

7 MODELLING THE IR SCENE IN A 3D VIRTUAL WORLD

7.1 Introduction

To visualize the IR signature of a scene, it is essential to first model the physical geometry of the entire scene from the missile seeker's viewpoint. The typical ingredients of an IR signature scene could be the targets, sky, ground and flares etc. The geometrical modelling and the movement of these objects in world coordinates could be done by using three dimensional (3D) virtual reality modelling tools. In this chapter, the steps involved in the development of 3D models of the targets, backgrounds and the flares are explained. The methods for controlling the missile, target and flare movements during dynamic missile-target engagement and countermeasures dispensation simulation are explained. Also the steps involved in the controlling of the appearance of the targets, background and flares as per their thermal radiations and radiometric properties are discussed. The steps involved in the modelling of the missile gimballed seeker head are also discussed.

7.2 Why VRML

For 3D modelling, a large number of representation formats are available. Out of these, because of the free resources available, VRML and Open InventorTM are the two most widely used formats [DUT04]. In this work, VRML has been selected for building the 3D virtual reality world. VRML is commonly used for visualization of physical worlds. The 3D models of several military targets are available on the internet in VRML format. As for computational algorithms, MATLAB has been used in this work and for 3D visualization MATLAB uses VRML. Therefore, it was the most convenient to build the 3D IR scene in VRML and control the physical properties and movements of the objects from algorithms written in MATLAB source code.

7.3 Developing the 3D Virtual World

The missile-target engagement scenario has been visualized in three dimensions. The development of a 3D scenario starts with a story board in which the key features (such as the targets, missile, flares and background etc.) are build and placed at appropriate locations in world coordinates. For simulation a plan script is written and animation

sequences are developed. The viewpoints are selected for analysing the scene from different aspects. In this work, the tool used for development of the 3D virtual world is V-Realm builder™ which was provided with MATLAB Virtual Reality Toolbox. The following paragraph explains the procedure for building the 3D virtual world for missile-target engagement scenario and countermeasures dispensation simulation.

7.4 Military Targets 3D VRML Models

There are several places on the internet from where the 3D VRML models of military targets can be downloaded. These sites are listed out in Chapter-5 Table 5-1. The models downloaded from the internet consist of hundreds of polygons. These polygons can be grouped as small sub-parts. These small parts with the same thermal and radiometric properties may be represented as sub-targets in the IR signature model. As the sub-targets material properties can be controlled independently, these models may be considered as high-fidelity from the IR signature modelling point of view. However, to increase the fidelity further, the models with higher resolution (with a higher number of polygons and more details) may be used instead and divided into further sub-targets. The VRML models of a few military targets have been downloaded from open sources on the internet. The modifications done to make them workable for the IR signature modelling are explained in paragraph 7-5. These models are selected to represent different groups of target platforms which could be used for missile-target engagement simulations.

7.4.1 F-16 Fighting Falcon 3D VRML Model

Figure 7-1 shows the orthographic projection of the F-16 model along with actual dimensions. For modelling purposes, the F-16 represents a typical military fighter aircraft group. The F-16 VRML model was downloaded from the internet [FAS06a]. Figure 7-2 shows the VRML model of the F-16 before any alterations are done. The exhaust gas plume and leading-edge geometry were not present in the original model. The details of the modifications carried out are explained in paragraph 7-5 of this chapter.

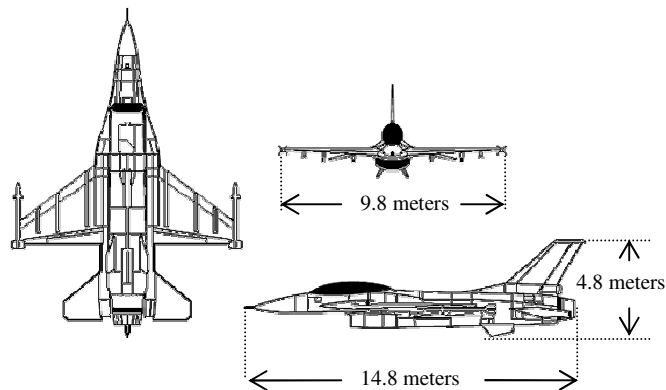


Figure 7-1 : Orthographic projection of F-16 model (Source [FAS06a])

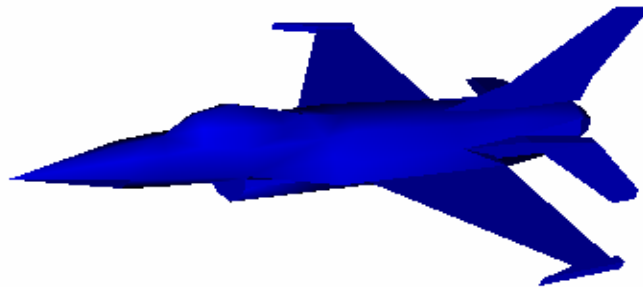


Figure 7-2 : VRML 3D model of F-16 before modifications (Source [FAS06a])

7.4.2 C-130 Hercules 3D VRML Model

The C-130 Hercules represents a typical military transport aircraft. Figure 7-3 shows the orthographic projection and the actual dimensions of the C-130 aircraft. The C-130 VRML 3D model was downloaded from the internet [FAS06b].

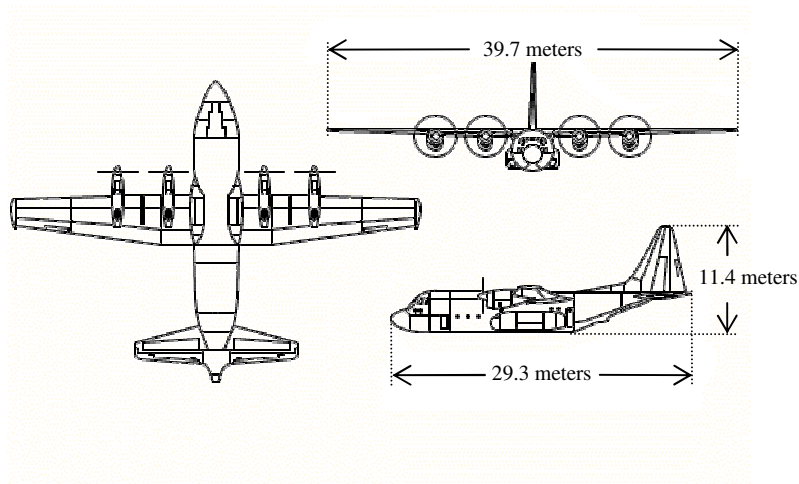


Figure 7-3 : C-130 Hercules Orthographic projection (source [FAS06b])

Figure 7-4 shows the VRML 3D model of the C-130. The VRML model of the C-130 may be transformed into the following sub-zones (or sub-targets) to make this usable for IR signature modelling.



Figure 7-4 : VRML 3D model of C-130 Hercules aircraft (Source [FAS06b])

- (a) Body of the aircraft (fuselage, tail and wings),
- (b) Nose of the aircraft,
- (c) Engines (number 1, 2, 3 and 4 separately),
- (d) Engine exhaust gas plume (separate for each engine),

- (e) Engine propellers (number 1, 2, 3 and 4 separately),
- (f) Leading-edges of wings and tail section,
- (g) Cockpit glass windows,
- (h) Flare dispenser unit.

7.4.3 T-62 Tank 3D VRML Model

Similarly, the 3D VRML model of a T-62 tank was downloaded from the internet [VRM06] and divided into sub-sections. The division is done on the basis of their temperature and radiometric properties. Figure 7-5 shows the 3D model of the tank. For the desired fidelity, the model may be divided into sub-zones and the geometry of the exhaust gas plume added in the same VRML file. Based on the different temperature and radiometric properties, the tank may be divided into the following possible sub-zones.

- (a) Tracks and wheels,
- (b) Machine guns and gun barrel,
- (c) Moving turret and,
- (d) Mudguards, fuel tank and other body parts,
- (e) Engine top cover at the back of the turret,
- (f) Exhaust gas plume,
- (g) Dust/ smoke.



Figure 7-5 : T-62 tank 3D VRML model

7.4.4 Boeing B-737 3D VRML Model

Although, the Boeing-737 is a civil airliner it may be a possible target for surface-to-air missile. Therefore, the IR signature modelling of an airliner may be required for analysing the countermeasures for civil airliner against surface-to-air missiles. The Boeing-737 typically represents two engine civil airliners. However, for four engine airliners, a model of the Boeing-747 may be used to represent large airliners. The Boeing-737 3D VRML model was downloaded from the internet [VRM06]. The different parts of the Boeing-737 could be group as sub-targets. The following list describes one possible division of the airliner as sub-targets. However, depending upon the nature of the simulation, this could be altered accordingly.

- (a) Body (Fuselage)
- (b) Wings (left and right)
- (c) Engines (left and right separate)
- (d) Engine exhaust gas plume (left and right separate)
- (e) Leading Edges (wings and Tail)
- (f) Nose of the aircraft
- (g) Windows (Cockpit and passenger)
- (h) Flare dispenser unit (one or more locations)

Figure 7-6 shows the VRML 3D model of Boeing-737 airliner. The exhaust gas plume and glass windows are not shown in the model which could be added in the same VRML file.

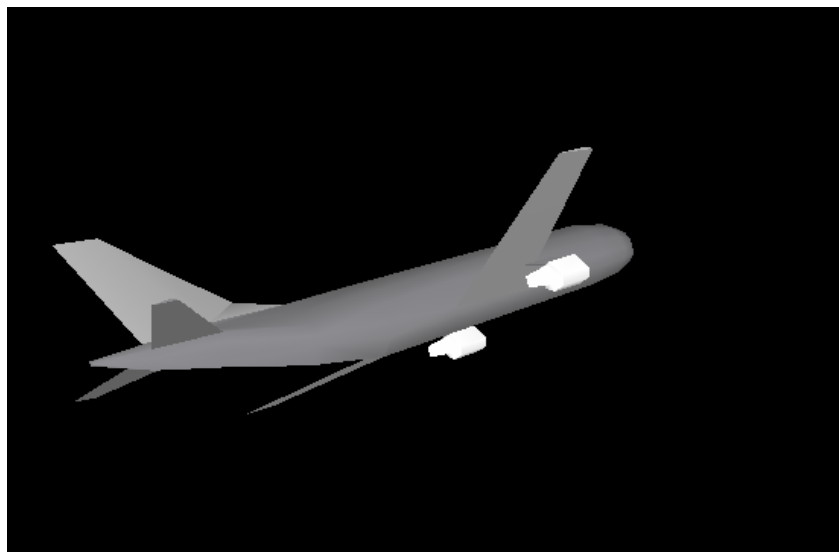


Figure 7-6 : Boeing-737 3D VRML model (Source [VRM06])

7.4.5 Helicopter 3D VRML Model

Figure 7-7 shows a 3D VRML model of a helicopter downloaded from the internet [VRM06]. For the IR signature modelling the helicopter may be divided into the following sub-parts. Depending upon the nature of the analysis and the desired fidelity, the division may be altered accordingly.

