Accelerating Virtual Walkthrough with Visual Culling Techniques

Mohd Shahrizal Sunar
Dept. of Computer Graphics and Multimedia
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia
81310 Skudai, Johor
Email: shah@fsksm.utm.my

Tengku Mohd Tengku Sembok
and Abdullah Mohd Zin
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor
Email: {tmts,amz}@ftsm.ukm.my

Abstract—Virtual walkthrough application allows users to navigate and immerse in the generated 3D environment with computer graphics assist. The 3D environment requires a large amount of geometry to make it look realistic. When the number of geometry increase, the performance of the application will become slower. Consequently, it creates a conflict between the needs of realistic and real time. In this paper, we discuss the implementation of visual culling techniques such as view frustum culling, back face culling and occlusion culling in the virtual walkthrough application. We render only what we can see during the application runtime and cull away unnecessary geometry. This will accelerate the performance of the system. Without the culling techniques implemented in virtual reality application such as virtual walkthrough, the system has to allocate a large space of memory to store the geometry data. We have tested these techniques to the Ancient Malacca data. With the visual culling techniques implemented, the virtual walkthrough system can work in real time mode without sacrificing realism factor.

I. INTRODUCTION

Virtual reality system have been used in many area and many industries such as aviation simulation, manufacturing, military, scientific research, construction and tourism. Generally, virtual reality system is needed for training, learning, planning, exploration and entertainment. Besides that, virtual reality system is important for modern problem solving and strategic planning tools. The virtual walkthrough application allows users to immerse the 3D environment. For example, urban simulation [1], driving simulation [2] and virtual heritage simulation[3] [4] [5]. All these mentioned application require that rendering to be done in real time.

The key factors of virtual walkthrough application are realism and the interaction between system and the user. Realism as define by [6] means how far the image can visualise in the users mental can be same as real experience in the real world. To generate the realistic image in virtual walkthrough, the geometric complexity level of 3D model have to be increase [7] [8]. This will also increase the memory usage simultaneously.

In the next section, we will discuss about the virtual walkthrough application. In section 3, we highlight the visibility concepts in computer graphics. It is including view frustum culling, back face culling and occlusion culling. We discuss our implementation and testing in the section 4 and 5 respectively.

II. VIRTUAL WALKTHROUGH APPLICATION

Simulation is largely used especially in the industry that involve with high operational cost and high risk. Therefore, in virtual simulation, the main consideration is the precise of the simulated environment with the real world. This meant, the movement and environment in virtual simulation must be same as the movement and environment in the real world. Most of the virtual walkthrough that involve with visual simulation in current market need a highly expensive special machine or better known as simulator.

Simulation in 3D environment used the object to be represented exactly same as real. But to do this, we require highly computational cost. This mean, more space for storage, higher performance processor and larger memory space. All of these are need to support the highly detail 3D graphics model in the simulation application. The detailed 3D graphics model normally consists of thousand polygons for each model.

There is a trade-off between realism and real-time [9] during the development of system that involve 3D graphics. If we increase the realism, it may slower the processing. But if we need more speed to run the system, then we need to sacrifice the realism. Virtual walkthrough application needs more real-time than realism. This is because a lot of interaction by user to navigate the application. If we choose to pick more realism factor, than the walkthrough application is meaningless. Nevertheless, realism still can be achieved without sacrificing to many realism needs [10].

The 3D environment with full of objects such as buildings, peoples and trees are very hard to visualise interactively at the real-time frame rates [11]. The virtual walkthrough application can be divided into two category; indoor [12] [13] and outdoor [14]. For indoor walkthrough application, the 3D
objects model is normally modelled in detail. Normally it is developed to be used by architect or building interior designer to give them the imagination of the pre-built environment. In the outdoor virtual walkthrough, the objects are consisting of an empty structure with only external appearance as cover. Most of the outdoor virtual walkthrough manipulate texture mapping [15] [16] for example urban walkthrough simulation. Sometimes, we also need to visualise both indoor and outdoor environment for a building. In the next section, we will discuss about visual culling techniques that can increase the real-time needs if the virtual walkthrough application.

III. VISIBILITY

In the projection from 3D environment to the screen, we can save a lot of things to improve the frame rate. Not all objects in the 3D environment will display in the screen at the same time. Therefore, the system has not to put into account the objects which are not to display in the memory space. We only render what we see.

The process of determining the visible objects are not an easy problem. Every little changes of the display may need to recalculate. For example, the movement of camera either horizontally or vertically may involve a lot of calculation for lighting and shadowing. This problem has been studied by many researchers as early as 1974 [17]. Figure 1 shows the different between visibility methods [18]. The visibility detection is not only useful for display but also for shadow generation and collision detection [19].

A. View Frustum Culling (VFC)

View frustum culling (VFC) method is the easiest visibility detection introduced by Clark [20]. It is an important algorithm to accelerate the rendering process which eliminates the unseen objects. It can be apply to all models in computer graphics. In this method implementation, only the objects which are inside the frustum will be displayed. To generate a frustum, a pyramid with surfaces is used. There are two surfaces that represent the nearest and furthers viewpoint. The nearest and furthers representing the boundary of visible objects. Figure 2 shows the representation of view frustum with six planes.

The view frustum is based on combination of six plane namely near, far, left, right, top and bottom. To start calculating the view frustum, eight points that combined all plane is determined. This is useful to check whether the object(s) is inside or outside the frustum view.

