www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 10 October, 2013 Page No. 2951-2957

An analytical study on conic volume technique vs bounding box technique in picking object in Non Immersive Virtual world

K.Merriliance [1], ***Dr.M.Mohamed Sathik*** [2]

[1]Lecturer, Department of MCA
Sarah Tucker College (Autonomous),
Tirunelveli– 627007, Tamilnadu, India
merriliance@gmail.com

[2]Principal, Sadakathulla Appa College (Autonomous),
Tirunelveli– 627011, Tamilnadu,India
mmdsadiq@gmail.com

Abstract: ***In this paper we present an analytical study to select 3D objects based on their shape pointing device in a virtual environment (VE). We adapt a 2D picking metaphor to 3D selection in Virtual environment's by changing the projection and view matrices according to the position and orientation of a pointing device and rendering a conic selection volume to an off-screen pixel buffer. This method works for triangulated as well as volume rendered objects, no explicit geometric representation is required. In this paper I focused the advantages of conic volume technique over bounding box technique in picking object in Non immersive Virtual world. The usefulness and effectiveness of the proposed evaluation measures are shown by reporting the performance evaluation of two algorithms. We then compare the application of both techniques with related work to demonstrate that they are more suitable***

Keywords: object picking, Virtual environment, bounding box, conic volume, 3D selection, complex object, Non immersive VE

1. Introduction

Virtual Reality refers to virtual environments (VE) where multiple users connected by a network can share information with each other. Non Immersive virtual environment is viewed through a portal or window by utilizing a standard high resolution monitor. This includes mouse controlled navigation through a three dimensional Environment on a graphics monitor via stereo glasses stereo projection systems and others. Interaction with the virtual environment can occur by keyboards, mice and trackballs or may be enhanced by using 3D interaction devices environments, user interfaces and interaction, and important technologies used in VEs. Interactivity consists of functions for navigation, manipulation, picking of objects or information, and communication with other users within the virtual world.

In the virtual environment there are virtual objects whose attributes have to be represented and made interactive with the user by sophisticated software mechanisms. In addition, the ability to manipulate the data in an intuitive manner, move through the virtual world, and move around these complex objects allows users to focus their attention on the data and away from the interface. Object selection is among the basic building blocks of every interactive virtual reality system. It can be described as the process of identifying an object that is subject to further actions[1]. Selection by ray-casting is done by pointing at objects with a virtual ray. The closest object is selected. In VEs, the virtual ray is usually attached directly to a 6DOF sensor controlled by the user and subject to hand jitter. The selection by volume approach is used in situations where precise pointing is not required or not possible. Such techniques extend ray selection by projecting a volume, for instance a cone into the world.

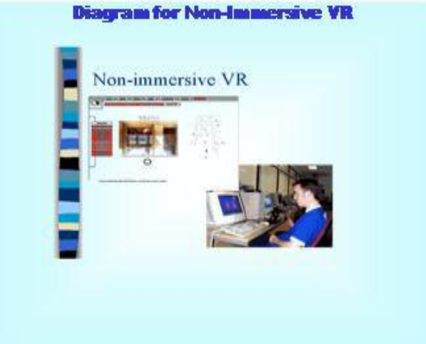


Figure 1:.Non Immersive Virtual Environment

Objects that lie inside the volume are candidates for selection. Common implementations of object selection use bounding boxes or bounding boxes to represent objects. The nearest bounding volume which is touched by the selection ray or volume is considered for selection. Non-immersive systems, as the name suggests, are the least immersive implementation of VR techniques. Using the desktop system, the virtual environment is viewed through a portal or window by utilizing a standard high resolution monitor. The boundaries are becoming blurred but all variations of VR will be important in the future. This includes mouse controlled navigation through a three dimensional Environment on a graphics monitor via stereo glasses stereo projection systems and others[2]. Non-immersive systems are a relatively new implementation of VR technology and borrow considerably from technologies developed in the flight simulation field. Non Immersive VE calls 'artificial reality,' requires no personal hardware. it can combine quite 'real' objects - tree, whatever – with 'unreal' ones, such as animatronics figures, holograms, disguised objects, optical illusions, etc environment - a room, a simulator, etc. - where normal sensory cues are not cut off, but are supplemented by additional sounds, images, or other sensations. The advantage of this type of VR is that it can be a communal experience, with many people participating at once. In a sophisticated enough environment, it might be difficult to determine who the "real" participants are. Non-immersive systems, as the name suggests, are the least immersive implementation of VR techniques. Interaction with the virtual environment can occur by conventional means such as keyboards, mice and trackballs or may be enhanced by using 3D interaction devices such as a Space Ball or Data Glove[3]. In this environment, they remain themselves, participating in their own quite real body, although obviously costuming and masquerade may be part of the experience. Figure1 illustrates the Non Immersive Virtual Environment.