- (a) Moving parts (propeller rod, horizontal blades and vertical-blades etc.)
- (b) Body parts, missiles and bombs
- (c) Canopy glass (windscreen)
- (d) Engine top cover, exhaust section and guns
- (e) Exhaust gas plume



Figure 7-7 : 3D VRML model of a military helicopter (Source [VRM06])

7.5 Altering Target VRML Model for IR Signature Modelling

Generally, the 3D VRML models available on the internet are based on the optical properties of the visible spectrum. Typically, the reflection and transparency properties of the material as per the visible spectrum are considered for rendering. These models are not made specifically for use in the IR waveband. Therefore, the following are the changes which are incorporated to make the original models workable for high-fidelity IR signature modelling.

- (a) Grouping parts with the same temperature or radiometric properties as one sub-target.
- (b) Separating parts with different temperature or optical properties which were initially as one part.
- (c) Adding exhaust gas plume geometry in the target VRML model.
- (d) Adding flare dispenser geometry at the appropriate location on the host platform.
- (e) Altering transformation nodes of the target for controlling rotation and translation during simulation.
- (f) Altering material properties (the colour and appearance) of each sub-target as per their thermal and radiometric properties.

The changes listed above need to be applied to all targets used in the IR signature model. Although, in this work, only one model of the F-16 is altered to demonstrate the procedure. However, by following the same procedure, the VRML models of other military targets may also be modified accordingly.

7.5.1 Grouping Parts as Sub-targets

For high-fidelity IR signature modelling, the aircraft parts with the same temperature profiles and the same radiometric properties may be grouped as one sub-target so that their appearance can be controlled as per the radiance values calculated in Chapter-6. The original 3D model is divided in many sub-parts such as the fuselage, right-wing, left-wing, tail section, engine intake, engine exhaust pipe, nose and canopy. Each part of the model is given a separate name so that these act as separate identities and can be independently controlled from the main algorithm written in MATLAB. Thus, the parts with the same temperature and radiometric properties are grouped as one sub-target, whereas, on the other hand, the parts that are attached but have different properties (such as the leading edges of the wings and tail) are separated by applying another surface on the original part. The parts which are not present (such as exhaust gas plume and flare dispenser) are added to the original model by altering the VRML code in V-Realm builderTM. The following are typical sub-parts (sub-targets) of a target aircraft grouped on the basis of their thermal and radiometric properties.

- (a) Body (Fuselage, wings and tail),
- (b) Nose leading edges,
- (c) Leading edges on wing and vertical tail section,
- (d) Engine Exhaust nozzle,
- (e) Engine intake,
- (f) Glass canopy,
- (g) Exhaust gas plume.

7.5.2 Modelling Appearance of Sub-targets

In the VRML file format, a 3D virtual world scene is described by a hierarchical tree structure of several objects. These objects are called “nodes”. A node may contain other nodes (child nodes) under it. When a new node is created, the required fields are automatically generated and filled with default values. These fields hold the data for each node. In VRML, every geometrical object contains an independent “*geometry*” node and “*appearance*” node [VIR04]. The geometry node controls the size and shape of each part. Whereas, the appearance node controls the texture and material properties. The “*appearance*” node specifies the physical properties such as the colour, shininess, transparency etc. The material properties of each sub-part can be controlled independently. However, for grouping the parts which have the same material properties the VRML “*USE*” node is used [VIR04]. The “*USE*” node repeats the same material properties on any other object in the model. It is better to set some materials at the start and then use these as and when required. The following are the material fields used in VRML [VIR04].

- (a) **AmbientIntensity.** The “*ambientIntensity*” specifies the amount of ambient light the surface should reflect. Ambient intensity can be represented as any scalar value between 0.0 to 1.0 both inclusive. The higher the value the higher the reflection.
- (b) **DiffusedColour.** The “*diffuseColor*” field indicates more or less the colour of light that reflects/scatters most from the object. The diffused colour is specified in RGB (red, green and blue) colour.

- (c) **EmissiveColour.** The “*emissiveColor*” field specifies the light produced by a glowing object. Emissive colour is useful for displaying radiosity based models.
- (d) **Shininess.** The “shininess” specifies how well a surface is polished. Shininess values range from 0.0 for dull surface to 1.0 for a highly polished surface.
- (e) **SpecularColour.** The “*specularColor*” represents the colour of the highlights on a highly reflected object. This is the colour of the light that reflects directly back to the viewer.
- (f) **Transparency.** The “*transparency*” field specifies the transmittance of an object, ranging from 0.0 for completely opaque to 1.0 for a completely clear surface. Any value in between may represent a semi-transparent object.

The VRML lighting model is based on the simple illumination equations given in [FOL96]. It provides detailed equations which define colours to apply to each geometric object. These equations are designed to simulate the physical properties of light striking a surface [VRML97]. An ideal VRML implementation will evaluate the following equation at each point on a lit surface based on the ambient, diffuse, specular and emissive properties and also the attenuation due to the intervening media between the light sources, objects and the view points. The details of the VRML light model are given in Annex “I” to this thesis, however, following equation summarizes the over all RGB intensities (I_{rgb}) at each point on geometry:

$$I_{rgb} = I_{Frgb} \cdot (1 - f_o) + f_o \cdot [O_{Ergb} + \sum (on_i \cdot atten_i \cdot spot_i \cdot I_{Lrgb} \cdot (ambient_i + diffuse_i + specular_i))]$$

where: I_{Frgb} currently bound fog’s colour
 f_o Fog interpolation (as given in Annex “I”)
 O_{Ergb} Material emissive colour
 on_i 1, if light source affect this point, otherwise, 0
 $atten_i$ attenuation factor (as given in Annex “I”)
 $spot_i$ spotlight factor (as given in Annex “I”)
 I_{Lrgb} light “i” colour

$ambient_i$	ambient factor (as given in Annex “I”)
$diffuse_i$	diffuse factor (as given in Annex “I”)
$specular_i$	specular factor (as given in Annex “I”)

The VRML material fields are used to model the radiometric properties of objects and IR signature in virtual reality. The list of material fields to model different radiometric properties are given in Table 7-1.

7.5.2.1 Converting Sub-target Radiances to RGB Colours

The total radiance of each sub-target was calculated in Chapter-6 as Equation 6-3. The radiance values of sub-targets are then converted into corresponding “*emissiveColour*”. For a target with “*n*” number of sub-temperature zones, the radiance of each sub-target is stored in a vector as given in Equation 7-1.

$$N_{tgt} = [N_{sub-tgt_1}, N_{sub-tgt_2}, N_{sub-tgt_3}, \dots, N_{sub-tgt_n}] \quad (7-1)$$

The radiances are then normalized over the range of the colour-map by using Equation 7-2. The details of different colour-map used are discussed in paragraph 7-5-2-3. A colour-map of different sizes can be used to map radiance data. The typical colour-indices are 128, 256, 512 or 1024 etc. However, to display the entire range of data correctly, the colour-index should be of sufficient resolution. I have selected 256 colour-index which quantizes the entire radiance data in 256 steps. However, a higher value of colour-index may be selected for better realism.

$$N_{tgt_{norm}} = \frac{N_{tgt}}{N_{max}} \cdot colour_index \quad (7-2)$$

Where, $N_{tgt_{norm}}$ is the normalized radiance of sub-targets,

N_{max} is the maximum value of radiance available in the scene,

$Colour_index$ is the number of indexed RGB colours in the colour-map.

The normalized radiance of each sub-target is then converted into corresponding RGB values of the selected colour-map as given in Equation 7-3. These RGB values are used in the material fields of sub-targets.

$$Color_{igt} = \begin{bmatrix} red_{sub-tgt_1} & green_{sub-tgt_1} & blue_{sub-tgt_1} \\ red_{sub-tgt_2} & green_{sub-tgt_2} & blue_{sub-tgt_2} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ red_{sub-tgt_n} & green_{sub-tgt_n} & blue_{sub-tgt_n} \end{bmatrix} \quad (7-3)$$

7.5.2.2 Changing Material Fields of Sub-targets

The sub-targets are modelled in VRML as objects with different appearance. The material fields of each sub-target are altered to resemble their IR signature. The dull metallic surfaces (such as the body and exhaust nozzle etc. of the aircraft) may be modelled by representing the radiance values as “*emissiveColor*” in VRML. The shining metallic surfaces (such as leading edges) may be modelled by representing leading edge radiances as “*emissiveColor*”, “*diffuseColor*” and “*specularColor*” along with “*ambientIntensity*” and “shininess” fields representing the reflectivity (or 1-emissivity) of the surface. The exhaust gas plume radiance may be modelled as “*emissiveColor*” along with “*transparency*” values representing transmittance of the plume. The cold-sky reflections from the glass canopy may be modelled as the “*emissiveColor*” and “*specularColor*” corresponds to the values of the glass canopy radiance and sky radiance respectively. Table 7-1 summarizes the different material types and their corresponding material fields. Using these material properties, the IR signature of sub-targets with different radiometric properties are modelled in 3-5 and 8-12 micron wavebands. The simulations are extensively carried out at different ranges and aspects. The details of these simulations and the discussion on the results are done in Chapter-9.

Table 7-1 : Material fields to model different material types

Material Type	Aircraft Part	Ambient Intensity	Diffuse Colour	Emissive Colour	Shininess*	Specular Colour	Transparency**
Dull Metal	Body	0	0	Body Radiance	0	0	0
	Exhaust Nozzle	0	0	Nozzle Radiance	0	0	0
	Engine Intake	0	0	Engine Intake Radiance	0	0	0
Shining Metal	Nose	1.0 to 0.5	Nose Radiance	Nose Radiance	1-emissivity	Nose Radiance	0
	Leading Edges	1.0 to 0.5	Leading Edges Radiance	Leading Edges Radiance	1-emissivity	Leading Edges Radiance	0
Hot Gases	Exhaust Plume	0	0	Plume Radiance	0	0	1-emissivity
Glass	Canopy	1.0	0	Canopy Radiance	1-emissivity	Sky Radiance	0

* Shininess value ranges from 0 to 1 with 0 representing dull surface with 180 deg spread and 1 gives only 8.4 deg spread

** Transparency ranges between 0 and 1 with 0 representing opaque and 1 fully transparent

7.5.2.3 Selecting “Colour-map” to represent Radiance Values

In real life, the visible colours do not correspond to IR radiations (as the IR band is outside the visible spectrum). However, for realism the different values of radiance in the IR waveband may be represented as corresponding false colours. The predefined range of colours used to map an entire range of particular data values are called a “colour-map” [MAR96] [MAT07]. Figure 7-8 shows different colour-maps which are supported by MATLAB. These colour-maps represent different ranges of colours. The details of these are given as follows [MAT07].

