

Command Guidance Missile Tracking Algorithms Evaluation Based on Visual Simulation

A. R El-gabri, A. N Ouda, Ahmed S. Ahmed, Yehia Z. Elhalwagy, Ahmed Khamis
Department of Guidance and Control, Military Technical College
Cairo, Egypt

Abstract— Visual simulation has proven to be a conclusive part of any system development cycle. visual simulation first necessitates that a model to be developed to represent the characteristics of the selected physical system or process. The model represents the system itself but in a simpler manner to avoid the complexity and to be trackable, while the simulation represents the operation of the system over time. This paper uses 3D visualization to model real-life situations to observe how different tracking algorithms react. By changing the simulation conditions, the real system behavior is able to be predicted. The Simulation involves both missile and target objects which are created by Multigen-Creator, 6-DOF (degree of freedom) mathematical model of the missile to derive its dynamical parameter during its flight trajectory implemented with Visual C++ all integrated in an infrared simulation environment which is implemented using Vega and finally the tracking algorithm needed to be evaluated.

Keywords— Visual C++; Vega; Multigen-Creator; Texture Material Mapper; MOSART Atmospheric Tool; 6DoF;

I. INTRODUCTION

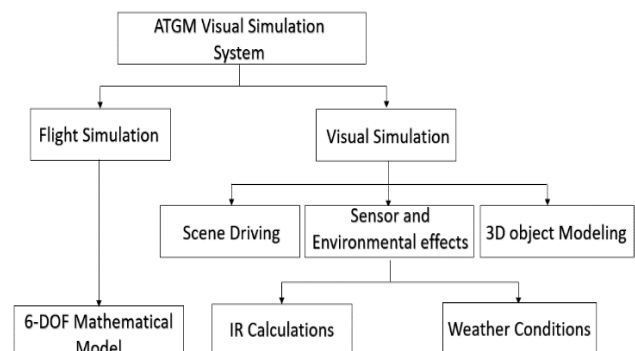
Antitank Guided missiles (ATGM) are one of the basic weapons in the modern war. ATGM missiles are command guidance systems launched against tanks and armored vehicles. These missiles are classified into three generations; the first generation in which both the target and missile are manually tracked using optical telescopes. The second generation in which the target is manually tracked using optical telescopes while the missile is automatically tracked by including an infrared sensor in the launcher with the telescope to detect the IR radiation from a source strapped on the missile rear, Then the motion parameters are transferred automatically to signals applied to the guidance unit. The third generation is characterized by manual or automatic target tracking through optical telescopes, TV, laser or radio devices and the missile is automatically tracked as in the second generation. However, the transmission of guidance commands to the missile in this generation are accomplished through a remote link instead of wires [1]. Building ATGM visual simulation system gives the flexibility to the researchers and system developer to evaluate tracking algorithm performance against different real scenarios [2]. ATGM visual Simulation is used to earn an insight vision into their functionality to display the real effects of alternate situations and action courses. ATGM visual simulation also used when the real system cannot be involved, because it might not be available, or it may be unsafe or undesirable to involve, or it may be designed but not built yet, or it may not completely exist yet [3,4]. The visual simulation usually focuses on either the motion simulation or the visual scene performance. The former is obtained using opengl Library, but with the drawbacks of the code complexity, long development cycle while the latter is obtained using Vega which supports a

quicker development cycle and better visualization, but with the drawbacks of the weakness of the motion simulation [5].

The paper organization is as follows. In section II, begins with a summery describing the flight and visual simulation phases and the steps needed to construct a complete ATGM visual simulation system, then a detailed discussion of flight simulation phase describes ATGM six degree-of-freedom (6-DoF) mathematical model which is solved by Ouda in his work [6-9,13]. Different coordinate reference frames are introduced with their mutual relation to describe the flying vehicle(missile) position and orientation in addition to forces and moments and ends with a detailed discussion of visual simulation phase introduces 3D object modeling and model optimization, Sensor and environmental conditions and scene driving of ATGM till the interception with the target In Section III, introduces experimental results of two different tacking algorithms. In Section IV, the conclusion of this paper.

II. MATERIAL AND METHOD

In this section, the development cycle of ATGM visual Simulation System mentioned in this paper has two main distinct phases, the visual simulation phase and flight simulation phase [10,11]. when integrating these phases with each other, a completed visual simulation system will be obtained. Figure 1 shows main structure of ATGM visual simulation system.



