Enhanced Aerial Radar Line of Sight Performance

Mr. Oren Koler, M.Sc.
Israel Aerospace Industries
Israel
okoler@iai.co.il

Ms. Tal Shintel, M.Sc.
Israel Aerospace Industries
Israel
tshintal@iai.co.il

ABSTRACT

Airborne Aerial Radars are often required to track a large number of ground vehicles moving within a specific area. When simulating such a radar, detection computation must consider the existence of line of sight between the radar and each of the simulated ground platforms, resulting in multiple long range LOS computations performed simultaneously, from a single aerial point. When using very high resolution terrain with ground vehicles scattered over large areas (hundreds of square miles) in dense vegetation, urban structures and mountainous terrain, the polygon count required for geometry intersection calculations used by each LOS query can be very high. In some cases, processing multiple LOS queries, results in poor simulation performance. This paper suggests a simple approach for reducing the number of required LOS calculations, where multiple long-range LOS queries originate from the same aerial point. By using a shadow map generation method and positioning the source of light at the LOS origin point, Ground points that are inside a shadow can then be filtered out from further LOS computations. The more ground points there are the better cost effective this method is.

ABOUT THE AUTHORS

Oren Koler currently works for the Israeli Air Industry (IAI) Lahav division (LD) simulation department as a system engineer. He has been part of the M&S community since 1997 starting out as a programmer for the Israeli Defense Force, in the weapons development simulation lab. He later became head of the CGF and simulation infrastructures team working on AI behaviors architecture, scenario and terrain enrichment, enhanced after action review, handling large scale distributed scenarios and research on advanced Human & Machine Interface (HMI) technology. Oren Koler holds an M.Sc. degree in Computer Sciences from Bar-Ilan University,Israel

Tal Shintel is currently leading several simulation activities in IAI, Lahav division. She has been working for over 20 years as a software architect in several simulation projects and has lead the development of HarTech's scenario generator, focusing mainly on supporting high-resolution, large scale scenarios and complex behavior modeling. Tal Shintel holds a M.Sc. degree in Computer Sciences from the Technion, Israel.

Enhanced Aerial Radar Line of Sight Performance

Mr. Oren Koler, M.Sc.
Israel Aerospace Industries
Israel
okoler@iai.co.il

Ms. Tal Shintel, M.Sc.
Israel Aerospace Industries
Israel
tshintal@iai.co.il

INTRODUCTION

In modern warfare, surveillance aircrafts are used to collect intelligence information from hostile enemy territory. These aircraft are equipped with advanced systems that provide different types of data, which can later be processed to form a coherent and comprehensive picture of the arena. An example for such a surveillance system is the EL/M-2055. This system consists of a high performance Synthetic Aperture Radar (SAR) and Ground Moving Target Indication (GMTI) and produces radar images, which approach photographic quality, and operates as a true all-weather, day and night sensor capable of penetrating clouds, rain, smoke, smog, fog and manmade camouflage. The EL/M-2055 is typically installed on Unmanned Air Vehicles (UAV). The SAR Imagery and GMTI plots generated on board are transmitted via the UAV's data link to the Ground Exploitation Segment (GES) for interpretation and extraction of valuable intelligence.
Another example would be the North Grumman E-8 Joint surveillance Target attack radar system (Joint STARS), which can be installed on manned aircrafts. The Joint STARTS tracks ground vehicles, collects imagery and relays tactical pictures to ground and air theater commands. The gathered information helps air and land commanders to control the battle space.

The benefits of aerial surveillance systems seem clear at first glance, as they provide comprehensive information of the arena. The quality of the information depends not only on the capabilities of each individual sensor but also on the deployment of sensors and aircrafts. The impact of surveillance aircrafts and systems on the battlefield in different scenarios is tested and analyzed in Battle Laboratories. The operation of Battle Laboratories is based on simulation of both the arena (using scenario generator) and of each individual sensor installed on the surveillance aircraft. The simulation of imaging sensors requires a very high resolution visual database and a graphical engine that can render high-quality pictures for the trainee to examine. This includes high resolution ground elevation, textures, dense vegetation modeling, urban structures and detailed entity models.
The simulation of GMTI, or any other tracking device, is based on modeling of the (simulated) environmental conditions and the sensor's detection capabilities. One of the criteria is the existence of line of sight (LOS) between the GMTI and each of the platforms in the scanned area.