- (a) **Autumn.** The “*Autumn*” colour-map varies smoothly from Red, through Orange to Yellow.
- (b) **Bone.** The “*Bone*” is the Gray-scale colour-map with a higher value for the Blue component. This colour-map is useful for adding an "electronic" look to Gray-scale images.
- (c) **Cool.** The “*Cool*” consists of colours that are shades of Cyan and Magenta. This varies smoothly from Cyan to Magenta.
- (d) **Copper.** The “*Copper*” varies smoothly from Black to bright Copper.
- (e) **Gray.** The “*Gray*” returns a linear Gray-shade colour-map.
- (f) **Hot.** The “*Hot*” varies smoothly from Black through shades of Red, Orange and Yellow, to White.
- (g) **HSV:** The “*HSV*” varies the hue component of the hue-saturation-value (HSV) colour model. The colours begin with Red, pass through Yellow, Green, Cyan, Blue, Magenta, and Return to Red. This colour-map is particularly appropriate for displaying periodic functions.
- (h) **Jet.** The “*Jet*” ranges from Blue to Red, and passes through the colours Cyan, Yellow, and Orange. It is a variation of the HSV colour-map. The “*Jet*” colour-map is associated with an astrophysical fluid jet simulation from the National Centre for Supercomputer Applications [MAT07].
- (i) **Lines.** The “*Lines*” produce a colour-map of colours specified by the shades of Gray.
- (j) **Pink.** The “*Pink*” contains pastel shades of pink. The pink colour-map provides sepia tone colorization of Gray-scale photographs.

- (k) **Spring.** The “*Spring*” consists of colours that are shades of Magenta and Yellow.
- (l) **Summer.** The “*Summer*” consists of colours that are shades of Green and Yellow.
- (m) **Winter.** The “*Winter*” consist of colours that are shades of Blue and Green.



Figure 7-8 : Different colour-maps supported by MATLAB (Source[MAT07])

Each of these thirteen colour-maps have been tried to see how they represent the radiance values in the two IR wavebands of 3 to 5 micron and 8 to 12 micron. The effect of the different colour-maps on the IR signature appearance is monitored by running the simulation several times. Some colour-maps do not effectively represent the IR signatures. Whereas, other colour-map produces much better realism of the IR signature scene. Finally, it is revealed that the “*Jet*” colour-map may be the most suitable to represent IR signature in both 3-5 and 8-12 micron wavebands. Also using the “*jet*” colour-map, the leading-edge reflections and canopy cold-sky reflections are obvious. The “*Summer*”, “*Cool*” and “*Hot*” also produced reasonable results. The details of the simulations performed and the results generated are discussed in Chapter-9.

7.5.3 Controlling the Translation and Rotation fields of the Target

In VRML, for controlling the movement of objects, each object is attached to an independent “transformation” node. Each node has the following fields which may be

varied from the main algorithm to control the motion of the object in the 3D virtual world [VRML97].

- (a) **Rotation.** The “*rotation*” field controls the orientation of the object. Equation 7-4 shows the rotation field of VRML. The axis of rotation is given by the direction-cosines (X_m , Y_m and Z_m) and the rotation angle (ϕ_{tgt}) is given in radians. The default direction is pointing towards the negative Z-axis (i.e. [0, 1, 0, 0]).

$$myworld.aircraft.rotation = [X_{m_{tgt}}, Y_{m_{tgt}}, Z_{m_{tgt}}, \phi_{tgt}] \quad (7-4)$$

- (b) **Translation.** The “*translation*” field specifies the location of the object in the virtual world coordinates. Equation 7-5 shows the translation field of VRML. The (X, Y, Z) shows the location of the target in the world coordinates.

$$myworld.aircraft.translation = [X_{tgt} \quad Y_{tgt} \quad Z_{tgt}] \quad (7-5)$$

- (c) **Centre.** The “*centre*” field specifies the origin of the object to be taken as a reference for the rotation operation and relative distance between the target and missile.

7.5.4 Adding Leading-edge Geometry

In VRML, the “*geometrical*” node contains many physical shapes such as sphere, cylinder, box, cone and indexed-face-set (to defined irregular shape) [VIR04]. Using these nodes, the geometry of any physical object can be modelled. To show the leading-edge effects, on the wings and tail section, a separate geometry is added on their edges. Figure 7-9 shows the leading-edges on an F-16 model. The leading-edge geometry is formed by using the “*box*” node of VRML [VRML97]. The dimensions and shape are matched and the leading-edge geometry is superimposed on the wings and tail geometry. The nose section geometry is also separated from the main body so that the aerodynamic heating effects on the leading-edge can be highlighted. The leading-edge geometry is added in the F-16 VRML model by altering the original VRML file in V-Realm builder.

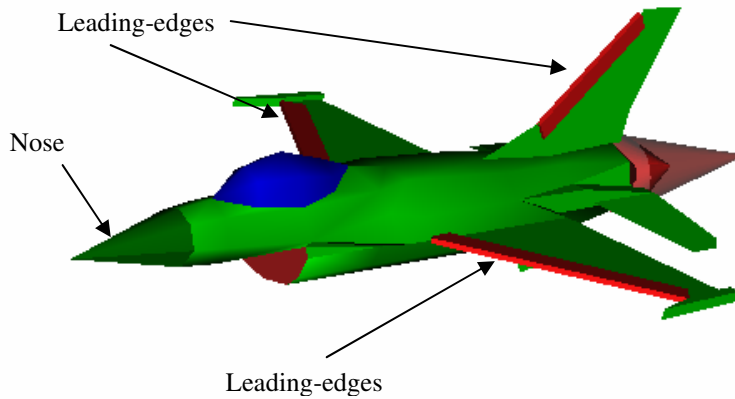


Figure 7-9 : F-16 3D VRML model after modifications

7.5.5 Adding Exhaust Gas Plume Geometry

The aircraft exhaust gas plume geometry may be created by using the “*cone*” geometrical node of VRML [VIR04]. The plume may be modelled as layers of co-centric cones as shown in Figure 7-10. These cones represent the inviscid core of the engine exhaust which contains the hottest exhaust gas region as discussed in paragraph 6.3.5 of Chapter-6. The plume is placed as a “*child node*” to the main F-16 model so that the plume moves along with the body of the aircraft in the world coordinates. Figure 7-11 shows the two cones of exhaust gas plume at the back of the F-16 3D VRML model.

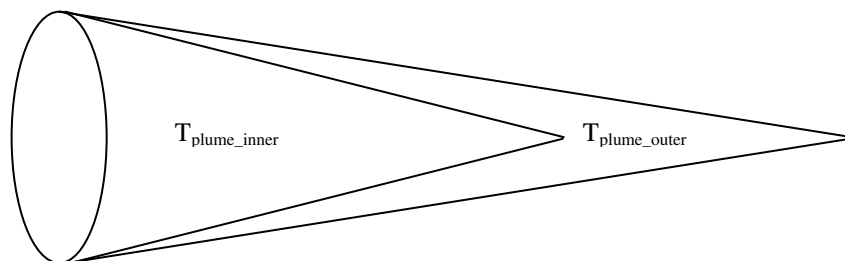


Figure 7-10 : Exhaust gas plume modelled as co-centric cones

The Equation 6-5 of Chapter-6 contains two components of plume radiance. The radiance due to the emissive properties of the plume ($N_{plumemis}$) and the other component of the plume radiance is due to the transmittance as ($\tau_{plume} \cdot N_{behindplume}$). The first component may be represented as the “*emissiveColour*” in VRML. Whereas, the second component may be modelled by making the plume semi-transparent with the “*transparency*” field of VRML representing the plume transmittance (τ_{plume}) and the colour of the background is representing the radiance behind the plume ($N_{behindplume}$). During the missile-target engagement simulation, the radiance behind the plume is varying with the change in aspect. However, in the VRML 3D virtual world, this is automatically catered for as the colour of the object behind the semi-transparent plume is added to the total colour of the plume as per the viewing aspect.

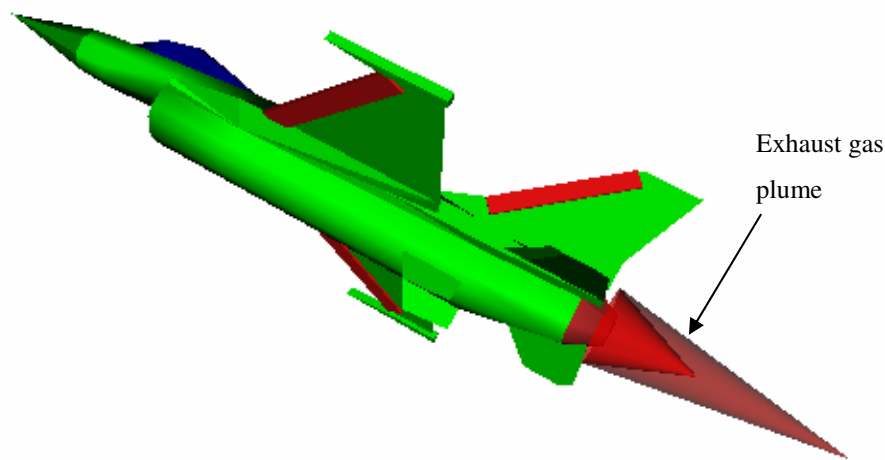


Figure 7-11 : F-16 3D VRML model with exhaust gas plume

7.6 Modelling IR Background

Based on the missile-target engagement scenario, the background can be planned accordingly. For the air-to-ground mode, the background may be objects such as runways, buildings, ground, sand, woodland, hills etc or any combination of these. In the case of surface-to-air missile simulation, the background is typically the sky or clouds. Whereas, in the case of air-to-air missiles, depending upon the target and

missile relative positions, the background may be either sky or ground or both (horizon). Therefore, depending upon the planned scenario, any type of background may be modelled accordingly. In VRML, there is a provision for the readymade “*background*” node that models the whole space around any scene as layers of sky in the upper hemisphere and layers of ground in the lower hemisphere. The colour and size of these layers can be controlled by the following fields [VIR04].

