been demonstrated in military applications but only preliminary aspects of ATR were addressed [18]. Indeed, the applications of SfM have largely focused on slow-moving ground platforms rather than ATR [19], although one exceptional case (not extended to ATR) involved drone navigation [20]. Ultimately, SfM-based 3-D ATR has not yet received sufficient attention, a challenge we address by proposing an innovative architecture.

The rest of this paper is organized as follows. Section II of this paper introduces the SfM 3-D ATR architecture, and then Section III evaluates our method by testing it against highly credible simulated scenarios, challenging a number of current 3-D ATR descriptors. The contents of this paper are summarized in Section IV.

Manuscript received December 16, 2017; revised March 24, 2018 and May 27, 2018; accepted July 8, 2018. *(Corresponding author: Odysseas Kechagias-Stamatis.)*

The authors are with the Signals and Autonomy Group, Centre for Electronic Warfare Information and Cyber, Cranfield University Defence and Security, Shrivenham, SN6 8LA, U.K. (e-mail: o.kechagiasstamatis@cranfield.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2855065

TABLE I
ANALYSIS OF DATA DOMAIN DRIVEN ATR SOLUTIONS

	3D LIDAR	3D SfM	2D
Penetration of sparse structures	+	-	-
Template size	+ (can use one 3D model)	+ (can use one 3D model)	- (multiple views)
Target pose invariance	+	+	-
Target illumination invariance	+	+	-
ATR based on underlying structure	+	+	-
ATR based on texture	+ (during keypoint description)	+ (during keypoint description)	+
Operating day and night	+	-	-
Processing time	high	low	low
Equipment cost	-	+	+
Equipment size	medium / large	very small	very small
Power consumption	medium	very small	very small
Data acquiring rate	- (scanning LIDAR) + (flash LIDAR)	+	+
Maximum operating range	≈100m	>100m	>100m
Reveal sensor position	Yes (active)	No (passive)	No (passive)

II. SfM 3-D ATR Architecture

The proposed ATR architecture extends a previously suggested pipeline [10] to generate and utilize a 3-D SfM-based point cloud. The architecture comprises offline and online phases.

A. Offline Phase

During the offline phase, we use the hidden point removal algorithm [21] to simulate an aerial view $\boldsymbol{P_m}$ of the target's computer-aided design (CAD) model as a template. $\boldsymbol{P_m}$ is then uniformly subsampled at 0.3-m resolution and described using one of the 3-D descriptors presented in Section III-B.

B. Online Phase

This phase comprises the SfM module, which aims to create a 3-D reconstruction of the scene that can be input into the online part of the 3-D ATR architecture.

1) SfM Module: We propose an SfM module that exploits the gyroscope, accelerometer, and visual band (RGB) camera sensor of a flying UAV platform.

Given two 2-D scene images I_1, I_2 of size $m \times n$, acquired by the same camera positioned on a flying UAV at instances t and $t+1$, we perform keypoint detection and tracking on I_1 and I_2. Specifically, we detect and describe keypoints $p_a^{I_1}$, i.e., image pixels that are prominent among their surroundings in image I_1, by applying the good features to track (GFTT) algorithm [22] with a minimum corner quality of 10^{-3} and a 3×3 Gaussian filter. Then we use the Kanade–Lucas–Tomashi (KLT) tracker [23] to track these keypoints in I_2, but due to the camera's motion, only the subset $p_b^{I_1} b \leq a$ is tracked. For KLT, we use a forward-backward error [24] of one pixel, a 11×11 tracking window over 13 scales, and ten iterations. Finally $p_b^{I_1}$, $p_b^{I_2}$, and the camera's transformation matrix R_{cam} at instance $t+1$ in relation to t are input into a triangulation process to create the 3-D reconstruction of the matched keypoints $p_b^{I_1}$ and $p_b^{I_2}$.

In contrast to current SfM methods that calculate R_{cam} based on the I_1, I_2 image correspondences, we calculate the camera's 6-D real-world pose shift R_{cam}^* between instances t and $t+1$ by extracting the gyroscope and the accelerometer measurements R_{cam}^t and R_{cam}^{t+1} at both instances. Specifically, we calculate

$$R_{\text{cam}}^* = R_{\text{cam}}^{t+1} \cdot (R_{\text{cam}}^t)^{-1} \tag{1}$$

where, see (2)–(4), as shown at the bottom of this page, where R is the rotational and T the translational part of the transformation matrix R_{cam}^t; u, v, and w are the pitch, roll, and yaw, respectively; and a is the acceleration per axis on an XYZ reference frame set at the UAV's center of gravity. Fig. 1 shows an example of SfM 3-D reconstruction. For a detailed analysis of the standard SfM method, the reader is referred to [25].

We also perform SfM 3-D reconstruction by exploiting the speeded up robust features (SURF) [26] keypoint detection and description technique. We apply SURF on images I_1 and I_2 to extract keypoints $p_a^{I_1}$ and $p_a^{I_2}$. SURF is applied over six scale levels with a blob threshold of 10^{-3}. The features $f_a^{I_1}$ and $f_a^{I_2}$ of $p_a^{I_1}$ and $p_a^{I_2}$, respectively, are then matched based on the nearest neighbor distance ratio (NNDR) criterion [27] with a threshold empirically set at 0.6. The correspondences $p_b^{I_1}$ and $p_b^{I_2}$ undergo the same process as described for the GFTT/ KLT case.

$$R_{\text{cam}}^t == [R(u, v, w) | T(X, Y, Z)] \tag{2}$$

$$R(u, v, w) = \begin{pmatrix} \cos u \cdot \cos v & \cos u \cdot \sin v \cdot \sin w - \sin u \cdot \cos w & cosu \cdot \sin v \cdot \cos w + \sin u \cdot \sin w \\ \sin u \cdot \cos v & \sin u \cdot \sin v \cdot \sin w + \cos u \cdot \cos w & \sin u \cdot \sin v \cdot \cos w - \cos u \cdot \sin w \\ -\sin v & \cos v \cdot \sin w & \cos v \cdot \cos w \end{pmatrix} \tag{3}$$

$$T(X, Y, Z) = \left(\int_t^{t+1} \int_t^{t+1} \Delta a_x, \int_t^{t+1} \int_t^{t+1} \Delta a_y, \int_t^{t+1} \int_t^{t+1} \Delta a_z \right)^T \tag{4}$$