When simulating surveillance systems. The detection computation of the GMTI must match the images generated for the same scanned area. In other words, the coherency of the information,(i.e. a moving ground vehicle that can be seen in an image is also detected by the GMTI , and vice versa) is more important than its accuracy, this coherency cannot be guaranteed unless the LOS computation uses of the same visual database and the same rendering engine as the one being used for simulation of the imaging devices.

In this paper we show how to reduce the computational load of LOS calculation on a graphical engine that is mainly designed for image generation, in the specific problem domain of long distance and large terrains.

DIFFICULTIES IN THE MODELING

Line of Sight

An LOS query determines whether two locations are mutually visible. To answer an LOS query a ray between the two locations is tested for intersection with the polygons that represent terrain, buildings or vegetation. An example of such computation is schematically presented in Figure 1. In the Figure, an LOS exists between the two green dots, but the line between the two red dots intersects four polygons, meaning that the polygons obscure the view from one point to the other.

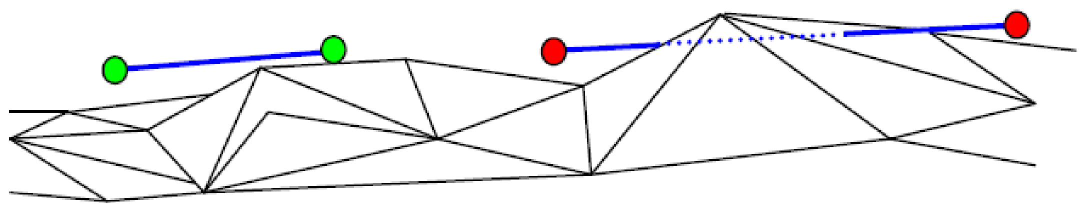

Figure 1. Schematic description of LOS computation

The complexity of LOS computation depends on the number of polygons that light be intersected. This number grows as a function of both range and resolution; more polygons will be tested over longer distances, and over detailed areas.

As Acquisition technologies keep improving, the simulation of sensors makes use of increasingly detailed terrain representations. An example of such detailed terrain is presented in Figure-2. The area presented in the picture is relatively small (200Km by 200Km), and it contains approximately 20,000 triangles.

Figure 2. High resolution terrain representation

LOS computation also depends on the distance between the two points of interest. When simulating a sensor that is installed on a (virtual) surveillance aircraft, the distance between the sensor and each potential target, which is usually a ground vehicles, increases as a function of the aircraft's altitude.
When simulating target detection of a single sensor there is only need to determine visibility from one to many O(n) entities and not many to many $O(n^2)$. However, the almost real time response time requirement demands conducting thousands of queries per second.

In a reference scenario, a surveillance aircraft flying at 30,000 feet is monitoring a ground area that contains 1,000 widely spread moving ground vehicles. For high quality images the terrain was in very detailed resolution with large dense vegetation areas. Initially, the computational time for 1,000 LOS queries was over one second. Reducing detection computation response time will allow an increased number of detections.

There are several methods to reduce the computation requirements for LOS calculation. For example, it is possible to use Bounding Volume Hierarchies (BVH) to reduce the number of intersection checks for each line of sight query. A different method utilizes the GPU by using 2D grid representation of the terrain. Both methods require pre-processing of the relevant area and therefore cannot be used for dynamic terrain. In addition, though these methods may be accurate in relation with the DTM, there is no guarantee that they will be compatible with the image generated by a rendering engine.

REDUCING LINE OF SIGHT WITH SHADOWS

Shadowing can be regarded as a different representation of LOS; if a point is located in a shadow then there is no line of sight between it and the source of light. As the application described above requires multiple LOS from a single source, the source (i.e. the sensor) can easily be regarded as the source of light for a shadow map. This map can then be used to identify the points for which no LOS exists without further computation. In addition, shadow mapping capability is supported by most rendering engines, and utilizes the same rendering algorithms. Therefore, the shadow map generated by a rendering engine is guaranteed to match the image generated by the same engine.

Shadow Calculation

Shadows are a result of the absence of light due to occlusion; when a light source's rays do not hit an object because it blocked by some other object the object is in shadow. In Figure-3 below, the blue line marks the illuminated parts of each object. Occluded segments that will be rendered as shadows are marked with brown lines.