- (a) **Ground Angles.** The ground can be divided into “*n*” number of layers with “*n-1*” angle fields below the horizon.
- (b) **Ground Colour.** The colour of “*n*” layers of ground can be selected independently as the RGB values.
- (c) **Sky Angle.** The sky can be divided into “*n*” number of layers with “*n-1*” angle fields above the horizon.
- (d) **Sky Colour.** The colour of “*n*” layers of sky can be selected independently as the RGB values.

The “*background*” node of VRML may possibly be used to model the IR background in the air-to-air and surface-to-air missile engagement simulations. The sky is divided into three layers with two angles dividing the upper hemisphere. However, based on the desired fidelity, the number of layers of the sky may be increased or decreased accordingly. The colour of each layer corresponds to the sky radiance at selected altitudes. The sky radiance at different altitudes is calculated by running the LOWTRAN transmission code as explained in Chapter-6 paragraph 6.2.2. In general terms for “*n*” number of layers, the sky radiance values of all layers may be stored in one vector as given in Equation 7-6.

$$N_{sky} = [N_{layer_1}, N_{layer_2}, N_{layer_3}, N_{layer_4}, \dots, N_{layer_n}] \quad (7-6)$$

The sky radiance was normalized over the colour-index as given in Equation 7-7.

$$N_{sky_{norm}} = \frac{N_{sky}}{N_{max}} \cdot colour_index \quad (7-7)$$

Where, $N_{sky_{norm}}$ is the normalized radiance of all layers of sky,

N_{max} is the maximum value of radiance available in the scene,

$Colour_index$ is the number of indexed RGB colours in the colour-map.

The normalized sky radiance is then converted into corresponding RGB values of the selected colour-map as given in Equation 7-8. These RGB values are then used as the “*sky colour*” in the background node.

$$Color_{sky} = \begin{bmatrix} red_{layer1} & green_{layer1} & blue_{layer1} \\ red_{layer2} & green_{layer2} & blue_{layer2} \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ red_{layer_n} & green_{layer_n} & blue_{layer_n} \end{bmatrix} \quad (7-8)$$

However, the ground in the lower hemisphere is not divided into any layers. The ground is kept as a uniform one layer. The ground is modelled as per the ground radiance values calculated in Chapter-6 Equation 6-1.

7.7 Modelling IR Atmosphere as “Fog” in VRML

In VRML, the “*fog*” node allows for the reduction in visibility to simulate the effects of fog, smoke, clouds or other obscuring atmospheric phenomena. The fog is implemented by simply blending the given fog colour with everything in the scene [HAR96]. There are two types of fog available; the linear and exponential. The fog type controls how rapidly the fog colour is blended with an objects colour as a function of distance. The “*linear*” fog allows gradual fading over distance while “*exponential*” fog decreases visibility more rapidly as distance increases. The “*exponential*” fog representation is closer to real fog [HAR96]. The following are the fields which control the fog features:

- (a) **Fog Colour.** The “*fog colour*” field selects the RGB value for the colour of the fog.
- (b) **Fog Type.** Linear or exponential, the rate at which the fog thickens as distance increases. The “*fog type*” can be selected out of these two options.
- (c) **Visibility Range.** The maximum distance beyond which the fog obscures everything.

An effort has been made to model the effects of atmospheric attenuation in the IR waveband by using the “*fog*” node of VRML. The colour of the *fog* is kept as that

of the background and the visibility range as the maximum range beyond which the seeker can not detect targets. The exponential *fog* is selected to model the transmission in the IR waveband. In VRML, the fog is calculated as a function of range $f(R)$ as given in Equation 7-9 [VRML97].

$$f(R) = e^{\frac{-R}{\text{Visibility_Range} - R}} \quad (7-9)$$

Although, the *fog* feature of VRML is not a true representation of the atmospheric effects in the IR waveband and can not produced accurate results as compared to those modelled by the LOWTRAN or any other atmospheric propagation code. However, in future work the possibility of extracting range information from Z-buffer and then interpolating LOWTRAN data to provide more accurate results could be studied.

7.8 Modelling Multiple Viewpoints

In VRML, a “*viewpoint*” node describes a point in the world coordinates for viewing the scene from different aspects. The following fields are required to create a viewpoint:

- (a) **Field-of-view.** The “*field-of-view*” specifies the angle in radians that indicates the projected area of the scene which is visible during rendering.
- (b) **Orientation.** The “*orientation*” field specifies the rotation relative to the default orientation of looking down the negative Z-axis. If the viewpoint is a child node, then the viewpoint orientation is affected by the parent transformation rotation.
- (c) **Position.** The “*position*” field specifies the relative location of the viewpoint in the world coordinate system. The position of the viewpoint is affected by the parent transformation if the viewpoint is a child node.

To monitor the missile-target engagement simulation from different aspects, multiple viewpoints are defined. Each viewpoint has independent “*position*”, “*orientation*” and “*FOV*” fields. The following are the viewpoints which have been created in the VRML model:

- (a) **Gimballed Seeker-head Viewpoint.** The gimbal viewpoint is placed as a child node to the gimballed seeker head geometry. The gimbal viewpoint shows all that the missiles’ sensor sees or detects. The FOV field represents the seeker optics FOV and the orientation field represents the direction in which the seeker head is looking. The seeker head viewpoint is integrated in the missile tracking algorithm to capture the image as seen by the missile detector.
- (b) **Missile-LOS Viewpoint.** The missile-LOS view is created as a child node to the body of the missile. The “*orientation*” field is set to look in the direction of the missile motion (missile-LOS). This view monitors the missile’s heading at any instance during the simulation.
- (c) **Missile Launcher Viewpoint.** To observe the scene from the launchers viewpoint, a viewpoint is attached as a child node to the launching platform. For surface-to-air missiles, this viewpoint is a static point on the ground. However, for air-to-air missile this may move with the host aircraft.
- (d) **Aerial Viewpoint (Bird’s-eye View).** The aerial viewpoint is placed high above the target in a look down mode. This viewpoint gives the overall picture of the missile-target engagement scenario.
- (e) **Fly-along Missile Viewpoint.** The fly-along missile viewpoint is created by attaching a viewpoint node to the body of the missile at some distance to observe the movement of the missile and gimballed seeker head.

7.9 Modelling Missile Optics and Seeker-head

For the missile-target engagement simulation, the “*translation*” and “*rotation*” fields of the missile-LOS (body) and gimballed seeker-head are integrated with the missile tracking and guidance algorithm. For IR signature modelling of the missile seeker, the IR scene is what the IR sensor/detector captures. Thus the missile itself will not be

part of the IR scene. Therefore, the appearance and material properties of the missile are not modelled. However, for controlling the movement and rotation of the missile, the 3D VRML model of a missile is added in the VRML code. The missile VRML model was downloaded from the internet [VRM96]. The gimballed seeker head geometry is attached as a child node to the missile body. The seeker head can rotate independently in the hemisphere in front of the missile. A viewpoint is attached to the seeker head to model the seeker head optics and detector. The “FOV” and “orientation” fields of the viewpoint are modelled as per the seeker optics. The image captured by the gimbal viewpoint models the detector output. The following fields of the missile 3D VRML model are integrated with the missile tracking and guidance algorithm to control the missile and seeker head position and orientation at any instance during simulation:

- (a) **Missile-LOS Translation Field.** The missile-LOS translation field specifies the location of the missile body in 3D world coordinates. Equation 7-10 gives the field values which show the missile location in 3D world coordinates.

$$myworld.missile_LOS.translation = [X_{mst} \quad Y_{mst} \quad Z_{mst}] \quad (7-10)$$

- (b) **Missile-LOS Rotation Field.** The missile-LOS rotation field specifies the direction of motion of the missile (missile-LOS) in 3D world coordinates. The missile-LOS rotation field given in Equation 7-11 shows the rotation vector (X_m, Y_m, Z_m) and the rotation angle (ϕ_{mst_LOS}) .

$$myworld.missile_LOS.rotation = [X_{mst} \quad Y_{mst} \quad Z_{mst} \quad \phi_{mst_LOS}] \quad (7-11)$$

- (c) **Missile Gimbal Rotation Field.** The missile gimballed seeker head is attached to the body of the missile. The rotation of the gimballed seeker head is specified by an independent rotation field in the world coordinates. Equation 7-12 shows the rotation vector and the rotation angle (ϕ_{gimbal}) .

$$myworld.gimbal.rotation = [X_{m_{gimbal}} \quad Y_{m_{gimbal}} \quad Z_{m_{gimbal}} \quad \phi_{gimbal}] \quad (7-12)$$

7.9.1 Missile Optics Field-of-View

The field-of-view (FOV) of an electro-optical system is an important design parameter [DRI99]. The FOV describes the angular space in which the system accepts the IR energy. The missile FOV and the range from the sensor to the object (R_{TMD}) determine the area that a sensor can capture. Consider the optical system shown in Figure 7-12, the FOV of missile optics depends upon two factors. The seeker optics focal length (f) and the diameter (D_o). The Equation 7-13 is derived from the geometry shown in Figure 7-12.

$$FOV = 2 \tan^{-1} \left(\frac{D_o}{2f} \right) \quad (7-13)$$

Whereas, the instantaneous FOV (IFOV) is the range of the incident angle seen by a single detecting element. The IFOV calculated in Equation 7-14 depends upon the focal length (f) and the size of detector element (d).

$$IFOV = 2 \tan^{-1} \left(\frac{d}{2f} \right) \quad (7-14)$$

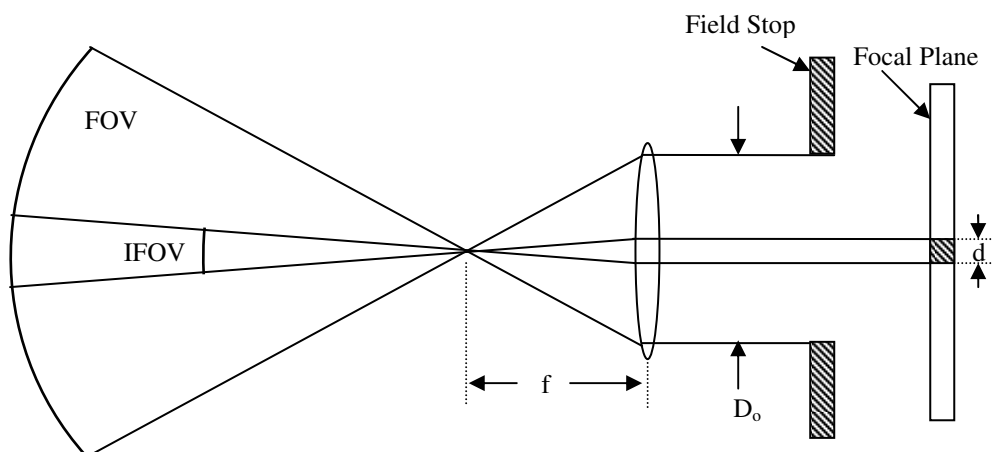


Figure 7-12 : Instantaneous field-of-view (source [FAS07b])

The horizontal and vertical IFOV in terms of their corresponding detector side length as shown in Figure 7-13 can be calculated as given in Equation 7-15 and 7-16 respectively.

$$IFOV_{horizontal} = 2 \tan^{-1} \left(\frac{b}{2f} \right) \quad (7-15)$$

$$IFOV_{vertical} = 2 \tan^{-1} \left(\frac{a}{2f} \right) \quad (7-16)$$

In VRML, the FOV used in the “*viewpoint*” node is the vertical FOV which corresponds to the vertical height (*a*) of the detector. However, for a square shape detector the width and height is the same and can be represented as “*d*” (i.e. $d=a=b$).