1. main structure of ATGM simulation system.

to increase development speed, assure system quality and improve system maintenance, Creator and Vega from MultiGen-Paradigm are used to visual simulation application [12]. Flight simulation model calculations are solved using visual studio c++ which develops a real-time controller for ATGM flight simulation model in Vega during its trajectory path till the interception with the target. The following steps describes the actual development process of ATGM visual simulation:

- 1) Multigen-Creator is used for 3D modeling target, ATGM.
- 2) Software Creator is used to create terrain texture from the source data (satellite images for the terrain) converted to creator file format using Creator terrain tools.
- 3) Performer Loader is used to load Openflight file into Vega.
- 4) initial Configuration, special effects, environmental configurations run and rendering of flight scene are through the call for .adf file in Vega.
- 5) Writing custom code to create a standalone Vega application within Vega development environment.
- 6) Parameters of ATGM in the virtual environment simulation, such as velocity, the attitude of the missile, acceleration and so on, are solved using visual studio c++.

Loading of the Visual simulation application within the MFC framework through Vega API to provide ATGM visual simulation system with graphical user interface to control simulation parameters. Figure 2 shows the actual development process of ATGM visual simulation.

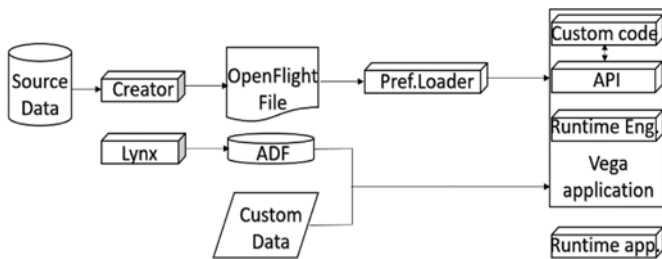


Fig. 2. actual development process of ATGM visual simulation.

A. Flight Simulation Phase

The basic subsystems that form a guidance system which directs the missile according to a specific guidance law are briefly described. The coordinate systems and reference frames have to be considered at the formulation and solving problems of flying vehicles to correctly describe the various dynamical parameters including position, velocity, forces, acceleration, and moments. The acting forces on the missile (flying vehicle) are weight, aerodynamic, and thrust forces. These forces have different reference frames and accordingly coordinates transformation is crucial. This frames transformation is done using Euler's angles transformation method [13]. Figure 3 displays an example for ground and board coordinates relation.

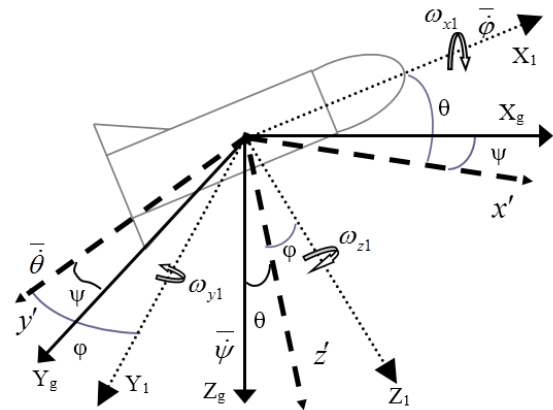


Fig. 3. Ground and Board Axes

Using Euler's angles, the transformation from body into ground coordinate system can be solved using the following transformation matrix:

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = T_{bg} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta\phi & s\psi c\theta + c\psi s\theta\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta\phi & -s\psi c\theta + s\psi s\theta\phi \\ -s\theta & s\theta\phi & c\theta\phi \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (1)$$

Where $X_1, Y_1,$ and Z_1 (X_g, Y_g and Z_g) are the components of the vector over the board (ground) reference axes.

Using Euler's angles, the transformation from body into ground coordinate system and from velocity into ground coordinates can be solved as previous transformation. The equations of ATGM motion include translational rotational equations. The simulation uses these nonlinear coupled differential equations that describe the rigid missile behavior can be summarized as follows [13]:

Force components in the velocity reference frame:

$$\begin{aligned} \sum F_x &= F_{Ax} + mg_x + F_{Tx} = m(\dot{v}_x) \\ \sum F_y &= F_{Ay} + mg_y + F_{Ty} = m(v_x \Omega_z) \\ \sum F_z &= F_{Az} + mg_z + F_{Tz} = -m v_M \Omega_y \end{aligned} \quad (2)$$

Moment components in body reference frame:

$$\begin{aligned} M_{x_1} &= (I_{zz} - I_{yy})\omega_{z_1}\omega_{y_1} + I_{xx}\dot{\omega}_{x_1} \\ M_{y_1} &= (I_{xx} - I_{zz})\omega_{z_1}\omega_{x_1} + I_{yy}\dot{\omega}_{y_1} \\ M_{z_1} &= (I_{yy} - I_{xx})\omega_{x_1}\omega_{y_1} + I_{zz}\dot{\omega}_{z_1} \end{aligned} \quad (3)$$

The flight simulation phase contains the previous equations that describes ATGM 6-DoF mathematical model. The flight simulation phase is designed in a modular structure. Each module in the structure describes a really current subsystem. Figure 4 shows a simplified modular structure to represent ATGM 6-DoF mathematical model.