All six planes is defined by the plane equation as below:

\[ Ax + By + Cz + D = 0 \]

Every objects in the virtual environment is bounded by a bounding box such as Axis-Aligned Bounding Box (AABB). During the application runtime, if all the boundary of bounding box is outside the view frustum, then the object is cull away from the visibility calculation. Assarsson and Moller [21] had modified the algorithm with hierarchical concept that calculate more precise view frustum culling.

B. Back Face Culling (BFC)

The surface of object that is not facing the camera is not appeared in the display. Therefore, it is waste of memory resources for a real-time system to allocate a memory space the things which are not include in display. This method handles the polygons that are back facing the viewers not to be displayed [22]. Usually it is automatically performed by the rendering API such as Direct3D or OpenGL. If the angle between the polygon and viewer is more than 90 degrees, the polygon can be discarded. The back face culling can be expected to cull roughly half of the polygons inside the viewing frustum.

C. Occlusion Culling (OC)

The purpose of this visibility method is to determine Potentially Visible Set (PVS) to eliminate the occluded geometry. A survey of occlusion culling method have been done by Cohen-Or [18]. In general, these algorithm is classify into three types:

a. Conservative or approximate
b. Object space or image space
c. Point based or region based

There are some effective algorithms developed for specific environment. For example, cell and portal algorithm [23], [24] for indoor environment. This algorithm is useful for such application like interior architecture walkthrough. There are also algorithms for urban data set [25], [26], [27], [28], [15] which is consider to be used for larger environment using convex occluder.

The conservative algorithms calculate PVS including all visible primitives including the occluded primitives [25], [29], [26], [30]. While the algorithms for the approximate category are calculating only the estimation of visible objects [31], [32], [30].

Object space algorithms produce more efficient and precise selection using larger occluder. Boundary and subdivision method is used by [25] and [26]. Each object is divided into some fraction and only display the part that include in visible area. This method is difficult and involves a lot of calculation to implement [33], [27], [28]. The result can be used for large environment but it is difficult to generate occluder fusion that consist of many small occluder. For this kind of environment, the image space algorithm (hierarchical z-buffer [29], [34] and hierarchical occlusion maps (HOM) [30] are more effective. For the object space algorithms, subdivision and bounding method are used by [25] and [26].

From region based algorithms calculate PVS in the beginning to decrease computational during application runtime [33] [28]. These algorithms are suitable for broader environment and it also can balance between PVS quality and memory usage. These kinds of algorithm are difficult to find the best selection. The point based algorithms are more precise but expensive computational cost. [35] developed an efficient algorithm to calculate visibility with shadows for real time 3D environment.

IV. IMPLEMENTATION

This section describe the implementation of the visibility method in the virtual walkthrough application.

A. Virtual Environment Development

The basic thing that we need to do is to construct a virtual environment. Virtual environment consist of a 3D space with geometric objects. In this project, we develop a virtual environment using C++, OpenGL, GLVU and GLUI. The 3D geometric objects data is taken from Ancient Malacca project developed by Multimedia Development Corporation. The 3D objects are modelled by 3D Studio Max.

B. Visibility Techniques Implementation

The visibility techniques have been implemented to the developed virtual environment. The main purpose of implementation is to prove that with visibility technique, the performance of virtual walkthrough application can be increased.

Figure 3 shows the virtual walkthrough application without any implementation of visibility techniques. Each object is bounded by an Axis-Aligned Bounding Box (AABB). The total number of object in this virtual walkthrough is 137 objects. While figure 4 shows the implementation of view frustum culling, back face culling and occlusion culling techniques to our virtual environment. The occlusion culling techniques used are on geometric based. We can see the objects display is same as in figure 3 but the frame rates is increasing. The number of objects in memory is only 20 percent of the total object in the virtual environment. The number of polygon that holds is only 12,428 polygons compare to 71,442 polygons without visibility techniques.

V. TESTING

A set of camera movement path is fixed in our virtual walkthrough application. This is important to ensure the camera to follows the same path for every testing with different culling method. We used Intel Pentium 4 HT
3.2 GHz with 512MB RAM and nVidia GeForce FX5950 graphics display card to test the visibility method in our virtual walkthrough application. The comparison of tested visibility culling method in our virtual walkthrough application is showed in the graph in figure 5.

Graph in figure 5 shows the performance of walkthrough application in frame per second. The higher frame rates are the better performance. The application running without frustum culling shows the lowest frame rates. The highest frame rates can be achieved by implementing view frustum culling. With the implementation of occlusion culling techniques with bounding box, the number of objects can be decreased and we can get higher frame rates. But the result displayed is not giving all details. It is because the objects that are occluded by bounding box are cull away but in reality the objects still can be seen. The geometric based occlusion culling technique need more computational cost but the result is more precise. Therefore the frame rates are lower than bounding box based occlusion culling.

VI. CONCLUSION

The acceleration techniques are the most important thing in order to develop a good virtual walkthrough application. It is not only increasing the speed of the application but also increasing the mental of the user. Just imagine if the navigation system is full of lag because of the high computational burden.

Visual culling techniques are one of the acceleration techniques that can be apply to the virtual walkthrough application. It is significant to the trade-off for the realistic and real-time needs. From this paper, we can see the occlusion culling technique is increasing the frame rates without sacrificing the display quality of the virtual environment. As conclusion, we hope that this research is useful for the visual system simulation and games developer. Consequently, it can also benefit the edutainment community.

ACKNOWLEDGEMENTS

We would like to thanks Creative Application Development Centre (CADC), Multimedia Development Corporation, Cyberjaya, Malaysia for allowing us to use the 3D model from Ancient Malacca project as a testing data for this research.

REFERENCES


Fig. 5. The comparison of visibility method testing to the virtual walkthrough application.