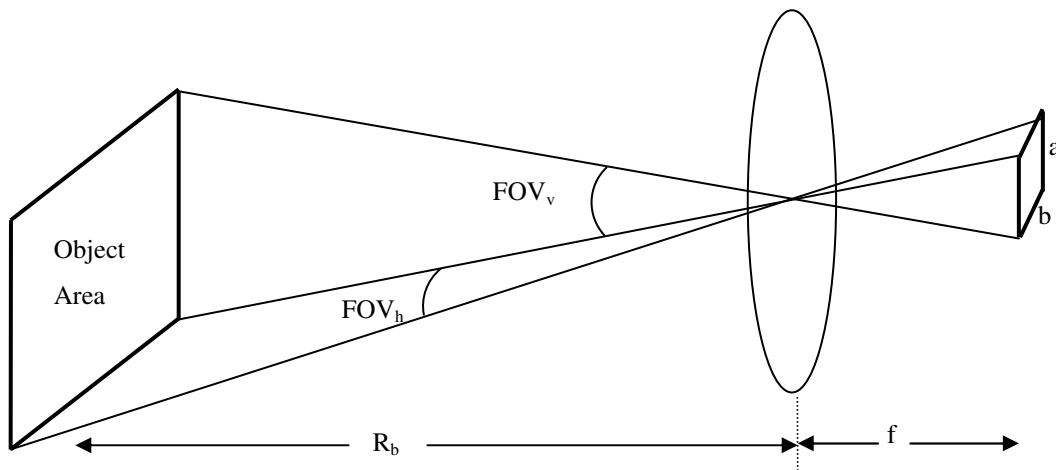


Figure 7-13 : Horizontal and Vertical Field-of-view (source [DRI99])

7.9.2 FOV to Focal Length Conversion

In Chapter-6 paragraph 6-7-1, the focal length (*f*) was used to calculate the projected area of the missile optics. The focal length of the missile seeker optics can be calculated from the FOV accordingly. The F-number (F#) is defined as $F\# = \frac{f}{D_o}$ and also F# is equal to $F\# = \frac{1}{2 \tan \theta}$ [SCH95]. Where, θ is the half angle of the FOV. Therefore, by equating and rearranging the two expressions of F#, we can calculate the focal length from FOV as given in Equation 7-17.

$$f = \frac{1}{2} D_o \frac{1}{\tan \left(\frac{FOV}{2} \right)} \quad (7-17)$$

7.9.3 Modelling “NavigationInfo” Node for Missile Seeker View

The “*navigationInfo*” node contains information about the physical characteristics of the viewing model. The “*navigationInfo*” node controls the viewing characteristics of

the virtual world. The “*navigationInfo*” node contains the following fields [VRML97]:

- (a) **AvatarSize.** The “*avatarSize*” field specifies the viewers physical dimensions in the virtual world for the purpose of collision detection and terrain following. The avatar field has three values. The first defines the allowable distance between the viewer’s position and any other geometry before a collision is detected. The second value specifies the height above the terrain at which the browser should maintain the viewer. The third value is for the height of the tallest object over which the viewer can jump or step.
- (b) **Headlight.** The “*headlight*” is a default light source pointing in the direction the viewer is looking. This field can turn the “*headlight*” on or off. For emissive colour the “*headlight*” or any other light source is not required.
- (c) **Speed.** This field sets the rate at which the viewpoint travels through a scene.
- (d) **VisibilityLimit.** The “*VisibilityLimit*” field sets the farthest distance the viewpoint can see. A value of 0.0 indicates an infinite visibility.
- (e) **Set_bind.** The “*set_bind*” field is an event binding field to decide which “*navigationInfo*” node is active for the browser’s viewer. This feature is used when more than one “*navigationInfo*” node is present in the virtual world.

The default values for “*avatarSize*” field as given in VRML are [0.25, 1.6, 0.75] which represent the collision distance, height over the terrain and step size in meters respectively [VRML97]. With this value of avatar collision distance, during the missile target engagement simulation, the target was disappearing once the target-missile distance (R_{TMD}) was less than 925 meters. To overcome this problem, the collision distance was gradually reduced and finally the value for the collision distance (for which the target was visible till it reaches very close to the missile) was found. Although by changing the avatar collision distance to 0.0002 the problem of the target disappearing once it comes close to the missile was alleviated, however,

now the target was disappearing once the R_{TMD} was more than 1000 meters. This showed that in VRML there is a limit to see a fixed number of depth frames at any one time (number of vertical frames in to which the scene can be rendered in one go). To overcome this problem the algorithm is modified to change the “*avatarSize*” field value during simulation. For the situations when R_{TMD} is more than or equal to 1000 meters ($R_{TMD} \geq 1000$) the “*avatarSize*” field is kept to the default values as given in Equation 7-18.

$$myworld.navigationInfo.avatarSize = [0.25 \quad 1.6 \quad 0.75] \quad (7-18)$$

Whereas, once the R_{TMD} reduces below 1000 meters (i.e. $R_{TMD} < 1000$) the collision filed value is changed to 0.0002 as shown in Equation 7-19.

$$myworld.navigationInfo.avatarSize = [0.0002 \quad 1.6 \quad 0.75] \quad (7-19)$$

This is achieved by defining two “*navigationInfo*” nodes in the virtual world (each with different “*avatarSize*”) as shown in Equation 7-18 and 7-19, and then selecting the required values by activating the “*set_bind*” field of the desired node as per the R_{TMD} condition.

7.10 Modelling IR Flare Geometry and Aerodynamics

An IR flare is modelled as a small target, characterized by its physical and radiometric properties. The radiant characteristics of the flare are discussed in Chapter-6 paragraph 6-4. The following paragraphs explain the steps involved in modelling the flare pallet 3D geometry, the flare pallet ballistic trajectory, dispensation mechanism and flare IR plume 3D geometry and appearance in the 3D virtual world IR scene as per the radiant characteristics.

7.10.1 Inputs for Flare Modelling

For the modelling of 3D geometry and aerodynamics of the flare the following input data is required:

- (a) Mass of the flare (m_{flare}),
- (b) Flare ejection velocity (V_{eject}),
- (c) Flare eject angle (θ_{flare}),
- (d) Host platform Speed ($V_{aircraft}$),
- (e) Host platform location in global coordinates ($X_{aircraft}, Y_{aircraft}, Z_{aircraft}$),

- (f) Host platform direction at the time of flare ejection ($\theta_{aircraft}$),
- (g) Flare drag coefficient (C_D),
- (h) Atmospheric density (ρ_{air}),
- (i) Flare ejection time (t_{eject}),
- (j) Flare pallet dimensions (different for rectangular or circular pallet),
- (k) Acceleration due to gravity (g),
- (l) Flare dispensation sequences (salvo/burst count, salvo/burst interval).

7.10.2 Assumptions for Flare Modelling

Several assumptions are made for modelling the IR flare trajectory and geometry. These assumptions are made to achieve some degree of fidelity of the model as well as keep the model as simple as possible. However, the fidelity of the model may be increase by incorporating these features in the model:

- (a) **Mass of flare.** In actual flares, the mass of the pallet reduces as the flare burns. This affects the ballistic coefficient of the pallet trajectory. However, the mass of the pallet is considered to remain constant throughout the trajectory.
- (b) **Cross-sectional Area of Flare.** In practical cases, the cross-sectional area of the flare decreases as the pallet burns. Also the flare pallet rotates or spins once it comes in to contact with the air stream. This change in the cross-sectional area affects the drag produced by the pallet. However, for modelling purpose, this is assumed that through out the trajectory the flare maximum cross-sectional area is facing the air stream. The decrease in cross-sectional area due to pallet burning is also not considered.
- (c) **Drift in Trajectory due to Wind.** The flare trajectory may drift due to the blowing wind in the area. Or in the case of the aircraft or helicopter the turbulence caused by the engine exhaust and the propellers may disturb the flare trajectory. However, for modelling purpose it is assumed that the air is still or no wind is blowing and there is no turbulence in the air.

- (d) **Atmospheric Density.** The drag produced by the pallet depends upon the atmospheric density. The atmospheric density changes as the altitude changes. However, as the flare functional time or burn time is typically 3 to 5 seconds and in this time the change in the altitude may not be more than few hundred feet [POL93]. Therefore, for the duration of the flare flight, it is assumed that the atmospheric density remains constant. However, depending upon the altitude at which the flares are dispensed, the corresponding atmospheric density is used.
- (e) **Atmospheric Turbulence.** It is assumed that the atmosphere is homogenous and the affects due to atmospheric turbulence on the flare trajectory are not considered.
- (f) **Direction of Flare Ejection.** Ideally, two angles are required to model the flare ejection in any direction with respect to host aircraft. However, it is assumed that flare can only be ejected at any angle in a vertical plane along the LOS of the host aircraft. Or in other words flare can not be ejected diagonally or sideways.