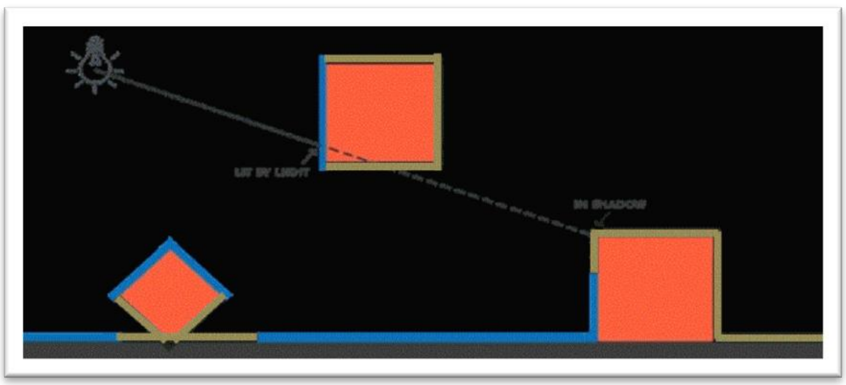

Figure 3. Shadows

Shadow mapping is a process in which shadows are added to 3D graphics, and is supported by many available rendering engines. In a typical computation of a shadow map each ray from the light source to the relevant area is separately tested; each point along the ray can either be illuminated or shadowed. Iterating through possibly thousands of rays from a light source is an extremely inefficient approach. A much more efficient algorithm is Curved Shadows on Curved Surfaces. This consists of two phases. First, the scene is rendered from the point of view of the light. Only the depth of each fragment is computed and saved in a z-buffer or depth image. Next, the scene is rendered as usual with each pixel compared to the z-buffer or depth image of the light source's view; Today, Much of the shadow calculation is done on the GPU.

Implementation

We use the graphical engine's shadow rendering capabilities to perform a conservative filtering step. We quickly cull away LOS queries with a definite terrain block. A point that is in a shadow is unseen. This shadow method is used in many video games and graphics today.

The visual database is first configured to load only terrain and structures in the highest Level of Detail (LOD), loading vegetation and entity models in high LOD dramatically increases shadow computation. This does not cause the shadow filtering step to return false results, but will only result in more regular ray casting (In small areas it is possible to load all data in max LOD).

The light source is attached to the simulated sensor, so that its location will be synchronized with the location of the surveillance aircraft. The observer's position, used as rendering point of view, is placed above the area of interest. An example of a terrain image and the shadow map generated over the same area is presented in Figure-4.

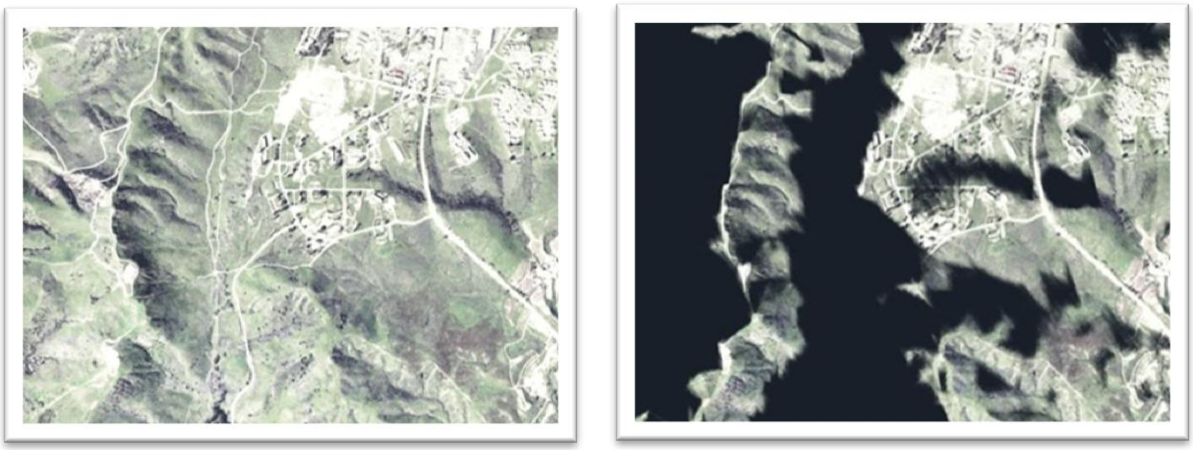

Figure 4. Left – terrain image, Right – shadow image

Once a shadow map is generated, a corresponding shadow matrix can be created from the rendered picture. This matrix contains only black or white values

Figure 5. Shadow matrix (black areas are shadowed, white areas are illuminated)

The matrix is saved with its geographical location and is regularly updated in a separate thread (the higher the refresh rate the more accurate the results are). In each detection computation cycle, the matrix is used to determine which of the moving entities are shadowed. These entities are definitely hidden from the sensor, while other points can be hidden or not, depending on the ray casting function.