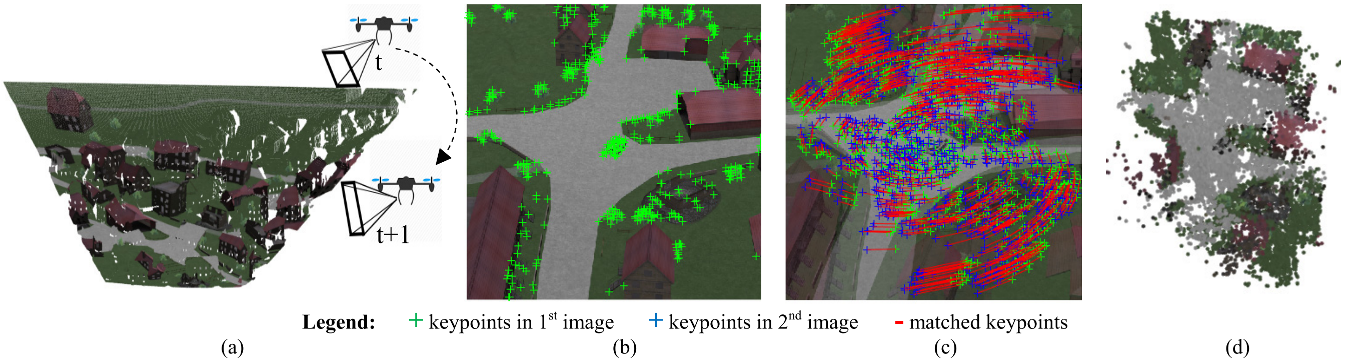

t
t+1
Legend: + keypoints in 1st image + keypoints in 2nd image - matched keypoints
(a)
(b)
(c)
(d)

Fig. 1. SfM-based concept for 3-D ATR applications on UAVs. (a) Image pair acquisition at UAV's position t and $t+1$. (b) Keypoint detection and description. (c) Keypoint correspondences. (d) 3-D reconstruction [number of keypoints detected and matched in (b) and (c) is reduced for better visualization].

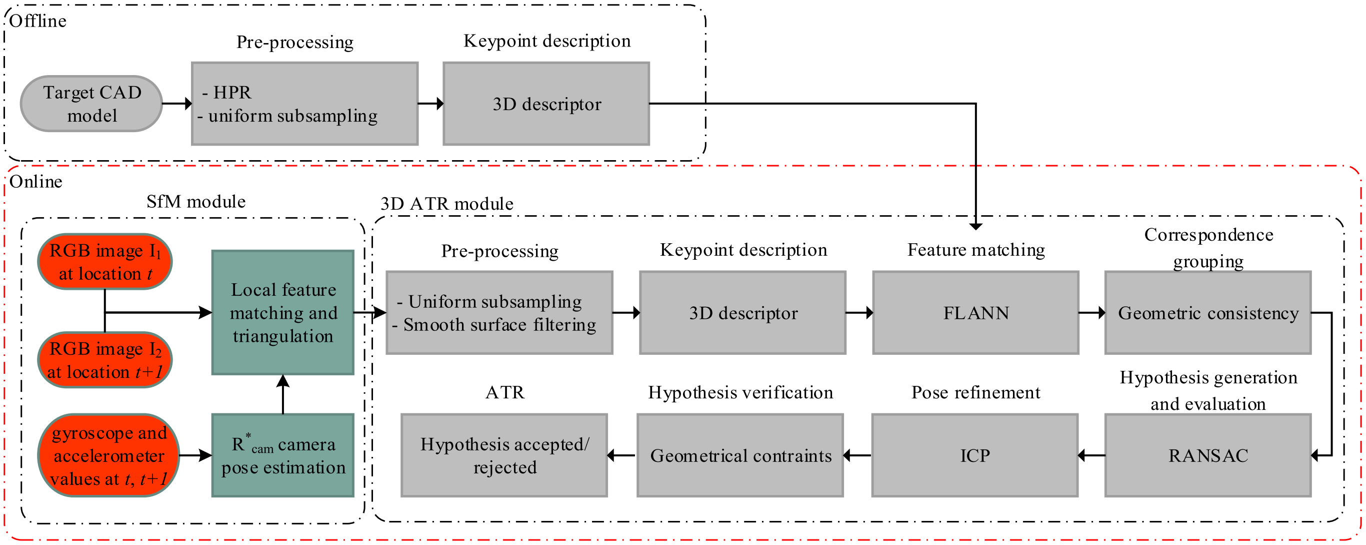


Fig. 2. SfM 3-D ATR architecture.

Despite the availability of several options to improve the accuracy of the point cloud reconstructed in 3-D by SfM methods, these were disregarded because computational efficiency is necessary for the UAV applications considered here. Although the UAV dynamics are already known from the gyroscope and accelerometer readings and can be incorporated into the SfM estimation via a Kalman filtering process to verify the matched keypoint correspondences, this imposes an additional processing burden, and is therefore omitted. Similarly, the resulting SfM point cloud is sparse, but the additional processing cost to make it dense substantially increases the processing time, and given that the performance of the ATR is already appealing (Section III), we did not attempt to create a dense point cloud. Superresolution [28] can improve 3-D reconstruction but the resulting computational burden was too great. Finally, we did not use multiple images to construct the point cloud, allowing us to investigate the limits of SfM for 3-D ATR applications.

2) 3-D ATR Module: During the online phase, the scene point cloud $\boldsymbol{P}$ is also uniformly subsampled at 0.3-m resolution. $\boldsymbol{P}$ is then refined to $\boldsymbol{P_f}$ by filtering its smooth surfaces based on the angular variation of the normal that is set on each vertex, compared to the normal of its surrounding vertices. Normal estimation considers fitting a plane on the six closest neighbors of the vertex for which we calculate the normal.

TABLE II

SCENARIO PARAMETERS

Scenario N°	1	2	3
N° of runs	4	4	1
Obliquity (°)	0°–45° per 15°	0°–45° per 15°	30°
UAV-target (m)	50	100	200
Resolution (cm)	11	18	30
Scenes with target/out of total	334/345	327/364	78/78

$\boldsymbol{P_f}$ is then described using the same 3-D descriptor as used for $\boldsymbol{P_m}$. Feature matching relies on a k-NNDR scheme where $k = 10$, whereas the main keypoint matching process involves the creation of groups of $\boldsymbol{P_f}- \boldsymbol{P_m}$ keypoint correspondences that are geometrically consistent. Each group $\{H_1, H_2, \ldots, H_g\}$, with g indicating the number of groups, is input into a random sample and consensus algorithm using 1000 iterations to define a transformation hypothesis between the CAD model $\boldsymbol{P_m}$ and the scene $\boldsymbol{P_f}$. Then, each hypothesis is verified for correctness by applying it to $\boldsymbol{P_m}$ followed by alignment with $\boldsymbol{P_f}$ using an iterative closest point scheme. Finally, the geometrical accuracy of this hypothesis is validated if the aligned model and the scene have overlapping vertices that exceed a threshold. The proposed 3-D SfM ATR architecture is presented in Fig. 2.

TABLE III
3-D DESCRIPTORS USED

Descriptor	Descriptor Length	Implementation platform	Operating principle
SHOT	352	C++ (Matlab Exchange (MEX) wrapper)	Angular variations
USC	1980	C++ (MEX wrapper)	Accumulating points
HoD / HoD-S	240 / 40	MATLAB / MATLAB	L2-norm distances, HoD coarse and fine encryption, HoD-S coarse encryption
FPFH	33	C++ (MEX wrapper)	Angular variations
3DSC	1980	C++ (MEX wrapper)	Accumulating points
RoPS	135	MATLAB	Low order statistics

III. EXPERIMENTS

A. Data Set

Real military data sets are restricted and we, therefore, used OpenFlight [29] to simulate three highly credible air-to-ground UAV reconnaissance scenarios (Table II). All scenarios considered the UAV flying a circular orbit at several UAV—target ranges, altitudes and headings, and under various pitch, and roll and yaw angles. Each scenario involved a T-72 main battle tank (MBT) in an urban environment that included clutter (nontarget objects) such as buildings and trees. Depending on the UAV's flight parameters, the T-72 target might be partially or even completely occluded by clutter. Notably, our scenarios simulated not only the size of the target, which depends on the UAV–target range, but also for the LIDAR case they also considered the laser spot size and how this affects the LIDAR point cloud. In contrast to [7], [30], and [31], our military scenarios were affected by more parameters and are therefore more challenging and realistic. For each scene, we generated a 3-D LIDAR point cloud and the corresponding 2-D visual image. Camera intrinsic and extrinsic parameters are the ones used while creating the scenarios.