7.10.3 Modelling Flare Pallet Geometry

Typically, the flare pallets are manufactured in three basic shapes. The square base cylinder, the rectangular base cylinder and the cylindrical rod with circular cross-sectional base. The typical dimensions of the square base flare pallet (e.g. MJU-50/B) are 1x1x8 inch (approximately 25x25x206 mm) [FAS07]. The rectangular-base pallet (eg. MJU-51/B) are typically 1x2x8 inch (approximately 25x50x206 mm) [FAS07]. The typical dimensions of a circular base pallet (eg. MJU-49/B) are diameter 1.4 inches (approximately 35 mm) and length 5.8 inches (approximately 147 mm) [FAS07]. Figure 7-14 shows the three typical shapes of flare pallets. The 3D geometry of the flare pallet is made by using the “*geometrical*” nodes of VRML. The square and rectangular base pallets are modelled using the “*box*” node of VRML [VRML97]. The input dimensions of a rectangular base pallet are given as in Equation 7-20.

$$myworld.flare_pallet.box = [length \quad width \quad height] \quad (7-20)$$

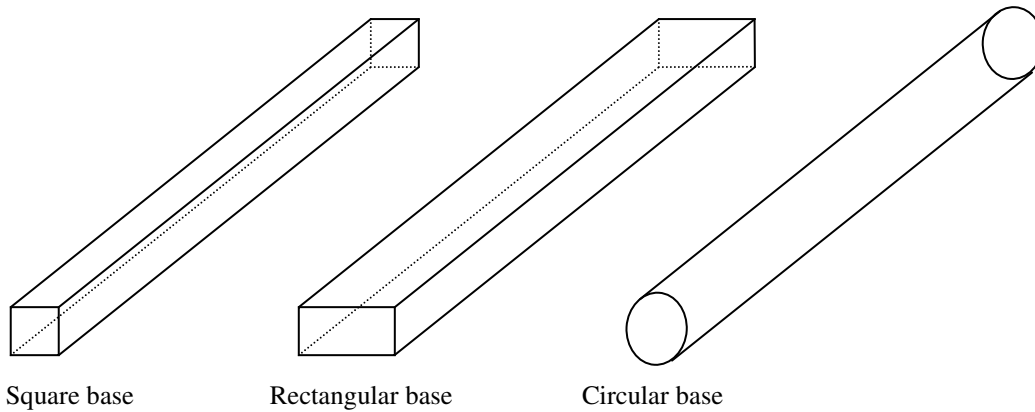


Figure 7-14 : Flare pallet 3D geometry

The circular base pallet is modelled using the “*cylinder*” node of VRML [VRML97]. The input dimensions of a circular base pallet are given in Equation 7-21.

$$myworld.flare_pallet.cylinder = [length \quad diameter] \quad (7-21)$$

The flare pallet is added in the main VRML file as an independent node. The flare “*translation*” and “*rotation*” fields are modelled as given in Equation 7-22 and 7-23 respectively. However, before ejection the flare pallet shares the same translation and rotation values as that of the target platform so that it remains inside the aircraft. To control the movement of more than one flare separately as per their dispensation sequences and trajectories, each flare pallet is given independent translation and rotation fields.

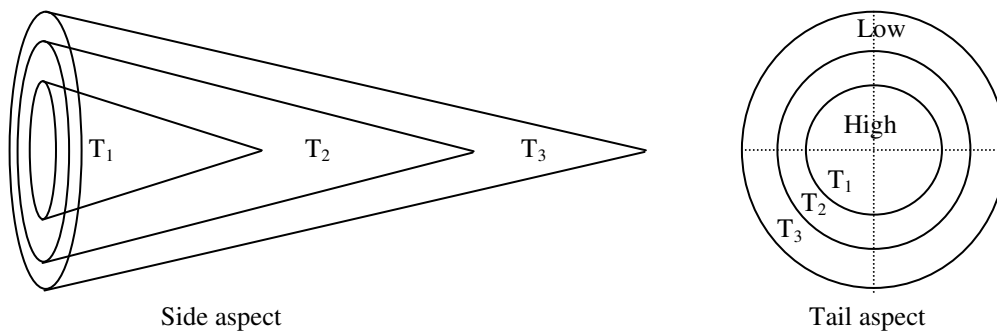
$$myworld.flare_1.translation = [X_{flare_1} \quad Y_{flare_1} \quad X_{flare_1}] \quad (7-22)$$

$$myworld.flare_1.rotation = [0 \quad 1 \quad 0 \quad \phi_{flare_1}] \quad (7-23)$$

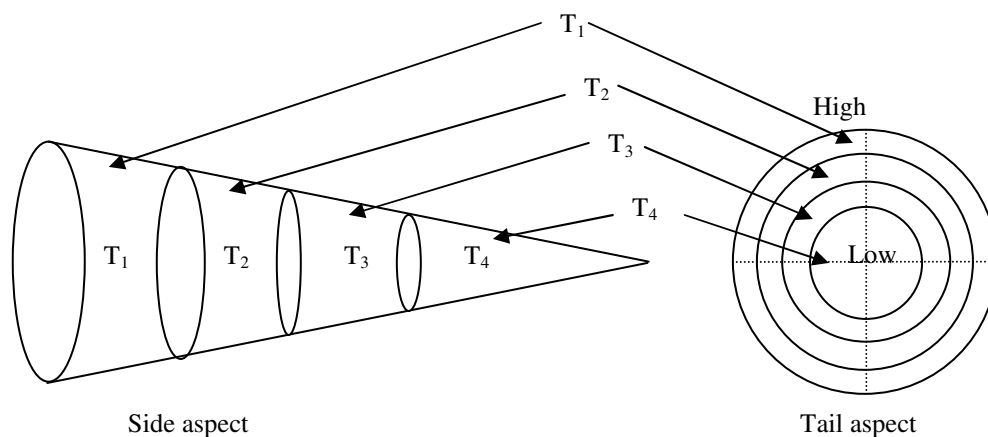
7.10.4 Modelling Flare IR Plume 3D Geometry

There are several different possibilities for modelling the flare plume geometry. If the flare is considered as a static thermal point source, this may be modelled as a set of co-centric spheres with varying intensity layers. However, as the flare is falling and

also moving with high speed, so the centre of intensity will shift towards the direction of the motion and the shape of the plume would stretch along the direction of motion. One option for modelling the flare plume 3D geometry may be as cones of different temperature/radiant intensities. Figure 7-15(a) shows the side and tail aspect of the flare plume modelled as co-centric cones. The inner layers represent higher temperature zones as compares to the outer layers. The outer layer is kept as semi-transparent to see the effects of all layers. The number of layers may be two or more depending upon the desired fidelity. Another option is to model the flare plume as a stack of circular discs with reducing diameter and reducing temperatures. Figure 7-15(b) shows the flare plume modelled as circular discs from side and tail aspects. Although from the side aspect the circular discs may represent the flare plume intensity variation reasonably, however, from the tail aspect this shows as rings of different temperatures with the outer ring as high temperature and the inner with



(a) Cone Shape plume 3D geometry



(b) Disc shape plume 3D geometry

Figure 7-15 : Flare plume 3D geometry

low temperature, which is not a true representation of the actual flare plume temperature profile. Whereas, on the other hand, the flare plume modelled as concentric cones of different temperature, as shown in Figure 7-15(a), models the temperature profile from all aspects (only side and tail aspects are shown in Figure 7-15(a)).

7.10.5 Modelling Flare Dispenser

The modelling of the flare dispenser involves three steps: first selecting a location on the host platform to place the dispenser unit; second, modelling the flare ejection speed and direction; third, the dispensation sequences. The location of the flare dispenser may be any place under the wings or fuselage. In the case of large aircraft, there may be more than one dispenser placed at different locations. The dispenser geometry is placed as a child node to the aircraft body. Ideally, two angles are required to model the flare ejection in any direction all around the aircraft. Figure 7-16 shows these two angles. The angle α controls the direction of the flare ejection in a lateral plane (XZ-plane). To cover all around, the angle α may hold any value from

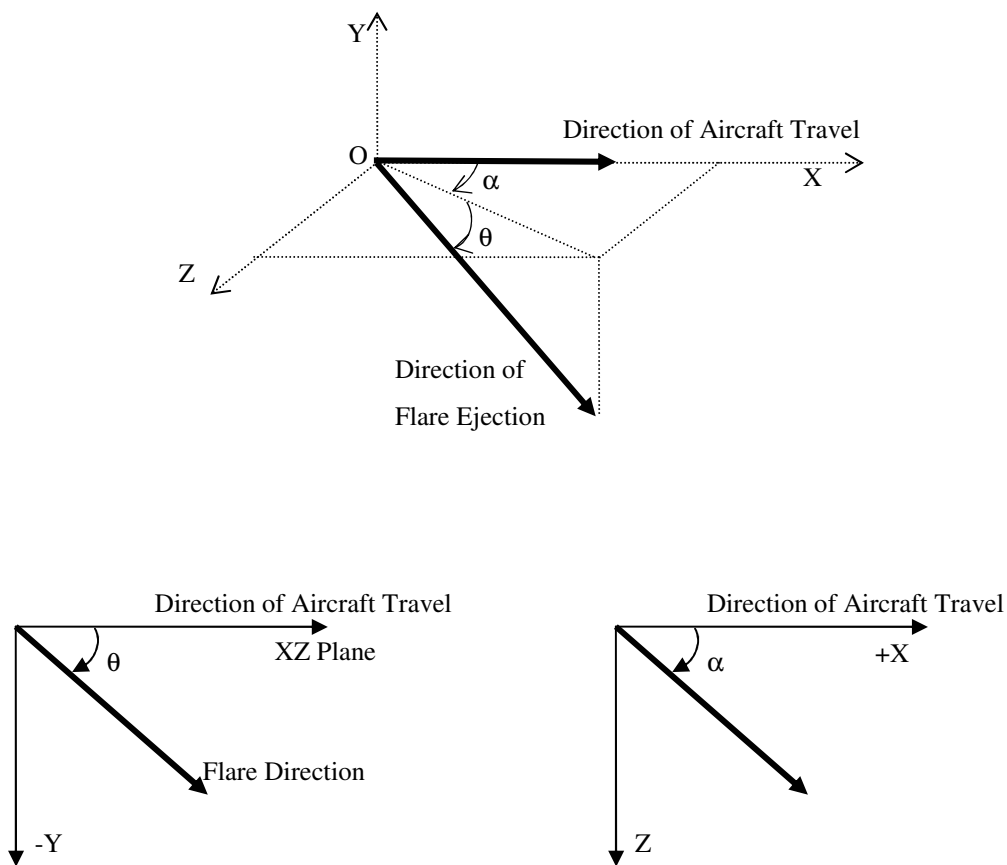


Figure 7-16 : Modelling Flare ejection direction

zero to $\pm 180^\circ$. The zero is in the direction of the aircraft travel. Now the second angle θ is the elevation or depression angle which the flare makes with XZ-plane. Angle θ may be any value between zero and $\pm 90^\circ$. The positive θ shows the ejection angle down the LOS of the aircraft plane. Presently, only angle θ is used for modelling the flare ejection direction. The angle α is kept as zero which means the flare can only be dispensed in a vertical plane along the direction of travel of the aircraft.

7.10.5.1 Multiple Flare Dispensation Sequence

During real missile-target engagement scenarios, the target aircraft may eject multiple flares to counter against IR heat seeking missile threats. The multiple dispensations mean more than one flare may be ejected simultaneously or with some time intervals. The following four variables model the multiple flare dispensation sequences:

- (a) **Salvo Count.** Salvo count controls the number of flares to be fired simultaneously or with some short interval.
- (b) **Salvo Interval.** Salvo interval decides the time interval between dispensations of two flares. To dispense more than one flare simultaneously the salvo interval may be selected as zero or a very small value such as a fraction of a second.
- (c) **Burst Count.** The burst count describes the number of times the same sequence as described by the salvo count and salvo interval is to be repeated.
- (d) **Burst Interval.** Burst interval controls the time between two burst of flares.