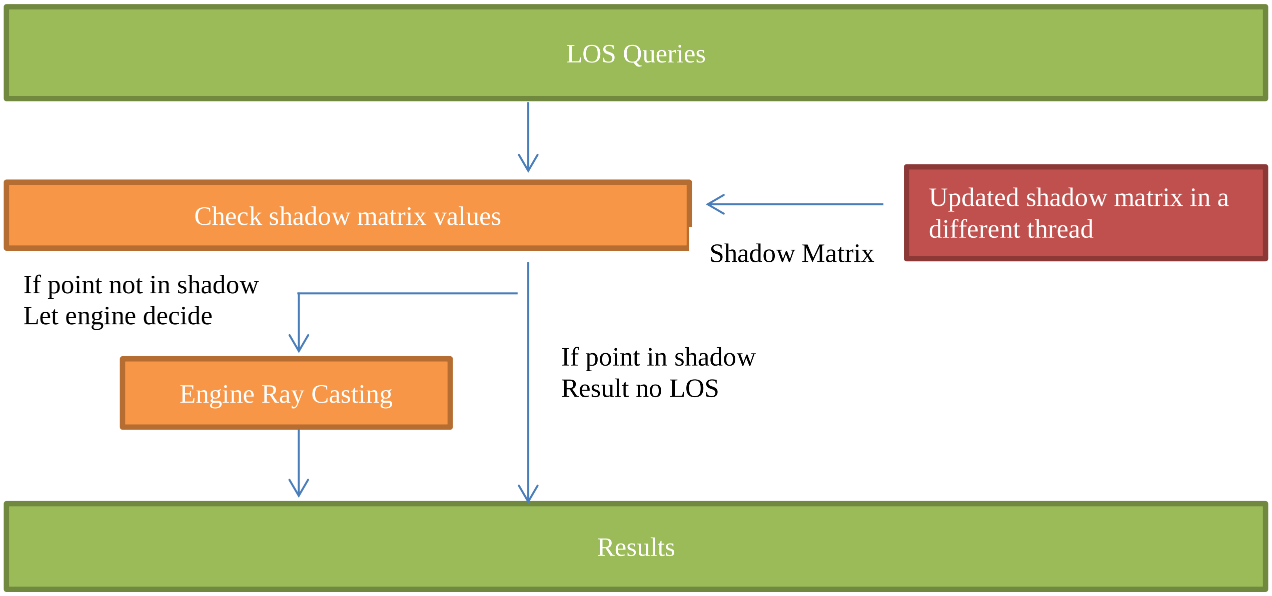

Figure 6. The culling is first performed in the shadow matrix, definite hidden targets are sent out and only targets that were not in shadow are sent for regular ray casting

In cases where a point is located at the edge of a shadow there is uncertainty if the 3D model is definitely hidden. In order to remove this uncertainty, the light source can be positioned a few meters above the sensor. This will result in smaller shadowed areas but insure that any point within a black value in the matrix is hidden. As in the case where vegetation and 3D models were neglected from the rendering, this too will result in more queries sent to the regular ray casting.

An example of such a solution can be depicted in Figure 7. In the left image the light source has been positioned at the location of the sensor. The red and orange dots are at the edge of a shadow, these are uncertainty points. In the right image, the light source has been positioned several meters above the actual sensor location. The orange dot is still within the shadow while the red dot is definitely outside the shadow thus cannot be determined by the shadow matrix.