B. Experimental Setup

We evaluated the effectiveness of the new SfM-based ATR using a multilevel scheme, i.e., challenging the effectiveness of several current 3-D ATR descriptors on SfM point clouds compared to LIDAR point clouds as well as classic 2-D ATR methods based on local features.

Specifically, for the SfM-based 3-D ATR, we exploited the ATR pipeline presented in Fig. 2, but for the LIDAR 3-D ATR we replaced the SfM module with the LIDAR-based point cloud. In both cases, we evaluated the following descriptors: signature of histograms of orientations (SHOT) [32], rotational projections statistics (RoPS) [33], fast point feature histograms (FPFH) [34], 3-D shape context (3-DSC) [35], unique shape context (USC) [36], histogram of distances (HoD) [37], and HoD-short (HoD-S) [10]. The description radius of each 3-D descriptor was $\rho \cdot r$, where r is the average point cloud resolution of the CAD model [32], [33], [38] and ρ a multiplier as suggested by the authors of each descriptor (e.g., for HoD and HoD-S, r is the scene resolution [37]). Table III presents each 3-D descriptor and its parameters, which were fixed either to those originally proposed by their authors or to their point cloud library implementation [37], [39]. Given that each 3-D descriptor was applied on a spherical volume $\boldsymbol{V}$ of radius ρ centered at a keypoint p, the operating principle of each 3-D descriptor can be summarized as follows.

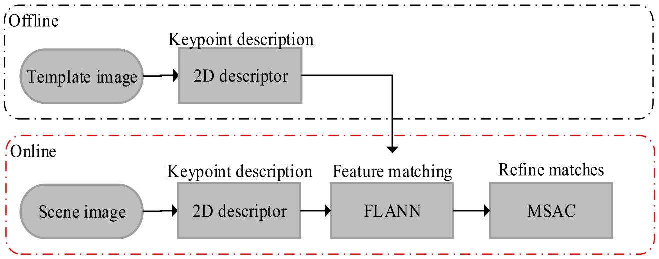


Fig. 3. 2-D ATR architecture used for comparative purposes.

1) SHOT [32] establishes a local reference frame (LRF) on p and divides $\boldsymbol{V}$ into a number of subvolumes along the azimuth, the elevation, and the radius. For each subvolume, SHOT encodes the normal variation among p (including its neighboring vertices) with the normal of each subvolume.
2) RoPS [33] establishes an LRF on p, then $\boldsymbol{V}$ is rotated around each axis of the LRF's coordinate frame and is finally projected on each of the coordinate planes. RoPS encryption involves a low-order moment and entropy description of each projection, and these are concatenated to formulate a histogram.
3) FPFH [34] establishes an LRF on p, and for each vertex belonging to $\boldsymbol{V}$, FPFH encodes the angular relationship between p and its neighbors as provided by the LRF. Finally, that angular relationship is transformed into a histogram.
4) For 3-DSC [35], a local reference axis (LRA) is established on p, aligned to the normal produced by the vertices in $\boldsymbol{V}$, and $\boldsymbol{V}$ is divided into a number of subvolumes along the azimuth, elevation, and radial dimension. The 3-DSC descriptor is established by accumulating a weighted sum of the points within each subvolume. Weights are proportional to the subvolume to center-of-V distance. The 3-DSC is LRA based and compensates for 360° azimuthal rotation by describing $\boldsymbol{V}$ in multiple azimuthal orientations. USC [36] is identical to 3-DSC but the LRA is replaced with an LRF.
5) HoD [37] calculates the point-pair L2-norm distance distributions of the vertices within $\boldsymbol{V}$. L2-distances are encoded in a coarse and a fine manner. HoD-S [10] involves only the coarse component of HoD.

In addition to the 3-D SfM versus 3-D LIDAR comparison, we also compared 3-D SfM against classic 2-D local feature ATR. For that purpose, we used the pipeline presented in Fig. 3 with the 2-D keypoint descriptors and detectors,

TABLE IV
2-D KEYPOINT DETECTION AND DESCRIPTION COMBINATION USED

ID	Keypoint detector	Keypoint descriptor	Descriptor length	Implementation platform	Tuned parameters
#1	GFTT	Fast Retina Keypoint (FREAK)	64	C++ (MEX wrapper)	Min corner quality 10^{-3} Gaussian filter size 3x3
#2	SURF	SURF	64	C++ (MEX wrapper)	Scale levels 6
#3	Features from Accelerated Segment Test (FAST)	Binary Robust Invariant Scalable Keypoints (BRISK)	64	C++ (MEX wrapper)	Min contrast 10^{-3}
#4	FAST	FREAK	64	C++ (MEX wrapper)	Min corner quality 10^{-3} and Min contrast 10^{-3}

as shown in Table IV. Table IV also presents the parameters used for each keypoint detector and descriptor combination to maximize its ATR performance.

The parameters of each remaining combination were fixed to those originally proposed by the author. Feature matching was based on the NNDR criterion [27] with a threshold of 0.8 and the M-estimator sample consensus algorithm [40] was used to refine the correspondences.

C. Performance Metric

ATR performance was evaluated using the F1 score [10]

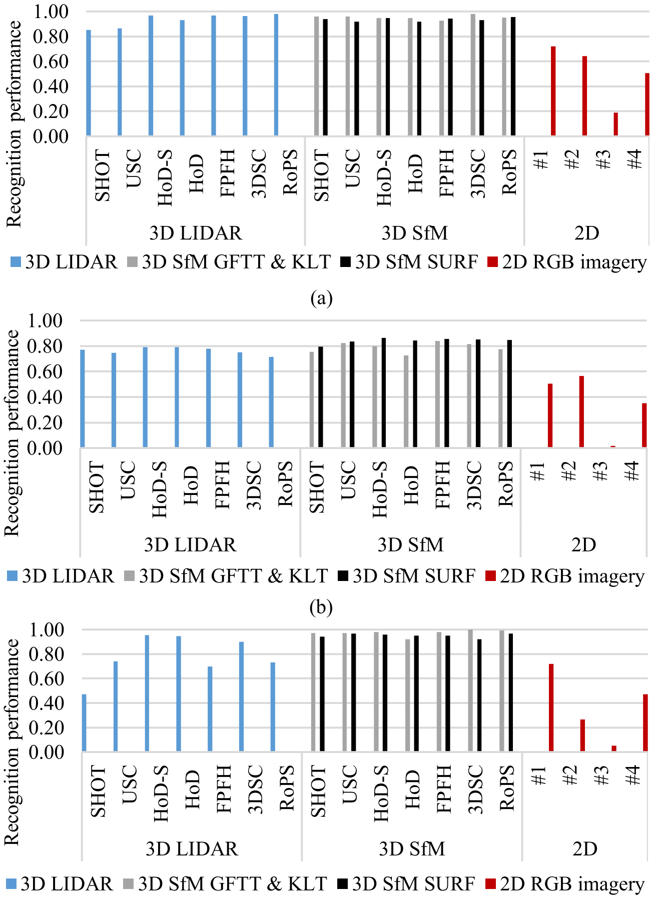
$$F1 - \text{score} = \frac{2\#\text{TP}}{2\#\text{TP} + \#\text{FP} + \#\text{FN}} \quad (5)$$