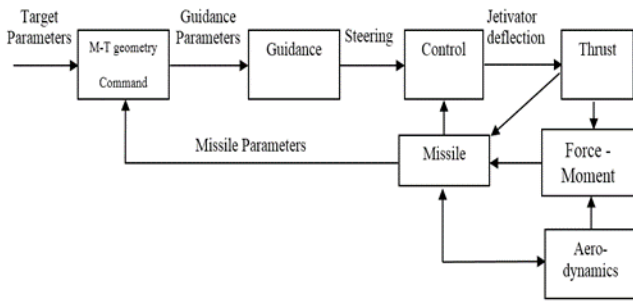


Fig. 4. Simple block diagram of main simulation modules.

B. Visual Simulation Phase

Using both Vega and Creator complete the visual simulation phase and provide real-time scene environment. To realize the virtual reality simulation platform, MultiGen-Paradigm Creator modelling software is used for establishing the model library and generating the entity model file which is saved in MultiGen-Paradigm's OpenFlight (.flt) database, which is the leading file format for major real-time application systems [14]. Using Vega simulation software package to configure *.adf file which describes the OpenFlight database used creating object models, explosions, flames, smoke and many more special effects and environmental effects, and other functionality; finally, use Visual C++ integrating simulation platform with the calculated missile position, velocity, acceleration from earlier step and generate application program. Visual simulation phase is constructed as follow:

1) *3D Object Modelling Using Multigen-Creator:* MultiGen-Creator is a tool that aids producing an efficient 3D models and terrain objects for interactive real time applications. Figure 5(a, b) shows both missile and target objects and database hierarchy in flt file formats.

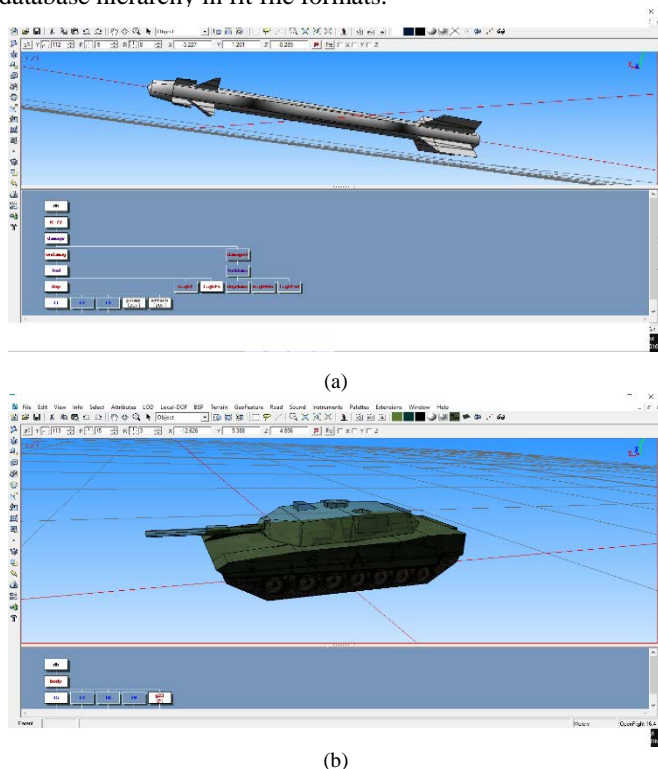


Fig. 5. (a) Second generation ATGM object, (b) Target object

The structure of an improved OpenFlight database is generally organized as follow:

- Database Nodes (DB Nodes)

The entire database description is included in these nodes. Only one node must be located in the top in the whole structure.

- Master Groups

The whole model is represented by these (g1 or g2). These nodes are eventually attached by the component nodes are. MultiGen-Creator offers g1 and g2 automatically.

- Significant Groups

One main database-component is described by each group. ATGM model for example has such important groups of nodes as the head, the subject, the wing, tail vents and the rear wing, etc. Other significant groups may lie under one significant group in complex models.

- Objects

The object nodes contain only some face nodes and simply describe parts without movements.