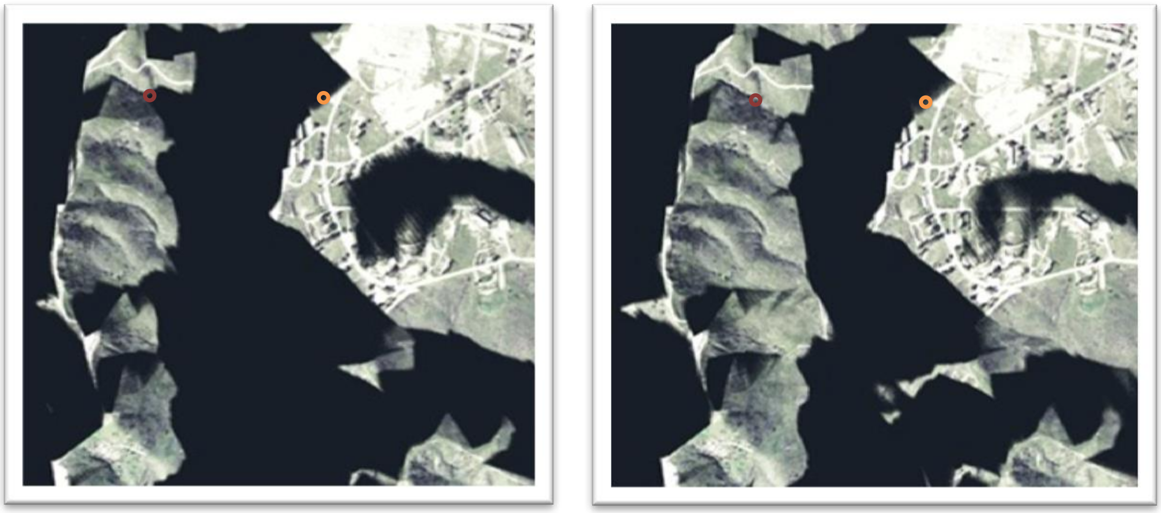

Figure 7. Reducing uncertainty by increasing light source altitude

In cases where there is on vegetation location vector data, it is possible to use the same mechanism to create a shadow matrix for filtering all definitely seen points. By decreasing the position of the light source a few meters below the sensor, it can be concluded that any point that is still within the white values of the shadow matrix is not blocked by terrain. If the point is also not within a vegetation area, it can be determined as a definite seen point.
An example of such a solution can be depicted in **Figure 8**. In the left image the light source has been positioned at the location of the sensor. In the right image, the light source has been positioned several meters below the actual sensor location (causing even more shadowed terrain). The dots that are still outside the enhanced shadowed areas are defiantly not hidden by the terrain**.** Those points need to be checked if there in the vegetation areas and if not, they can be considered as seen from the observation point.

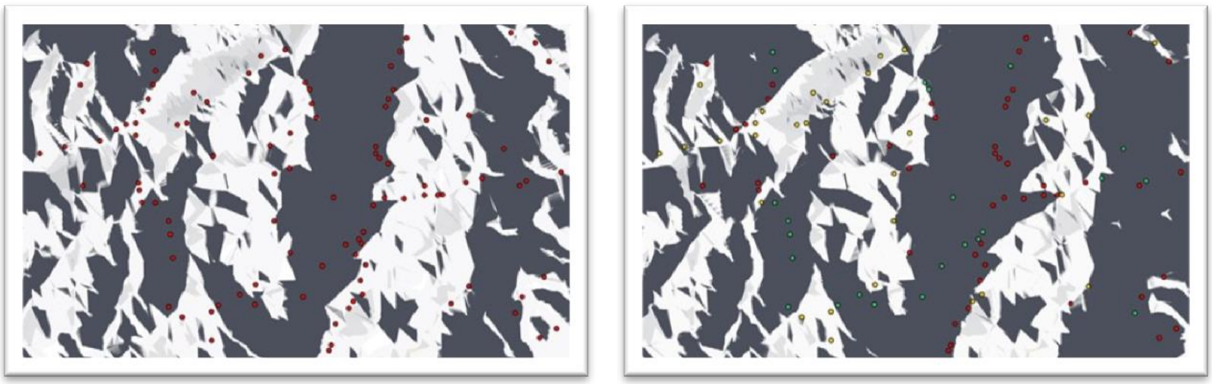

Figure 8. Reducing uncertainty by decreasing light source altitude

Experiment Results

Reference test scenarios for the system were located in a mountain area. 1,000 ground platforms were randomly placed within a 10Km by 100Km area of interest. Tests showed that 20-30 percent of all ground platforms were out of sight from the sensor due to large obstacles interference (see Figure-9 for example).