where # denotes the number of the metric that follows, i.e., true positive (TP), false positive (FP), and false negative (FN). We selected the F1-score metric because it encapsulates the classic precision and recall metrics without involving the true negative (TN) metric. This is important because in a number of runs per scenario the target is always present, i.e., TN = 0, and thus recall $= \#\text{TP} \cdot (\#\text{TP} + \#\text{TN})^{-1}$ would be biased.

As previously reported [10], these metrics not only compare the ATR prediction state with the actual state but also consider the Euclidean distance-based translational error T_{error} between the ground truth position of the target in the scene and its estimated final position. Hence, for a TP match the algorithm provides a transformation hypothesis for a scene where a target is present and $T_{\text{error}} < 2$ m. For an FP match, the algorithm provides a hypothesis for a scene that does not have a target or has a target with $T_{\text{error}} > 2$ m. This dual constraint, i.e., target presence in the scene and target localization accuracy (T_{error}), ensures that the FP match metric is not biased for scenarios in which the target is always present. Finally, the FN match case occurs if the algorithm does not provide a hypothesis for a scene that has a target. For fairness, T_{error} was also extended to facilitate the 2-D ATR scheme.

D. Assessment

We evaluated the ATR performance in terms of UAV–target range, obliquity variation, processing time, template storage, descriptor compactness, robustness to shot noise, and to Gaussian noise. The trials involved an UAV reconnaissance application for which we reduced the processing time of the 3-D ATR by exploiting a single CAD model, whereas for the 2-D ATR we minimized the number of templates as suggested [1]. Hence, we used 12 images of the target, evenly spaced across the 0°–360° azimuthal viewing angle, and these images were cropped from the first trial of the first scenario. It is worth noting that since the templates are cropped from the evaluation scenes, the performance of 2-D ATR is positively biased. To balance this, we only exploit image templates from a single scenario and run, while the experiments involve nine runs in total (Table II). Trials are implemented on an i7 at 2.6 GHz with 16-GB RAM.



(a)

(b)

(c)

Fig. 4. ATR for 3-D LIDAR versus 3-D SfM versus 2-D ATR in relation to UAV–target range. (a) Scenario 1—50 m. (b) Scenario 2—100 m. (c) Scenario 3—200 m.

1) UAV–Target Range Evaluation: In this trial, we compared the performance of the 3-D LIDAR, 3-D SfM, and 2-D ATR in relation to the UAV–target distance. Fig. 4(a) shows that the LIDAR and SfM 3-D ATR performed equally well at 50-m UAV–target range, and outperformed 2-D ATR because the SfM point cloud preferentially reconstructs the central region of the image close to the target, reducing misclassifications. We found that GFTT has a small performance advantage over SURF SfM that was consistent among all 3-D descriptors. Furthermore, GFTT outperformed the 2-D competitors, but was still inferior to both the LIDAR and SfM 3-D ATR techniques.

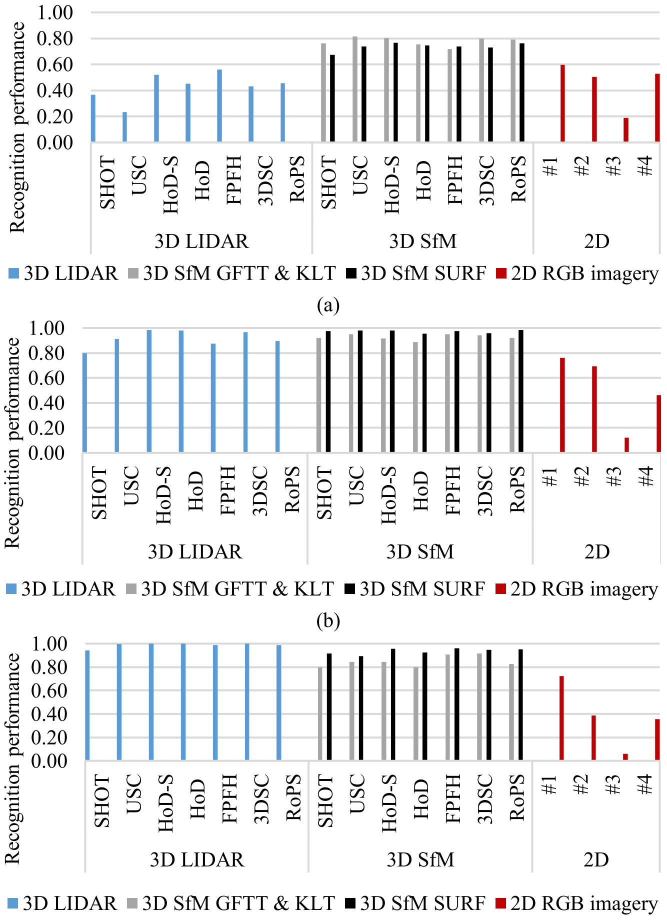
When the UAV–target distance increased to 100 m [Fig. 4(b)], the performance of all three solutions declined. When the distance increased to 200 m [Fig. 4(c)], the 3-D SfM achieved the best performance and the 2-D ATR the worst, as explained in more detail in the following.

SfM is created by matching 2-D keypoints from two images at the same range at 100 or 200 m. Hence, 2-D features are detected within the same scale and can be matched in sequential images for 3-D SfM reconstruction. Because the UAV flies a circular orbit, 3-D reconstruction is more accurate closer to the center of the orbit. In contrast, 2-D ATR encodes keypoints from a template presenting the MBT at a range of 50 m and aims to match these keypoints with those detected on an MBT at a different scale. Especially for the 200 m range, the MBT in the scene is four times further away than its template. That scale difference exceeds the scale invariance of all 2-D descriptors. In addition, templates are derived from 30° obliquity, whereas the angles are evaluated in the range 0°–45°, exceeding the out-of-plane invariance of the 2-D descriptors. Even though these are acknowledged as problems in 2-D ATR, we intentionally adopted a small template [1] to demonstrate the advantage of SfM 3-D ATR under a single-template scheme. Increasing the 2-D templates to accommodate several target poses and scales affects the computational and storage requirements, which are not always affordable, especially for time-critical applications. An analysis of the processing time and storage requirements is presented in Section III-D3.