Presently, one dispenser is modelled under the fuselage which can eject a maximum of four flares. The number of dispensers and the flares may be increased by adding similar geometries in the VRML file. By adding one more dispenser unit the dispensation of two flares simultaneously in two different directions may be modelled.

7.10.6 Modelling Flare Plume Appearance

The IR signature of the flare plume may be modelled in the virtual world by controlling the values of material fields of the plume geometry as per the thermal

characteristic of the flare. This may be done by selecting the “*emissiveColour*” field of each layer of the cone as per the corresponding radiance values calculated in Chapter-6 paragraph 6-4. In general, for a plume with “*n*” number of layers, the radiance of all the layers is stored in a vector as given in Equation 7-24.

$$N_{flare} = [N_{layer_1}, N_{layer_2}, N_{layer_3}, N_{layer_4}, \dots, N_{layer_n}] \quad (7-24)$$

The radiance values are normalized over the range of the colour-index. The normalized radiance of all the layers is calculated as given in Equation 7-25.

$$N_{flare_{norm}} = \frac{N_{flare}}{N_{max}} colour_index \quad (7-25)$$

Where, $N_{flare_{norm}}$ is the normalized radiance of all the layers of the flare plume,

N_{max} is the maximum value of radiance available in the scene,

$Colour_index$ is the number of indexed RGB colours in the colour-map.

The normalized flare radiance of each layer is then converted into the corresponding RGB colour out of the selected colour-map as given in Equation 7-26.

$$Color_{flare} = \begin{bmatrix} red_{layer1} & green_{layer1} & blue_{layer1} \\ red_{layer2} & green_{layer2} & blue_{layer2} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ red_{layer_n} & green_{layer_n} & blue_{layer_n} \end{bmatrix} \quad (7-26)$$

In the case of multiple flares, the colour of each flare may be calculated in the same manner. The radiance corresponds to the “*emissiveColour*” field in VRML. The “*specularColour*” and “*diffusedColour*” fields are kept blank as the reflectivity is considered zero. The outer layers may be considered as semi-transparent so that the effects of all layers could be seen. Presently, the flare plume is modelled as two layers of co-centric cones.

7.10.7 Modelling Flare Trajectory in 3D

The trajectory of the flare pallet ejected in to the air stream is governed by the simple physics of the projectile motion. The flare pallet is ejected at a certain velocity and orientation with respect to the flying host aircraft. After leaving the aircraft, the conventional flare pallet follows the ballistic trajectory. However, in the case of a self-propelled or kinematic flare, the pallet is placed inside an aerodynamic shell with fins and a nozzle which generate lift and thrust and the pallet follows a streamline aerodynamic path. The flare trajectory modelling deals with the aerodynamic model of the pallet based on the gravity and the drag force acting on the pallet surface area. In the case of the self-propelled flares, the lift and thrust effects are also to be considered. To model the flare trajectory in 3D the velocity, acceleration and the instantaneous position of the flare pallet at any one time frame has to be calculated. For animation, these values need to be recalculated at every frame or time interval. The following are the steps involved in the modelling of the flare trajectory in three dimensions.

7.10.7.1 Atmospheric Density, Reference Area and Drag Coefficient

For the detailed trajectory calculations, the precise variations of the standard atmosphere have to be considered. The density of atmosphere changes with the change in the altitude. At sea level and 20°C dry air has a density of 1.225 kg/m³. As altitude increases the air density (ρ_{air}) decreases. A tabular representation of the relationship of altitude, pressure, temperature and density based on the US Standard Atmosphere (1976) is shown in Table 7-2 [DRI99]. The atmospheric density must be taken into account when calculating the flare drag. For the calculation of drag, the altitude at which the flare is dispensed, the corresponding value of atmospheric density may be used as shown in the Table 7-2.

The reference area (A_{ref}) for the drag calculation may be taken as the biggest cross-sectional area of the pallet surfaces facing the air stream. For the rectangular base pallet the area is found by taking the product of the length and the width as given in Equation 7-27. The bigger side of the pallet base is taken for A_{ref} calculations.

Table 7-2: Atmospheric Parameters as function of height (source [DRI99])

HEIGHT (km)	PRESSURE (hPa)	TEMPERATURE (K)	DENSITY (kg/m ³)
0	1.013E+3	288	1.225
2	7.950E+2	275	1.007
4	6.166E+2	262	0.8194
6	4.722E+2	249	0.6601
8	3.565E+2	236	0.5258
10	2.650E+2	223	0.4135
12	1.940E+2	217	0.3119
14	1.417E+2	217	0.2279
16	1.035E+2	217	0.1665
18	7.565E+1	217	0.1217
20	5.529E+2	217	0.08891

$$A_{ref} = length \cdot width \quad (7-27)$$

Whereas, in case of the circular base pallet the reference area maybe approximated as the product of the length and the diameter of the base as given in Equation 7-28.

$$A_{ref} = length \cdot diameter \quad (7-28)$$

Keeping the reference area the same, the shape of an object has a very great effect on the amount of drag which can be produced. This effect is catered for by the drag coefficient (C_D). The drag coefficient is a number which is used to model all of the complex dependence of drag on shape, inclination and some flow conditions [AND00]. The drag coefficient of an object is determined experimentally. A flat plate gives the highest drag coefficient and a stream-lined symmetric airfoil produces the lowest drag. Shape has a very large effect on the amount of drag produced. Figure 7-17 shows the shape effect on the drag. The typical values of a flat plate facing the air stream may be as high as 1.28, and that of a bullet may be 0.295 and an aerodynamic airfoil may have a very low value of 0.045 [NASA07]. The drag coefficient of the conventional flare may be very high as the pallet shape is not aerodynamic. However, for self-propelled aerodynamic shape flares, the drag coefficient may be a low number.



The shape of an object has a very great effect on the amount of drag.

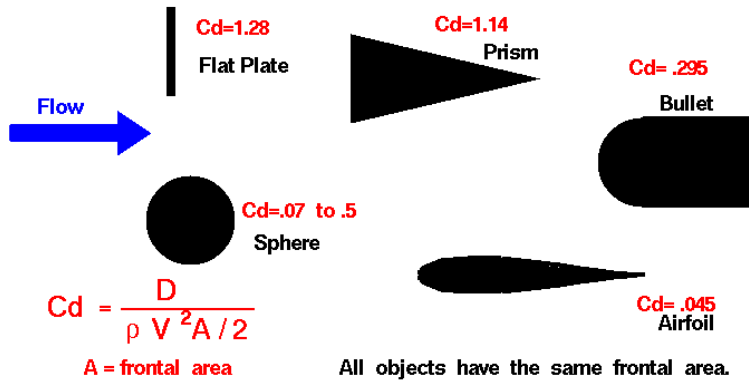


Figure 7-17 : Drag Coefficient of different shapes (source [NASA07])

7.10.7.2 Flare Velocity Components

The flare is dispensed from the host platform (aircraft) with some velocity and direction. The total flare velocity (V_{flare}) at the time of leaving the aircraft is the sum of the aircraft velocity ($V_{aircraft}$) and the flare eject velocity (V_{eject}). The flare eject-velocity (V_{eject}) as shown in Figure 7-18 can be represented as the three components along the three axes of the aircraft. The three components of V_{eject} are given in Equations 7-29 to 7-31.

$$V_{eject_{LOS}} = V_{eject} \cos \theta_{flare} \quad (7-29)$$

$$V_{eject_{down}} = -V_{eject} \sin \theta_{flare} \quad (7-30)$$

$$V_{eject_{lateral}} = 0 \quad (7-31)$$

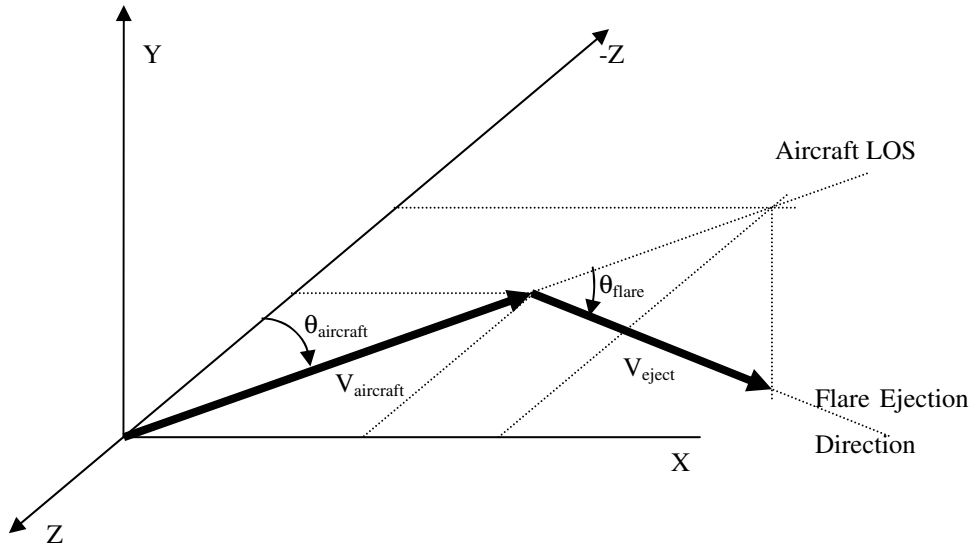


Figure 7-18 : Aircraft and flare velocity Vectors in 3D world coordinates

The V_{eject_LOS} is the component of V_{eject} along the LOS of the aircraft. The V_{eject_down} is the component of V_{eject} in the vertically down direction. The third component is the $V_{eject_lateral}$ along the lateral axis of the aircraft. Although, the lateral component is zero as the flare can only be ejected in a plane along the LOS of the aircraft. The three components of the V_{eject} given in Equation 7-29 to 7-31 are then transformed to the world coordinates (X, Y, Z) of the virtual world. Figure 7-19 shows the three components of V_{eject} transformed into world coordinates. Equation 7-32 to 7-34 gives the three components of V_{eject} in world coordinates.