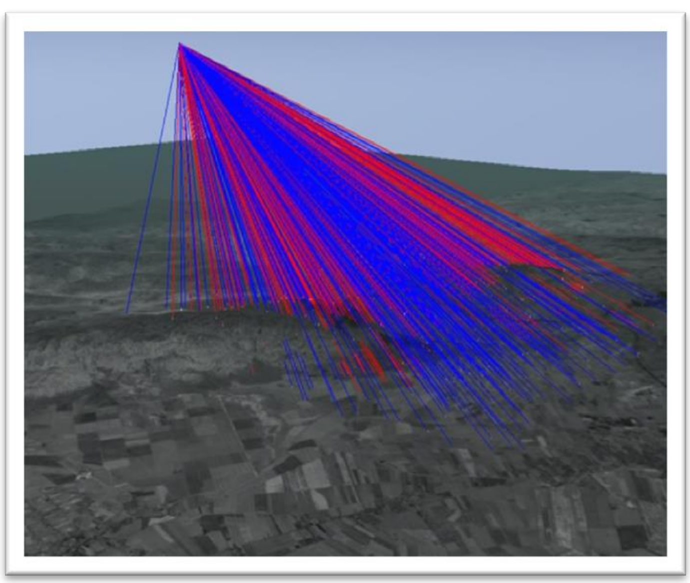

Figure 9. A red line indicates LOS from the sensor to a point blocked by terrain. A blue line indicates clear LOS

In a different reference test scenario where there was also vegetation vector data, 100 ground platforms were strategically placed within a 5KM by 5KM area of interest. Tests showed that 23 percent of all ground platforms were out of sight from the sensor due to large obstacles interference and 27 percent where not hidden by the terrain and where not in vegetation areas meaning definitely seen by aircraft visual dependent sensors. This shadow and light matrix filtered out a total of 50 percent of the points and thus cut the number of ray casting by half.
In our project the terrain was based on a Blueberry 3D format meaning very high dynamic resolution with relatively slow performance in LOS queries. While our shadow matrix and light matrix got updated every second in the background, the time for checking the matrix for every point was almost immediate O(1). Without the shadow filter our LOS server could determine only an average of 280 lines per second, with the shadow filter the amount jumped to an average of 600.

CONCLUSIONS

The use of shadows in a LOS system is commonly used in the gaming industry (especially in 2D gaming). The process of using shadows allows us to use the benefits of a fast computing GPU with no terrain preprocessing. It will work in the same manner in dynamic or static terrain, in high or low level of detail, in any graphical engine with shadow rendering capabilities and in any data base terrain format with regular hardware. In some cases it is possible to increase the resolution of the shadows and thus use the shadow map as the main LOS server and not just as a filter for blocking mountains but also for trees and smaller objects like human models and even for effects like smoke. However, it will only work with points clamped to the terrain and from one to many and thus excludes it from being a generic query server. Increasing or decreasing the light source altitude can increase filtering results, but needs to be configured and tested previously until the results are accurate. In large areas, long range, one too many, air to ground queries we found this method helpful in filtering above from 20 to 50 percent of LOS queries.

REFERENCES

Brian Salomon,Naga Govindaraju, Avneesh Sud, Russell Gayle, Ming Lin, Dinesh Manocha, (November 2005) Accelerating line of sight computation using graphics processing units. Department of computer science University of North Carolina at Chapel Hill.

Maheswari1, Ramaswamy reddy, (2013), Vehicle Object Detection in Aerial Surveillance, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 10, October 2013.

Alaa Abdullah Noori, (2012), Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian NetworksIEEE Transactions on image processing (volume 21, issue: 4).

C. Lauterbach, M. C. Lin, D. Manocha, S. Borkman, E. LaFave, G. Peele, M. Bauer (2008), Accelerating Line-of-Sight Computations in Large OneSAF Terrains with Dynamic Events , Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2008

John R. Isidoro, (2006), Shadow Mapping: GPU-based Tips and Techniques, Game Developer Confrence, March 20-24 San Jose,California.

Eden Paul,(2004), The encyclopedia of modern military aircraft. London, UK: Amber books.

A. F. M. Saifuddin Saif, Anton Satria Prabuwono, Zainal Rasyid Mahayuddin, (2014), Moving Object Detection Using Dynamic Motion Modeling from UAV Aerial Images, ScientificWorldJournal. 2014; 2014: 89061.

Donnelly, W, Lauritzen, A.,(2006), Variance Shadow Maps. Symposium on Interactive 3D Graphics, Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games. 2006, pp. 161–165.

Wolfgang F. Engel, Ed. Charles River Media,(2006) ,Cascaded Shadow Maps. ShaderX5, Advanced Rendering Techniques, Boston, Massachusetts. 2006. pp. 197–206.

Stamminger, Marc, Drettakis George,(2002), Perspective Shadow Maps, International Conference on Computer Graphics and Interactive Techniques, Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. 2002, pp 557–562.

Wimmer, M., Scherzer, D., Purgathofer, W., (2005), Light Space Perspective Shadow Maps. Eurographics Symposium on Rendering. 2004. Revised June 10, 2005. Technische Universität Wien.