Unsurprisingly, the performance of 3-D LIDAR ATR declined at a range of 200 m because the laser spot size increases as the beam propagates through the atmosphere, forcing the MBT in the scene to have simultaneously a smaller size and a lower resolution.

2) UAV-Target Obliquity Evaluation: This trial evaluated robustness in terms of obliquity variation but still considered the three UAV–target ranges. Even though the trials considered obliquity values of 0°–45° in increments of 15°, to improve clarity, we focus on the ATR performance for low, medium, and large obliquity angles of 0°, 30°, and 45°, respectively (Fig. 5).

For the low-angle test, 3-D SfM achieved the highest ATR performance by a large margin, with recognition rates of 81.5% for the GFTT with USC, and 76.7% for the SURF with HoD-S. The maximum performance of 3-D LIDAR was 56% with FPFH, whereas 2-D ATR achieved only 60% recognition. For the medium-angle test, 3-D SfM and 3-D LIDAR performed equally well at all three UAV–target ranges, achieving scores of 98% and 99%, respectively. Although 2-D ATR fared better than in the low-angle test, it was still inferior to the 3-D solutions, with a 76% recognition rate. The 3-D LIDAR ATR gained near-perfect scores in the high-angle test, and SURF SfM ATR was only mildly less successful, achieving a 96% recognition rate. Furthermore, GFTT SfM ATR and 2-D ATR achieved scores of 90.8% and 72.3%, respectively. The ATR performance attained is explained in the following.



(a)

(b)

(c)

Fig. 5. ATR for 3-D LIDAR versus 3-D SfM versus 2-D ATR in relation to target obliquity over all scenarios. (a) Low—15°. (b) Medium—30°. (c) High—45°.

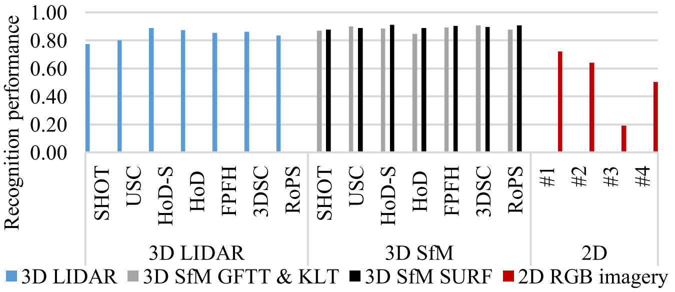


Fig. 6. Overall ATR performance of 3-D LIDAR versus 3-D SfM versus 2-D.

For the low-angle test, 3-D LIDAR suffered from a high rate of FP matches, leading to a low F1-score, because the LIDAR encapsulates a greater part of the scene. In contrast, in the context of 3-D SfM, the further away a keypoint is from the camera's optical axis, the larger its frame-to-frame motion. If this motion exceeds the one-pixel threshold, it is not reconstructed. Therefore, the 3-D SfM favors 3-D reconstruction near the camera optical axis and thus achieves a better performance than the 3-D LIDAR point cloud. Even if the two images used for SfM lack an MBT close to the camera's

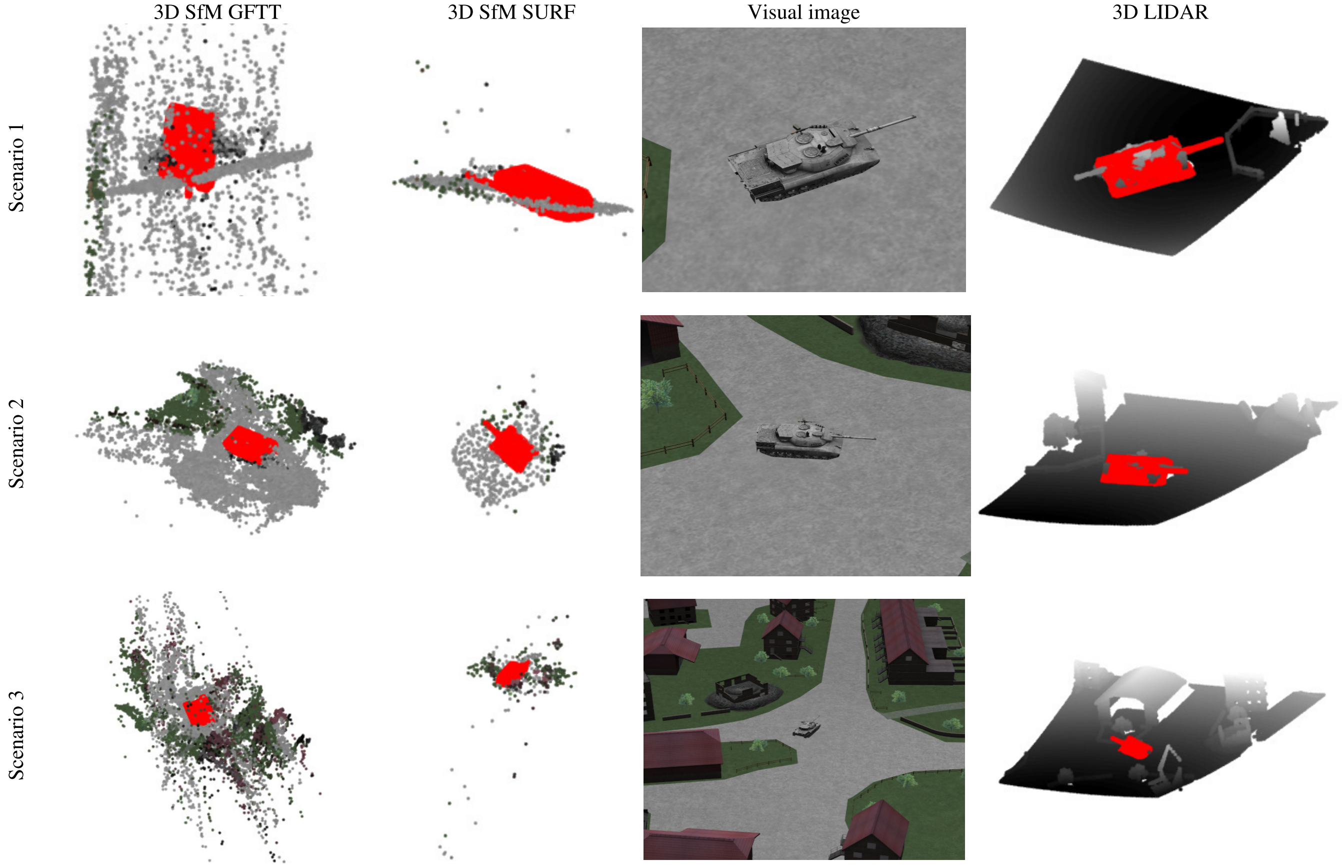


Fig. 7. Examples of 3-D ATR with SfM, exploiting only two images from the visual domain.