$$V_{eject_x} = -V_{eject_LOS} \sin \theta_{aircraft} \quad (7-32)$$

$$V_{eject_y} = V_{eject_down} \quad (7-33)$$

$$V_{eject_z} = -V_{eject_LOS} \cos \theta_{aircraft} \quad (7-34)$$

Similarly, the aircraft velocity ($V_{aircraft}$) is converted into world coordinates. The three components of $V_{aircraft}$ are given in Equation 7-35 to 7-37 and are shown in Figure 7-19. In Equation 7-36 the component of $V_{aircraft}$ along the Y-axis is kept as

zero. This is because the ROD of the aircraft is not considered for the flare trajectory calculations.

$$V_{aircraft_x} = -V_{aircraft} \sin \theta_{aircraft} \quad (7-35)$$

$$V_{aircraft_y} = 0 \quad (7-36)$$

$$V_{aircraft_z} = -V_{aircraft} \cos \theta_{aircraft} \quad (7-37)$$

Now by adding the corresponding X-, Y-, and Z-components of the V_{eject} and $V_{aircraft}$ from Equation 7-32 to 7-34 and Equation 7-35 to 7-37 respectively, the total flare velocity V_{flare} is calculated as given in Equation 7-38 to 7-40.

$$V_{flare_x} = V_{aircraft_x} + V_{eject_x} \quad (7-38)$$

$$V_{flare_y} = V_{aircraft_y} + V_{eject_y} \quad (7-39)$$

$$V_{flare_z} = V_{aircraft_z} + V_{eject_z} \quad (7-40)$$

The magnitude of V_{flare} is calculated using Equation 7-41.

$$V_{flare} = \sqrt{V_{flare_x}^2 + V_{flare_y}^2 + V_{flare_z}^2} \quad (7-41)$$

The reason for calculating the flare velocity in world coordinates is that after ejection from the aircraft the flare moves independent of the aircraft movement. To model the flare trajectory independent of the aircraft motion, the flare “*translation*” and “*rotation*” fields are to be changed as per the world coordinates.

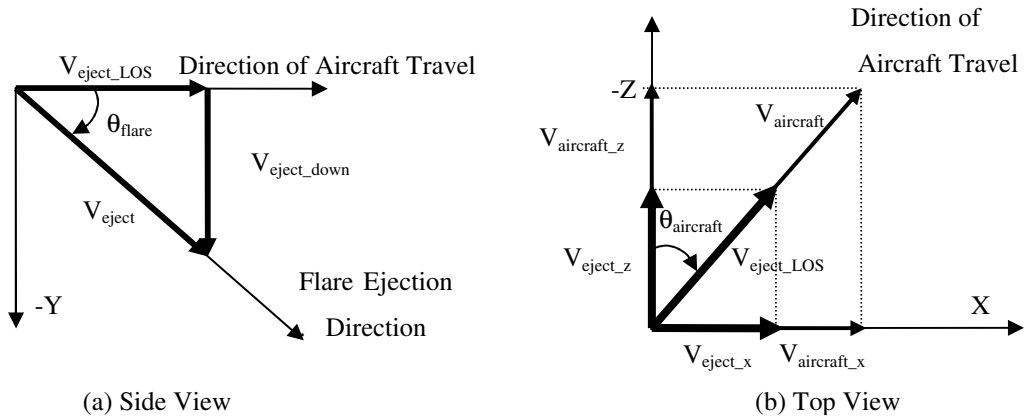


Figure 7-19 : Aircraft and Flare velocity components

7.10.7.3 Flare Acceleration Components

For the conventional flare the only acceleration is in the downward direction (negative Y-axis) due to the gravitational force. The deceleration due to drag may have three components along X-, Y- and Z- axes. Whereas, for the self-propelled aerodynamic flares the thrust and lift may also contribute in the overall flare acceleration. The sum of all the forces acting on the flare along the three axes can be written as given in Equation 7-42 to 7-44.

$$\sum F_{flare_x} = m_{flare} \cdot a_{flare_x} = -D_{flare} \cdot V_{flare} V_{flare_x} \quad (7-42)$$

$$\sum F_{flare_y} = m_{flare} \cdot a_{flare_y} = -m_{flare} \cdot g - D_{flare} \cdot V_{flare} V_{flare_y} \quad (7-43)$$

$$\sum F_{flare_z} = m_{flare} \cdot a_{flare_z} = -D_{flare} \cdot V_{flare} V_{flare_z} \quad (7-44)$$

Where, F_{flare} is the force applying on the flare,

m_{flare} is the mass of the flare pallet,

a_{flare} is the acceleration of the flare,

V_{flare} is the magnitude of the total flare velocity as calculated in Equation 7-41,

V_{flare_x} is the X-component of the flare velocity as calculated in Equation 7-38,

V_{flare_y} is the Y-component of the flare velocity as calculated in Equation 7-39,

V_{flare_z} is the Z-component of the flare velocity as calculated in Equation 7-40,

$D_{flare} = \frac{1}{2} \rho_{air} C_D A_{ref}$ is the flare drag,

ρ_{air} is the density of the air,

C_D is the drag coefficient for the pallet shape, and

A_{ref} is the reference area of the flare pallet.

Rearranging Equation 7-42 to 7-44, the corresponding flare acceleration components along the three axes are given as Equation 7-45 to 7-47.

$$a_{flare_x} = -\left(\frac{D_{flare}}{m_{flare}}\right)V_{flare}V_{flare_x} \quad (7-45)$$

$$a_{flare_y} = g - \left(\frac{D_{flare}}{m_{flare}}\right)V_{flare}V_{flare_y} \quad (7-46)$$

$$a_{flare_z} = -\left(\frac{D_{flare}}{m_{flare}}\right)V_{flare}V_{flare_z} \quad (7-47)$$

Now the V_{flare} after the time interval (Δt) is calculated using the flare velocity and acceleration components calculated in Equations 7-38 to 7-40 and 7-45 to 7-47 respectively. As the time interval Δt corresponds to the frame rate (i.e. $\Delta t = 1/\text{frame-rate}$), the flare velocity after Δt may be expressed as V_{flare} at “ $n+1$ ” frame.

$$V_{flare_x}(n+1) = V_{flare_x}(n) + a_{flare_x}(n)\Delta t \quad (7-48)$$

$$V_{flare_y}(n+1) = V_{flare_y}(n) + a_{flare_y}(n)\Delta t \quad (7-49)$$

$$V_{flare_z}(n+1) = V_{flare_z}(n) + a_{flare_z}(n)\Delta t \quad (7-50)$$

7.10.7.4 Instantaneous Position of the Flare

The instantaneous position of the flare pallet in the world coordinates after Δt may be calculated as given in Equation 7-51 to 7-53.

$$X_{flare}(n+1) = X_{flare}(n) + V_{flare_x}(n)\Delta t + \frac{1}{2}a_{flare_x}(n)\Delta t^2 \quad (7-51)$$

$$Y_{flare}(n+1) = Y_{flare}(n) + V_{flare_y}(n)\Delta t + \frac{1}{2}a_{flare_y}(n)\Delta t^2 \quad (7-52)$$

$$Z_{flare}(n+1) = Z_{flare}(n) + V_{flare_z}(n)\Delta t + \frac{1}{2}a_{flare_z}(n)\Delta t^2 \quad (7-53)$$

Finally, the position and orientation of the flare pallet is expressed in the world coordinates as the “translation” and “rotation” fields given in Equation 7-54 and 7-55 respectively.

$$myworld.flare_1.translation = [X_{flare_1} \quad Y_{flare_1} \quad Z_{flare_1}] \quad (7-54)$$

$$myworld.flare_1.rotation = [0 \quad 1 \quad 0 \quad \phi_{flare_1}] \quad (7-55)$$

Where, ϕ_{flare_1} represents the direction of the flare in world coordinates at the time of leaving the aircraft. ϕ_{flare_1} is taken as the aircraft angle $\theta_{aircraft}$ at the time of the ejection. In the case of more than one flare, the translation and rotation field of every flare has to be calculated independently in the same way.

7.11 VRML vs IR Comparison Summary

The various “nodes” and “fields” of VRML may be used for modelling the IR signature scene. These features are discussed in detail in this chapter. Table 7-3 summarizes the features of the IR signature modelling and the corresponding nodes and fields of VRML which may be used to model these features. Also an extended list of IR parameters and the corresponding VRML properties which may be explored to model these features in virtual reality is attached as Annex “II” to this thesis.

Table 7-3 : Summary of VRML nodes and fields used for IR signature scene modelling

IR Signature Scene	VRML Nodes and Fields	Remarks
Objects location and motion	“translation” and “rotation” fields	Movement of objects in a scene may be controlled by using translation and rotation fields
Shapes of objects	“geometry” nodes	Any 3D shape may be modelled using different geometrical shapes
Size of objects	“scale”	Size of an objects may be chaged as per requirements
Objects with same radiometric properties	“DFE” and “USE” node	Plan one “DFE” node (short of “define” to define a name for a node) in the start and call it for different objects by “USE” node
Landmarks, hills (terrain)	“elevationGrid” node	Irregular terrain may be modelled using elevation grid
Sky and ground as background	“background” node	Multi-layer sky and uniform ground may be modelled
Temperature and emissivity	“emissiveColour” field	Radiance of objects may be modelled as fake colours
Reflectivity (sun shine, earth shine, sky shine)	“ambientIntensity” “shininess” “specularColour” and light source	The leading-edge reflections and cold-sky reflections may be modelled
Transmissivity	“transparency”, “emissiveColour” and background colour	Exhaust gas plume, clouds, smoke etc. may be modelled using transparency
Atmospheric attenuation	“fog” (exponential) node	Atmospheric attenuation may be modelled using exponential fog
Seeker FOV	“FOV” field of viewpoint node	The viewpoint FOV may be used to modelled vertical FOV of the seeker optics
Missile target collision	“collision” node and “avatarSize” field	Hit criterion may be modelled using collision node and avatar size of “NavigationInfor” node
Missile-LOS view, launcher view, seeker head view	“viewpoint”	Multiple viewpoints may be modelled to analyse the simulation from different aspects

7.12 Conclusion

An effort is made to visualize the IR scene of the missile-target engagement sequences using three-dimensional virtual reality modelling and animation techniques. The IR scene represents high-fidelity, physically accurate radiance images of targets, flares and backgrounds in operational scenarios at wavelengths between 1 to 15 microns. The normalized radiance of the objects is used to render the IR scene as fake colours. The radiometric properties of sub-targets are modelled as “*shininess*” and “*transparency*” in the virtual world. The commercially available 3D VRML models of military targets are modified to model IR signature. The aircraft exhaust gas plume, leading-edges, flare pallet and flare plume geometry are modelled using virtual reality tools. The IR background is modelled using “*background*” feature of the VRML. The missile gimbaled seeker head is modelled and integrated with the missile tracking and guidance algorithm to control the movement of the missile. The flare ballistic trajectory with drag effects is modelled in the 3D virtual world.