- Faces

These nodes are distinguished because of the color similarity with the represented faces. These nodes can be connected to, object nodes, group nodes, and the other face nodes.

- Vertexes

These nodes are the face properties features and represent the vertex.

Figure 6 shows the structure of an OpenFlight database

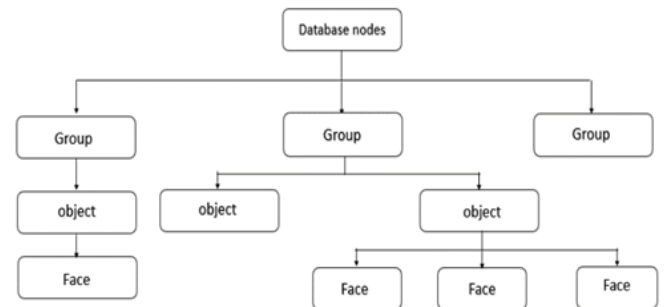


Fig. 6. OpenFlight File Structure

After the creation of 3D models, it must correctly be displayed in real-time system. The importance of a model optimization process in the design and development process because better real-time display can be obtained from an optimized model [15]. Following optimization methods are done for 3D models in this paper:

- Removing excessive triangles and polygons

If in one scene excessive triangles and polygons are existed, the memory may be exhausted. This problem should be optimized in the beginning of design.

- Levels of details (LODs)

LODs are used to rise the performance of the viewing. These are sets of the identical model with varying polygons number. The closer the eye point becomes; the more detailed versions of the model are substituted. As the polygons number increases, The LOD increases and presented when the model is closer to the eye point in runtime. Not much details needed when the eye point moves far from the model and lower LOD is substituted.

- Using texture mapping

The process of assign a texture pattern onto one or more polygons is called texture mapping. to store a mapped texture and its coordinates a Texture Mapping palette is used. This texture mapping can be applied quickly to many faces. This method is suitable when there is a large area to cover with an explicit texture. In exchange of using a Put Texture tool with each face selected to pick the coordinates for texture mapping, which for large databases, can be inefficient process. Texture choosing in Texture Mapping palette with its faces and defined alignment on which to apply can be selected.

- Using instances

A geometry reference in database file is defined as an instance and is used to avoid complexity and memory occupation if more geometry exists. Only a copy of an object's geometry is saved in memory. Instances look like various clones of a geometric solid. These instances with same properties except space position. The advantages of using instances are that we can save disk space, save time creating new models, and save time editing all occurrences of a model. A possible disadvantage is to have multiple objects that look exactly the same. This may not be noticeable in large, complicated databases but not in small databases which needed to be more flexible.

- Creating Billboards

Trees, poles, or people (Symmetrical objects) can be presented by single polygons that are called Billboards. Texture to billboards can be applied to form a realistic object with a small polygon count. Billboards turn around to face the eye point in the run-time loop system, so that the model's image always visible.

- External references

A geometry reference in another database is defined as an external reference. With external references, the polygons number can be reduced in large databases. They are helpful in big databases that in different areas use the identical geometry. For external reference to be created, an external reference node is placed in the database hierarchy, then the directory path and file name of the other database is assigned to the node. the directory path is used by the system to load each external reference. Generally, when the program is loaded all objects would be loaded if they exist in the main file and only objects which fall into sight are loaded rather all objects when using external references [16].

2) *Sensor and Environmental effects*: To evaluate different tracking algorithms, add varieties of test facilities and makes the scene environment looks real at different conditions, weather and IR sensor imperfections are added into virtual environment as follow:

- Weather conditions configuration

Figure 7(a, b, c, d) shows some of weather conditions

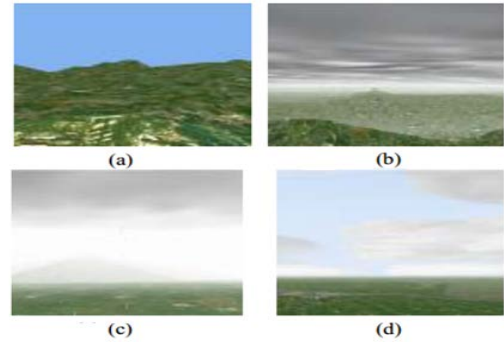


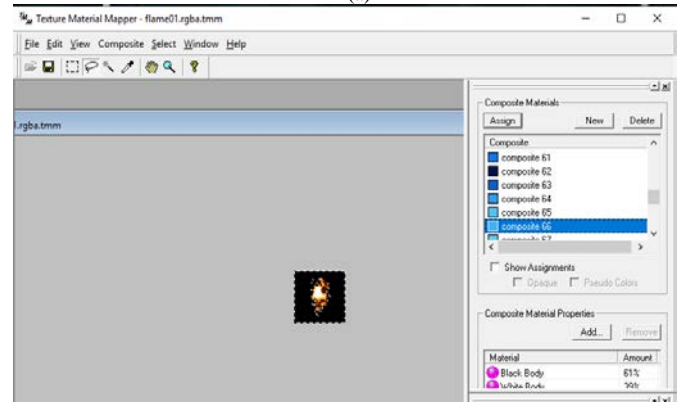
Fig. 7. (a) Clear Day, (b) Rainy Day, (c) Foggy Day, (d) Cloudy Day

- IR Scene Calculations

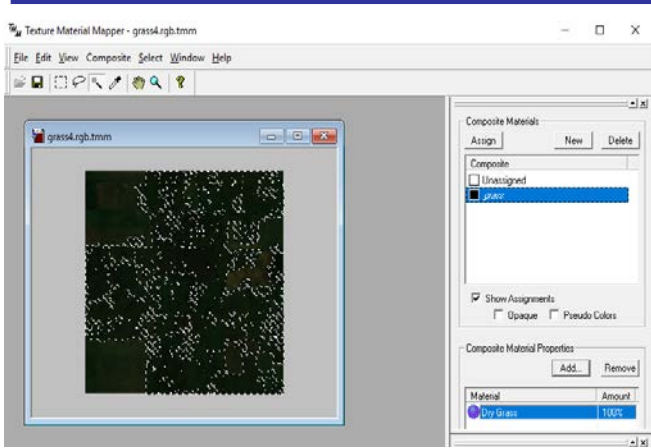
Vega comes with various modules to add realism to the simulation application. While most databases used for visual simulation look realistic, they lack this physical binding with the materials in the real world. The Sensor Products Utility Library comes with a database of many materials properties that are used by many of the sensor products. The default materials database that comes with the sensor products contains heat transfer, reflectance, specular, and radar data for over 150 materials. The assignment of the properties in the materials database to the geometry in a visual database is done through a process called material classification [17]. Figure 8(a, b, c) shows the process performed on each texture in the visual database using a TMM tool which is a shortcut the Texture Material Mapper



(a)



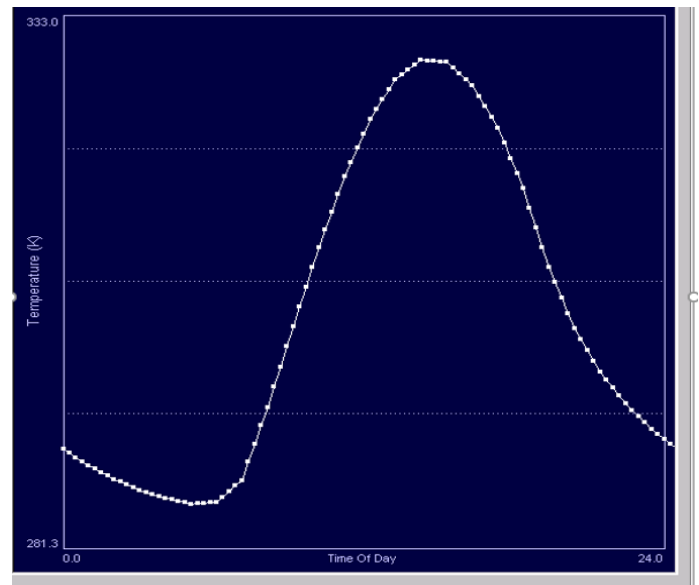
(b)



(c)

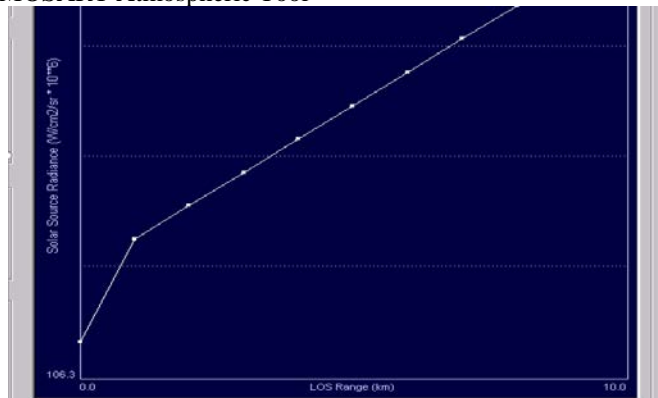
Fig. 8 (a). Target object material classification, (b) missile object trust material classification, (c) terrain object material classification