TABLE V
REQUIREMENT ANALYSIS

		3D							2D			
		SHOT	USC	HoD-S	HoD	FPFH	3DSC	RoPS	#1	#2	#3	#4
Template storage (KB)	SfM / LIDAR	2.9	15.9	0.32	1.92	0.26	15.9	1.1	2213	1389	1224	3106
Processing time / scene (s)	SfM GFTT	**0.42**	**0.57**	**0.25**	**0.26**	**0.38**	**0.69**	**0.30**	0.58	0.05	0.09	0.74
	SfM SURF	0.73	0.78	0.54	0.55	0.66	0.84	0.62				
	LIDAR	4	11.5	1.6	2.2	2.5	11.1	14.3				

optical axis (such that the target is not reconstructed in 3-D), the MBT will occupy the center of subsequent images as the UAV moves and thus the target will be reconstructed at some point. The 2-D ATR did not perform well because both the distance (scale) and obliquity exceeded the invariance of the 2-D descriptors. For the medium- and high-angle tests, more of the MBT's top view was revealed, which is more distinctive than the side view, favoring ATR. The overall performance of each method is shown in Fig. 6, highlighting the better performance of SfM 3-D ATR compared to 3-D LIDAR ATR. Fig. 7 shows 3-D ATR examples for both methods.

3) Computational and Storage Requirement Analysis: Recognition performance and computational efficiency are equally important for an ATR system. We, therefore, compared the 3-D SfM, 3-D LIDAR, and 2-D ATR methods in terms of their processing burden (Table V). Although SfM requires the scene to be reconstructed in 3-D before activating the rest of the pipeline (Fig. 2), 3-D SfM is faster than 3-D LIDAR because the SfM-based point cloud is sparser, speeding up the entire recognition process. Indeed, GFTT SfM produces a point cloud in the order of 10 000 vertices, whereas the equivalent values for SURF SfM and LIDAR are 500 and 260 000, respectively. A 3-D SfM exploiting GFTT keypoints combined with the HoD-S descriptor, therefore, requires only 0.25 s for completion, whereas the less efficient 3-DSC needs 0.69 s and 3-D SfM with SURF features needs up to 0.84 s. In contrast, the fastest 3-D LIDAR ATR was based on HoD-S (1.6 s) and the least efficient was RoPS (14.3 s). It is evident that the processing efficiency of the proposed 3-D SfM architecture is at least one order of magnitude faster than 3-D LIDAR ATR.

A detailed processing breakdown is shown in Fig. 8, indicating that the 3-D description of the SfM point cloud vertices is almost eight times faster than the LIDAR-based point cloud due to the sparsity of the SfM point cloud. This advantage is also evident from the considerably faster keypoint matching, correspondence hypothesis evaluation and verification achieved by both SfM methods.

As expected, the shortest processing time was observed for 2-D ATR. Although this is an appealing property, the template

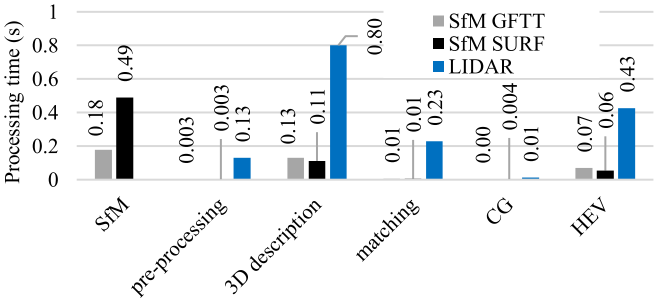


Fig. 8. Processing time breakdown (CG: correspondence grouping, HEV: hypothesis evaluation, and verification).

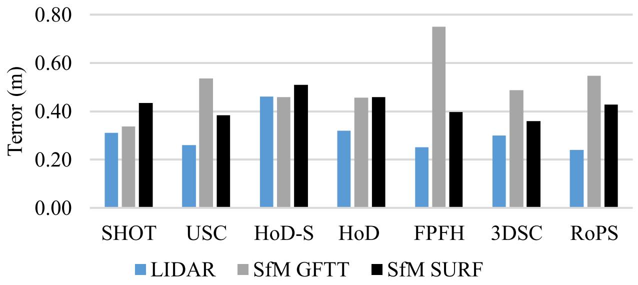


Fig. 9. Translational error evaluation.

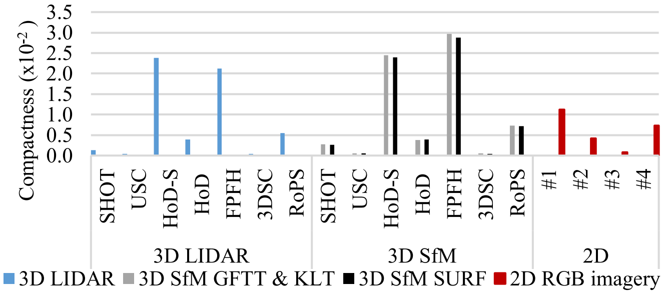


Fig. 10. Compactness.

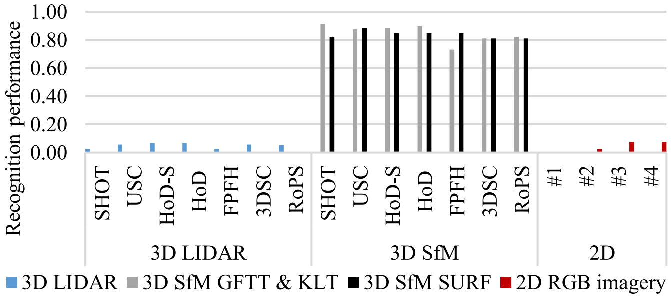


Fig. 11. Robustness to shot noise.

is reduced to a minimum, so expansion to provide more instances of the target at various viewing angles and ranges would increase the overall processing time. Furthermore, in terms of the storage capacity needed for template features, even in this minimal template case, the 2-D solutions already have greater requirements than their 3-D counterparts because the 3-D template is subsampled and only a few vertices from the entire CAD model are encoded.

4) Matching Accuracy: We also validated the 3-D SfM concept by highlighting the 3-D translational error (T_{error}) of each descriptor. Fig. 9 shows the T_{error} of the three 3-D approaches (GFTT SfM, SURF SfM, and LIDAR) for the UAV–target range of 200 m at 30° obliquity. For greater clarity, we evaluated the matching accuracy for the third scenario alone, which involves the largest UAV–target range among the three scenarios, and therefore is the most challenging.

As anticipated, 3-D LIDAR generated the smallest errors because the target within the point cloud was more complete than its corresponding sparse SfM reconstructions. Even so, both SfM solutions still produced low T_{error} values, confirming that the suggested SfM ATR architecture is an appealing creates that focuses on the target. For the GFTT SfM method, the largest T_{error} was generated by HoD-S (0.51 m), and for SURF SfM the largest value was generated by FPFH (0.75 m), but all these values are still very low. T_{error} fluctuations among the descriptors are related to the sparsity of the point cloud, whether the 3-D descriptor employs an LRF/LRA or not, and the concept used to estimate the LRF or LRA.

5) Compactness: This metric indicates the description power per element of a descriptor [10]

$$\text{compactness} = \frac{\text{F1-score}}{\#\ \text{descriptor cardinality}}. \tag{6}$$

Fig. 10 shows that for both LIDAR and SfM, HoD-S and FPFH were the most compact, with 3-D GFTT-based SfM displaying a minor advantage. The greater compactness of HoD-S and FPFH reflect the small feature length/cardinality of these descriptors, which in parallel achieve a competitive ATR performance. The least compact were USC and 3-DSC, because despite achieving better ATR performance compared to FPFH, their large feature length severely compromised their compactness.