As in immersive VR animation and simulation are interactively controlled in response to the user's direct manipulation. System attempts to give the users a feeling of being at least slightly immersed by a virtual environment, which is often achieved by different types of so called workbenches and reach-in displays. Non-immersive systems are a relatively new implementation of VR technology and borrow considerably from technologies developed in the flight simulation field. Using a wide field of view, these systems increase the feeling of immersion or presence experienced by the user. However, the quality of the projected image is an important consideration. It is important to calibrate the geometry of the projected image to the shape of the screen to prevent distortions and the resolution will determine the quality of textures, colors, the ability of define shapes and the ability of the user to read text on-screen which are more expensive.

Non-immersive virtual environments, such as Walls and Workbenches, are currently being used to incorporate virtual reality into scientific visualization. Unlike monoscopic displays used for current visualization environments, these VEs provide an environment where 3D visualization objects can be viewed in their native three dimensions that give the users an opportunity to gain a more complete understanding of these complex objects. In addition, the ability to manipulate the data in an intuitive manner, move through the virtual world, and move around these complex objects allows users to focus their attention on the data and away from the interface. The WALL is another commonly used device for creating non-immersive VEs. Stereoscopic scenes generated from the computer are projected onto a vertical screen. Many users can simultaneously see stereoscopic scenes on the screen by wearing special glasses. This device is commonly used for displaying data for a large group of people and interaction with 3Dobjects are rarely done since standing in front of the screen can obstruct the view of the audience. The simulation is an ideal case for positioning the viewing volume of the scene under the tabletop. This creates a VE that simulates a view similar to the environment where tuna swim. For visualization, time dependent unstructured cell data was loaded into AVS. Pressure distribution for the tuna's body was generated using standard visualization techniques. To visualize the changes in the pressure distribution, the geometric objects generated using time dependent data were animated.Non Immersive virtual environment is viewed through a portal or window by utilizing a standard high resolution monitor. This includes mouse controlled navigation through a three dimensional Environment on a graphics monitor via stereo glasses stereo projection systems and others. Interaction with the virtual environment can occur by keyboards, mice and trackballs or may be enhanced by using 3D interaction devices such as a SpaceBall or DataGlove. Here the person enters an environment - a room, a simulator, etc. - where normal sensory cues are not cut off, but are supplemented by additional sounds, images, or other sensations. System gives the users a feeling of being at least slightly immersed by a virtual environment, which is often achieved by different types of so called workbenches and reach-in displays. Non-immersive systems are a relatively new implementation of VR technology and borrow considerably from technologies developed in the flight simulation field. It-will comprise of a relatively high performance graphics computing system[4].

The concept of non immersive Virtual Reality is a three-dimensional, computer-generated environment that allows for single or multiple users to interact, navigate, respond, and experience a synthesized world modeled from the real world. It incorporates the interaction within a VE or virtual world by the help of various peripheral devices in order to achieve a multi-sensory and multi-modal experience. The user interaction in the system is divided into two different parts. The first one implements the implicit interaction and is used for inspection of 3D models and for navigation through them. The second one implements other user interaction possibilities like object selection and movement or menu handling.

Bounding volumes may be good in certain applications with widely spread objects but for non-convex objects that lie densely next to each other, a silhouette based selection may be more appropriate. However, efficient silhouette exact intersection methods, most likely hierarchical, usually require a huge amount of code for the seemingly rather simple task of object selection[5]. Furthermore, in most cases selection ambiguity is introduced by occlusion and cluttering of 3D objects. The limitation to bounding volumes will increase this ambiguity even more. Therefore, we believe that it is important to provide an efficient and above all simple way to perform object selection based on exact object silhouettes. We propose an implementation based on a selection technique that is well known on 2D desktop environments: select the object that corresponds to the pixel at the mouse cursor position. This technique can be implemented as a back buffer selection method. In a second rendering pass unique object identifiers are rendered to an off-screen buffer instead of colours and the pixel at the mouse position is read back from the graphics card. We adapt this method to 3D selection in VEs by changing the projection and view matrices of the off-screen buffer according to the position and orientation of a 6DOF pointing device. Hence, after the second render pass the off-screen buffer allows the computation of pixel based object statistics. State-of-the-art scoring schemes can then be applied to choose the active object[6]. Our approach is independent of any explicit geometric representation, meaning that it will work on triangular geometry as well as volume rendered objects.