The Sensor Products Utility Library also uses pre-computed atmospheric quantities and material temperature and reflection data to speed up its run time calculations. Figure 9 (a, b, c) shows the generated database using MAT which is shortcut for MOSART Atmospheric Tool

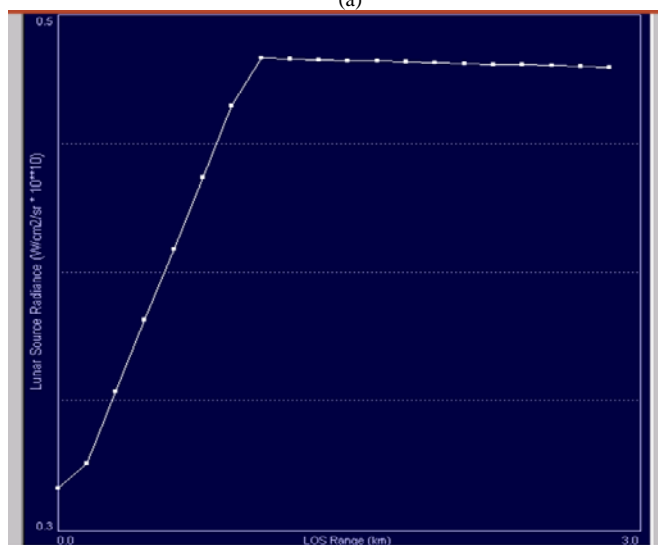


(c)

Fig. 9 (a) Solar atmospheric calculations, (b) Lunar atmospheric calculations, (c) heat transfer information calculated for each material



(a)



(b)

3) *Scene Driving*: Vega has lynx which is a graphical user interface and it is used to assign the initial simulation parameters to virtual scene. Weather, 3D objects and IR calculations are added to the virtual scene using lynx [18]. Visual simulation program is a program built within C++ and Vega environments to assign the initial inputs to flight simulation phase. Those inputs are launch conditions, position and target motion, while the outputs are ATGM flight trajectory data (range, angle of attack, speed, acceleration, hitting angle, the deflections of the jetivator etc.). These outputs are assigned to the ATGM object continuously at Vega runtime loop via Vc++ to update ATGM object parameters at each frame. Visual simulation Application is loaded within the MFC framework using Vega API to provide ATGM visual simulation system with graphical user interface to control simulation parameters.

III. RESULTS AND DISSCUTION

In this section, the simulation results are verified in MATLAB. 6-DoF ATGM mathematical model is used to calculate position and rotation angles of the missile in virtual environment. ATGM parameters are used as initial conditions to the mathematical mode. Various atmospheric conditions are used for evaluating the performance of the tracking algorithms used. Various imperfections that may arise in IR sensor is used for evaluating the performance of the tracking algorithms used. Figure 10 (a, b) shows color and frame difference tracking algorithms at ideal conditions.



(a)

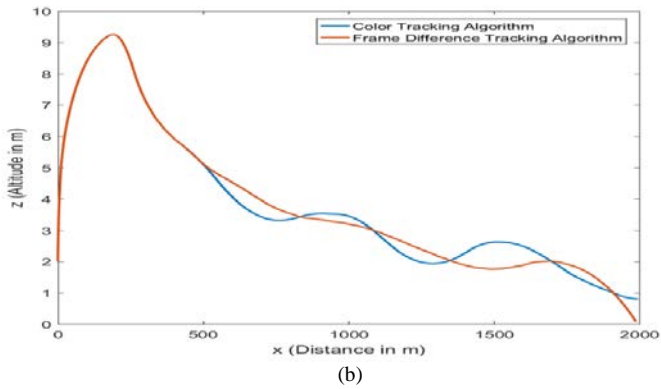


Fig. 10. (a) Visual simulation with ideal conditions, (b) Flight trajectory using two different tracking algorithms

Figure 11 (a, b) shows color and frame difference tracking algorithms at snow weather conditions.

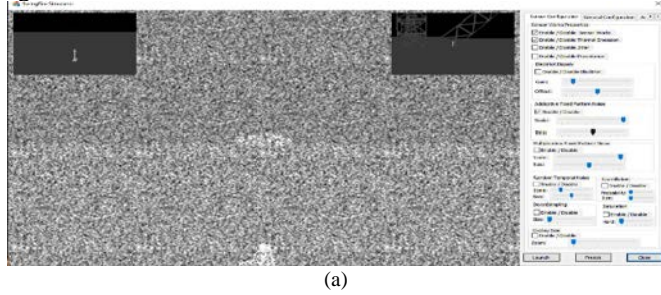


Fig. 11. (a) Visual simulation with adaptive noise conditions, (b) Flight trajectory using two different tracking algorithms.