Regarding the 2-D descriptors, even though their feature length is small, they all have a small compactness value due to their relatively poor ATR performance.

6) Robustness to Shot Noise: We compared the robustness of the proposed and competing ATR methods against shot noise by modeling shot noise with a Poisson distribution. Shot noise was applied on the core data required by each method. Hence, for the SfM 3-D ATR and the 2-D ATR tests, we applied shot noise directly to the 2-D RGB imagery, whereas for the 3-D LIDAR ATR test we applied shot noise to the vertices of the point cloud.

Specifically, we independently manipulated each pixel of the 2-D scene image $I_1(i,j)$, $1 \leq i \leq m$ and $1 \leq j \leq n$ according to

$$I_1(i,j) = e^{-I_1(i,j)} \frac{I_1(i,j)}{k!} \tag{7}$$

where $k \in \mathrm{N}^+$ randomly chosen. In the same manner, we applied shot noise to I_2. For the LIDAR 3-D ATR test, we independently manipulated the z-coordinate of each vertex in the LIDAR point cloud $\boldsymbol{P}$ according to the corresponding depth value of the 2-D depth image D that the LIDAR creates

$$\boldsymbol{P}(x,y,z) = \left[x \quad y \quad e^{-D(ii,jj)} \frac{D(ii,jj)}{k!} \right] \tag{8}$$

where ii and jj are the pixel coordinates of D.

Fig. 11 clearly shows that the SfM 3-D ATR architecture outperforms both competitors regardless of the descriptor. This is important because it demonstrates the advantages of using

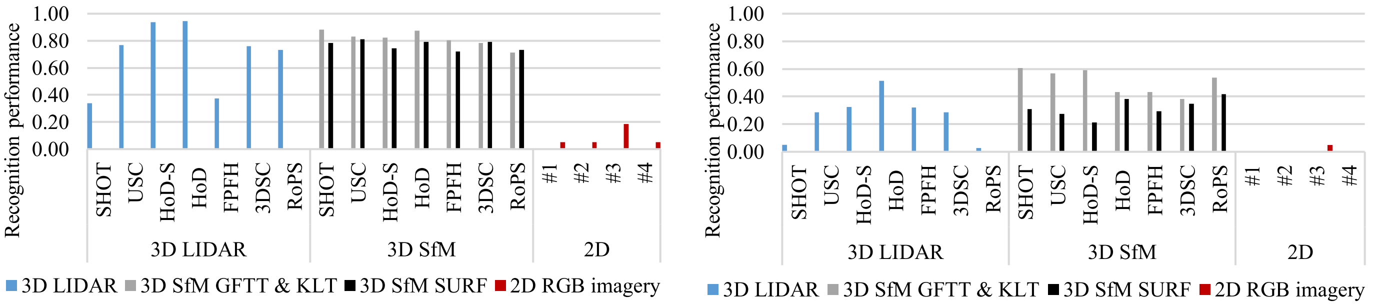


Fig. 12. Robustness to Gaussian noise with zero mean and (a) $\sigma = 10$ cm and (b) $\sigma = 30$ cm.

SfM rather than LIDAR 3-D data. The robustness of SfM 3-D ATR reflects the robustness of the 2-D local feature methods used in our SfM module, which successfully matched the images (corrupted by shot noise) acquired from the UAV's camera in order to create an accurate 3-D scene representation. As expected, the performance of the 2-D ATR pipeline was poor for the reasons presented in Sections III-D1 and III-D2.

7) Robustness to Gaussian Noise: We also evaluated the robustness of the proposed ATR technique under $\sigma = \{10, 30\}$ cm Gaussian noise levels [10]. Similar to the shot noise trial, we applied noise directly to the 2-D RGB imagery for both 3-D SfM and 2-D ATR, whereas for the LIDAR 3-D ATR the Gaussian noise was applied to the vertices of the point cloud.

Fig. 12(a) shows that for the $\sigma = 10$ cm Gaussian noise test, 3-D SfM ATR achieved a more stable performance, which was less dependent on the descriptor. In contrast, even though 3-D LIDAR ATR combined with the HoD and HoD-S descriptors achieved the highest ATR performance, our trials demonstrate that the selected descriptor had a substantial impact on the ATR performance.

For the $\sigma = 30$ cm Gaussian noise test, 3-D SfM ATR achieved a higher overall ATR capability [Fig. 12(b)]. This was more evident for the GFTT and KLT combination, where the majority of the descriptors achieved higher recognition rates than the best-performing 3-D LIDAR descriptor.

IV. Conclusion

We have developed a passive 3-D ATR architecture appropriate for small low-cost UAV platforms. Our architecture exploits the UAV's onboard sensors, i.e., visual band camera, gyroscope, and accelerometer, in order to create passive 3-D reconstructions of the UAV's surroundings. The 3-D scene thus created is input into a 3-D ATR pipeline. The method is appealing because it combines the advantages of 3-D and 2-D object recognition. Specifically, it combines the advantages of 3-D object recognition, such as pose and illumination invariance, exploiting the underlying structure of the target and reducing the template size to a single 3-D CAD model. In addition, it also preserves the advantages of 2-D object recognition, resulting in a small processing burden, low hardware costs (camera *vs* LIDAR), faster data acquisition, longer operating range, and undetectable passive operation.

We evaluated the new SfM ATR scheme by exploiting two 2-D keypoint detection and description techniques, i.e., the GFTT with a KLT tracker and the SURF with an NNDR criterion, and we tested these against classic 3-D LIDAR ATR and 2-D visual ATR. We measured target recognition performance over several UAV–target ranges and obliquities, as well as evaluating processing efficiency, translational matching accuracy, robustness to shot noise and to Gaussian noise, confirming its appealing features. One limitation of our technique compared to LIDAR 3-D is the constraint of sufficient lighting conditions, which reflect the camera's limitations. However, in the future, we intend to extend the 3-D SfM ATR concept to operate on low-light visual band cameras in order to improve the usability of the suggested architecture to include extreme lighting scenarios.

References

[1] G. J. Gray, N. Aouf, M. A. Richardson, B. Butters, R. Walmsley, and E. Nicholls, "Feature-based recognition approaches for infrared anti-ship missile seekers," *Imag. Sci. J.*, vol. 60, no. 6, pp. 305–320, 2012, doi: 10.1179/1743131X12Y.0000000012.

[2] O. Kechagias-Stamatis, N. Aouf, and D. Nam, "Multi-modal automatic target recognition for anti-ship missiles with imaging infrared capabilities," in *Proc. IEEE Sensor Signal Process. Defence Conf.*, Dec. 2017, pp. 1–5, doi: 10.1109/SSPD.2017.8233244.

[3] R. Paladini, M. Martorella, and F. Berizzi, "Classification of man-made targets via invariant coherency-matrix eigenvector decomposition of polarimetric SAR/ISAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 3022–3034, Aug. 2011, doi: 10.1109/TGRS.2011.2116121.

[4] D. Perissin and A. Ferretti, "Urban-target recognition by means of repeated spaceborne SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4043–4058, Dec. 2007, doi: 10.1109/TGRS.2007.906092.

[5] M. Martorella, E. Giusti, A. Capria, F. Berizzi, and B. Bates, "Automatic target recognition by means of polarimetric ISAR images and neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3786–3794, Nov. 2009, doi: 10.1109/TGRS.2009.2025371.

[6] M. Prashnani and R. S. Chekuri, "Identification of military vehicles in hyper spectral imagery through spatio-spectral filtering," in *Proc. IEEE 2nd Int. Conf. Image Inf. Process.*, Dec. 2013, pp. 527–532, doi: 10.1109/ICIIP.2013.6707648.

[7] A. Vasile and R. Marino, "Pose-independent automatic target detection and recognition using 3D laser radar imagery," *Lincoln Lab. J.*, vol. 15, no. 1, pp. 61–78, 2005.

[8] C. Grönwall, "Ground object recognition using laser radar data: Geometric fitting, performance analysis, and applications," Ph.D. dissertation, Dept. Elect. Eng., Linköping Univ., Linköping, Sweden, 2006.

[9] O. Kechagias-Stamatis, N. Aouf, and M. A. Richardson, "3D automatic target recognition for future LIDAR missiles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 6, pp. 2662–2675, Dec. 2016, doi: 10.1109/TAES.2016.150300.

[10] O. Kechagias-Stamatis and N. Aouf, "Evaluating 3D local descriptors for future LIDAR missiles with automatic target recognition capabilities," *Imag. Sci. J.*, vol. 65, no. 7, pp. 428–437, 2017, doi: 10.1080/13682199.2017.1361665.

[11] O. Kechagias-Stamatis, N. Aouf, G. Gray, L. Chermak, M. Richardson, and F. Oudyi, "Local feature based automatic target recognition for future 3D active homing seeker missiles," *Aerosp. Sci. Technol.*, vol. 73, pp. 309–317, Feb. 2018, doi: 10.1016/j.ast.2017.12.011.

[12] S. Matteoli, G. Corsini, M. Diani, G. Cecchi, and G. Toci, "Automated underwater object recognition by means of fluorescence LIDAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 375–393, Jan. 2015, doi: 10.1109/TGRS.2014.2322676.

[13] S. K. Rogers *et al.*, "Neural networks for automatic target recognition," *Neural Netw.*, vol. 8, nos. 7–8, pp. 1153–1184, 1995.

[14] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1584–1601, Oct. 2006, doi: 10.1109/TPAMI.2006.213.

[15] S. Y. Bao and S. Savarese, "Semantic structure from motion: A novel framework for joint object recognition and 3D reconstruction," in *Outdoor and Large-Scale Real-World Scene Analysis*. Berlin, Germany: Springer, 2012, pp. 376–397.

[16] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 44–57.

[17] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3D scene analysis from a moving vehicle," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[18] M. Shim, S. Yilma, and K. Bonner, "A robust real-time structure from motion for situational awareness and RSTA," *Proc. SPIE*, vol. 6962, p. 696205, Apr. 2008, doi: 10.1117/12.778074.

[19] Y.-M. Wei, L. Kang, B. Yang, and L.-D. Wu, "Applications of structure from motion: A survey," *J. Zhejiang Univ. Sci. C*, vol. 14, no. 7, pp. 486–494, 2013, doi: 10.1631/jzus.CIDE1302.

[20] Y. P. Huang, L. Sithole, and T. T. Lee, "Structure from motion technique for scene detection using autonomous drone navigation," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2745419.

[21] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 24, doi: 10.1145/1276377.1276407.

[22] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Ithaca, NY, USA: IEEE Computer Society Press, Jun. 1994, pp. 593–600, doi: 10.1109/CVPR.1994.323794.

[23] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, vol. 2, 1981, pp. 674–679. Accessed: Mar. 15, 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=1623280

[24] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Proc. 20th Int. Conf. Pattern Recognit. (ICPR)*, 2010, pp. 2756–2759, doi: 10.1109/ICPR.2010.675.

[25] D. P. Robertson and R. Cipolla, *Structure From Motion*. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 1–49. [Online]. Available: http://academic.research.microsoft.com/Search?query=structure+from+motion

[26] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008, doi: 10.1016/j.cviu.2007.09.014.

[27] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. II-257–II-263, doi: 10.1109/CVPR.2003.1211478.

[28] S. Knorr, M. Kunter, and T. Sikora, "Stereoscopic 3D from 2D video with super-resolution capability," *Signal Process., Image Commun.*, vol. 23, no. 9, pp. 665–676, 2008.

[29] Presagis. *OpenFlight Visual Simulation*. Accessed: Aug. 1, 2016. [Online]. Available: http://www.presagis.com/products_services/standards/openflight/

[30] X. Li, J. Xu, J. Luo, L. Cao, and S. Zhang, "Ground target recognition based on imaging LADAR point cloud data," *Chin. Opt. Lett.*, vol. 10, p. S11002, 2012, doi: 10.3788/COL201210.S11002.

[31] C. Gronwall, F. Gustafsson, and M. Millnert, "Ground target recognition using rectangle estimation," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3400–3408, Nov. 2006.

[32] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Comput. Vis. Image Understand.*, vol. 125, pp. 251–264, Aug. 2014, doi: 10.1016/j.cviu.2014.04.011.

[33] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, 2013, doi: 10.1007/s11263-013-0627-y.

[34] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2009, pp. 3212–3217, doi: 10.1109/ROBOT.2009.5152473.

[35] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2004, pp. 224–237, doi: 10.1007/978-3-540-24672-5_18.

[36] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *Proc. ACM Workshop 3D Object Retr. (DOR)*. New York, NY, USA: ACM Press, 2010, p. 57, doi: 10.1145/1877808.1877821.

[37] O. Kechagias-Stamatis and N. Aouf, "Histogram of distances for local surface description," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2016, pp. 2487–2493, doi: 10.1109/ICRA.2016.7487402.

[38] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "TriSI: A distinctive local surface descriptor for 3D modeling and object recognition," in *Proc. 8th Int. Conf. Comput. Graph. Theory Appl.*, Barcelona, Spain, 2013, pp. 86–93, doi: 10.5220/0004277600860093.

[39] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, 2016, doi: 10.1007/s11263-015-0824-y.

[40] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, 2000.

Odysseas Kechagias-Stamatis received the M.Sc. degree in guided weapon systems and the Ph.D. degree in 3-D ATR for missile platforms from Cranfield University, Shrivenham, U.K. in 2011 and 2017, respectively.

His research interests include 2-D/3-D object recognition and tracking, data fusion and autonomy of systems.

Nabil Aouf is currently the Head of the Signals and Autonomy Group, Centre for Electronic Warfare information and Cyber, Cranfield University, Shrivenham, U.K. He has authored over 100 publications in high caliber in his domains of interest. His research interests include aerospace and defense systems, information fusion and vision systems, guidance and navigation, tracking, and control and autonomy of systems.

Prof. Aouf is an Associate Editor of the *Imaging Science Journal*.