Figure 12 (a, b) shows color and frame difference tracking algorithms at multiplicative noise conditions.

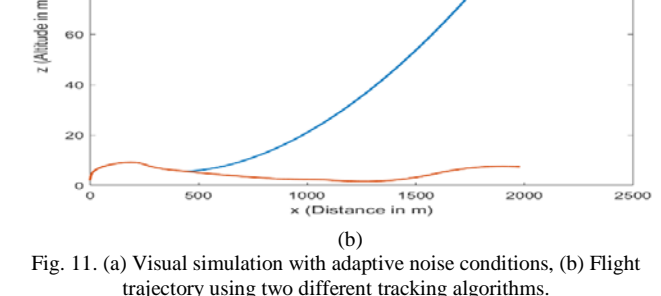


Fig. 12. (a) Visual simulation with multiplicative noise conditions, (b) Flight trajectory using two different tracking algorithms.

Figure 13 (a, b) shows color and frame difference tracking algorithms at snow weather conditions.



Fig. 13. (a) Visual simulation with snow weather conditions, (b) Flight trajectory using two different tracking algorithms.

Figure 14 shows target hit.



Fig. 14. Target hit

Figure 13 (a, b) shows color and frame difference tracking algorithms at snow weather conditions.

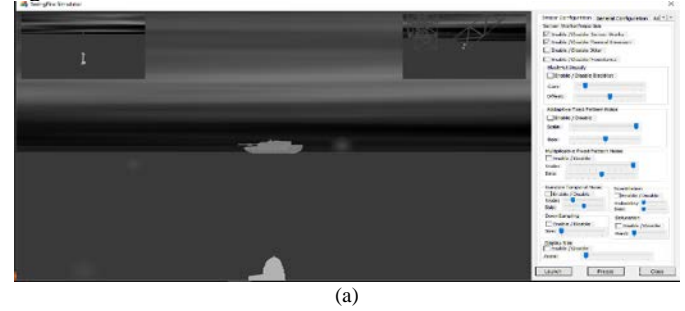


Fig. 13. (a) Visual simulation with snow weather conditions, (b) Flight trajectory using two different tracking algorithms.

Figure 14 shows target hit.

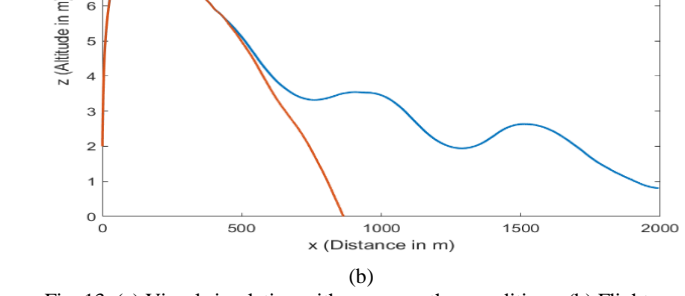


Fig. 14. Target hit

IV. CONCLUSIONS

This paper introduces a development of an IR visual simulation application by means of integrating Vega software with visual c++ to calculate different information [19] about the missile such as its speed, acceleration, missile relative position w.r.t target in the scene graph etc. to drive ATGM in visual simulation application. Multigen-Creator is used to create 3D models both missile and target objects and terrain database with all design aspects discussed before. MAT is used to calculate atmospheric quantities and material temperature and reflection while TMM tool is used to material classification as mentioned throughout the paper. Visual simulation application discussed in this paper is used for evaluating the performance of two different tracking algorithms (color and frame difference tracking algorithms). The tracking algorithms used in this paper are simple just to prove the concept of what this ATGM visual simulation application is created. The experimental results are verified using MATLAB and a curve is obtained from each test to analyze ATGM flight trajectory output. The simulation system discussed in this paper can be upgraded to test real parts of the missile itself in HIL visual simulation systems.

NOMENCLATURE

$X_1, Y_1, \text{ and } Z_1$	Vectors components about the board reference axes.
$X_g, Y_g, \text{ and } Z_g$	Vectors components about the ground reference axes.
T_{bg}	Transformation matrix from board to ground reference axes.
F_{AX}, F_{AY}, F_{AZ}	Drag, lateral, and lift forces along the velocity axes.
F_{TX}, F_{TY}, F_{TZ}	Thrust forces about the velocity reference axes.
$\omega_{x_1}, \omega_{y_1}, \text{ and } \omega_{z_1}$	Airframe-turn rates along board coordinate axes.
Ω	Angular velocity of VCS w.r.t GCS
$I_{xx}, I_{yy}, \text{ and } I_{zz}$	Moments of inertia components about the BCS.
$g_x, g_y, \text{ and } g_z$	Gravity acceleration about board coordinate axis

REFERENCES

- [1] Krupkin, Vladimir, and Shay Shlomi Alagem. "Countermeasure system." U.S. Patent 9,074,854, issued July 7, 2015.
- [2] Peng, Liang, and Xin Han Huang. "Design and Implementation of Real-time 3D Cruise Missile Simulation Training System Based on Vega Prime." In *Advanced Materials Research*, vol. 271, pp. 1819-1825. Trans Tech Publications, 2011.
- [3] GU Ying, GAO Jing-xiang, SUN Jiu-yun. Description of creator Pro/Vega Prime based three-dimensional topographic simulation techniques[J]. *Mine Surveying*, 2005(1): 27-29.
- [4] WANG Yun-xiang, ZHAO Jing-cheng, FU Zhan-cheng, et al. Research of scene simulation. Drive technique based on Vega Prime[J]. *Microcomputer Information*, 2008, 24(3): 214-216. Aly, M.S., "Dynamical Analysis of Anti-Tank Missile Systems", Proceedings of the 9th int. AMME conference, 2000.
- [5] LI Ya-chen, HU Jian, LI Yuan-zhong, et al. Research on various coordinate systems in spacecraft visual simulation based on Vega Prime[J]. *Journal of System Simulation*, 2007, 19(3):575-578.
- [6] Ouda AN, El-Sheikh GA, El-Halwagy YZ, Al-Ashry M (2010) Robust CLOS guidance and control part-2: system modelling and uncertainty evaluation. In: 14th International conference on aerospace sciences and aviation technology.
- [7] Ouda AN, El-Sheikh GA, Al-Ashry M, El-Halwagy YZ (2010) Robust CLOS guidance and control part-2: scalar H ∞ autopilot synthesis. In: 14th International conference on aerospace sciences.
- [8] Ouda AN, El-Sheikh GA, El-Halwagy YZ, Al-Ashry M (2010) Robust CLOS guidance and control part-3: HIL system simulation. In: 14th International conference on aerospace sciences and aviation technology.
- [9] Ouda AN, Cairo MTC (2012) Performance investigation of adaptive guidance algorithms and its effectiveness.
- [10] Zhang Bo, Ding Ying, Gong Guang-hong, 2007. "Research of Scene Simulation System Based on Vega Prime", Proceedings of 6th National Simulator Academic Conferences.
- [11] Liu Shu-xia, Fan Xiong-fei, Men Yan, 2009. "The Research on the Framework of Vega Prime Application Based on MFC", Proceedings of 2009 Conference of System Simulation Technology and Application.
- [12] Zhang, Ming-Hui, and Yao-Yu Zhang. "Software Implementation of Forced Parachuting Simulation System." (2015).
- [13] Ouda, A. N. "A robust adaptive control approach to missile autopilot design." *International Journal of Dynamics and Control* (2017): 1-33.
- [14] Guide, Vega Options. "Multigen Paradigm Inc." USA: Multigen-Paradigm Inc (2001).
- [15] Tian, Feng, et al. "Design and Implementation of Flight Visual Simulation System." arXiv preprint arXiv:1212.0365 (2012).
- [16] Sun, Dandan, Jiaobo Gao, Kefeng Sun, Yu Hu, Yu Li, Junhu Xie, and Lei Zhang. "Research on hyperspectral dynamic scene and image sequence simulation." In *Hyperspectral Remote Sensing Applications and Environmental Monitoring and Safety Testing Technology*, vol. 10156, p. 1015617. International Society for Optics and Photonics, 2016.
- [17] Wang, Ming-yin, et al. "Research on Key Techniques of Virtual Reality for Large Scenes Based on Creator/Vega Prime." *Journal of system simulation*21 (2009): 117-120.
- [18] Zhang, M.H. and Zhang, Y.Y., 2015. Software Implementation of Forced Parachuting Simulation System.
- [19] Yibo W, Yulong X, Xia C, Kai Z. Design of Airplane Flight Collision Detection Simulation System Based on Vega Prime. *Ordnance Industry Automation*. 2016;6:020.