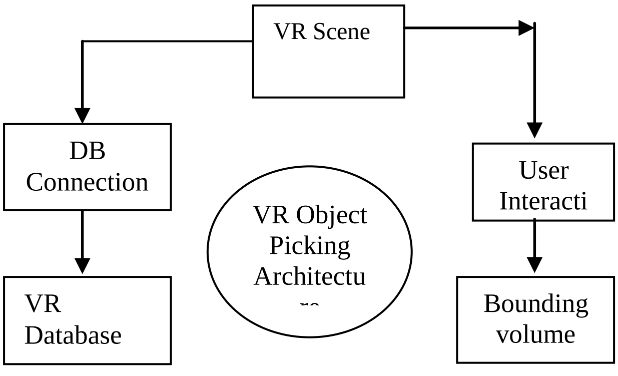
2. Object picking process

To perform picking in virtual reality is not easy, because we are restricted to our arms reach. It means that we are able to pick objects that are within notably limited area in the neighborhood to the user. Hence to realize these operations it is necessary to use interaction techniques that would allow us to accomplish given type of interaction better and faster. The interaction techniques that are used for object picking is very important for easier and faster user interaction with objects in virtual environment (VE). In this paper we present short review of picking an object from non immersive VE. We also show advantages and disadvantages of picking processing using conic volume technique and bounding box technique in VE and we try to adapt the methods to provide with better picking. The VE can contain all kinds of quite physical objects. Picking means objects can be selected from the scene. In a Virtual world there could be thousands of objects. Most of them will never need to be picked[7]. For example, in a virtual chat room based on a terrestrial metaphor, you may only want to select the door. The walls, windows and objects in the room never need to be picked. When picking objects, we need to map the 2D space onto the 3Dobjects in the scene. In 3D terms, this is referred to as picking. A virtual environment is composed of objects, which may be brought to life through their behaviour and interaction. Some objects will be static and have no behaviour. Some will have behaviour driven from the real world, for example by a user. Alternatively, object behaviour may be procedurally defined in a computer program. By replicating object behaviour we reduce dependency on the network and therefore make better use of available bandwidth thus increasing responsiveness. Early systems replicated object states but not their behaviour. Each state change to any object was sent across the network to every replication.The development of object picking algorithms has become a popular area of research in recent years with many applications developed. Since, many picking algorithms are being proposed, objectively measuring the efficiency of a picking algorithm has become necessary for comparing the performance of object picking algorithms. We may pick object within a specific bound which can be updated dynamically depending on changes in the view point of a user with in the 3D world using mouse. Clicking a mouse will create an appropriate picking bound at a 3D coordinate associated with the current mouse position. Object within a bound is selected. When no bounding box intersects with the picking ray, no object is selected.

Now that we have the ability to put objects in virtual world and move around them, it would also be nice to be able to choose which object we are focused on. One method of doing this would be to click on the object on the screen and have the camera refocus itself around that object. This method of choosing an object from the screen with the mouse is called picking. The first thing we have to write code for setting up the framework for picking and we need to do is have some input from the mouse to play with and see if an object in our scene was clicked on. The first part of picking is simply getting the mouse clicks and sending them on to our scene. The second thing is for getting the Scene to pick all of our Objects and to write the next part of the picking function, converting the 2D point into a 3D ray by projecting it using an inverse matrix we will create by taking a few settings. The method for finding out whether an object was hit by the ray is much simpler to implement because DirectX does a lot of this for us. All we have to do is convert the ray into the local coordinates of the model we are checking and have the built in Mesh. Intersect function tell us whether we have hit home or not. Now our engine will successfully test any mesh based objects that we load and report back with a true or false whether that object was clicked on or not. It will also set the clicked on object to be active in the scene so we can access it and play with other things once we know what was clicked. When picking an object, typically we are not interested in the entire scene graph tree to that object, just the actual object that was picked. Since the object is a visible item, it has no children and the end of the path is the picked object. The picked item can never be an item in the middle of the path. This module performs very important task of this algorithm[8]. Because this module only finds the top most plane among all the selected planes. This is done by maintaining the distance of the plane to eye. Object selection is among the basic building blocks of every interactive virtual reality system. It can be described as the process of identifying an object that is subject to further actions.

3. General Model for picking a 3D object

Now that we have the ability to put objects in virtual world and move around them, it would also be nice to be able to choose which object we are focused on. One method of doing this would be to click on the object on the screen and have the camera refocus itself around that object. This method of choosing an object from the screen with the mouse is called picking. The first thing we have to write code for setting up the framework for picking and we need to do is have some input from the mouse to play with and see if an object in our scene was clicked on. The first part of picking is simply getting the mouse clicks and sending them on to our scene. The second thing is for getting the Scene to pick all of our Objects and to write the next part of the picking function, converting the 2D point into a 3D ray by projecting it using an inverse matrix we will create by taking a few settings. The method for finding out whether an object was hit by the ray is much simpler to implement because DirectX does a lot of this for us. All we have to do is convert the ray into the local coordinates of the model we are checking and have the built in Mesh. Intersect function tell us whether we have hit home or not. Now our engine will successfully test any mesh based objects that we load and report back with a true or false whether that object was clicked on or not. It will also set the clicked on object to be active in the scene so we can access it and play with other things once we know what was clicked.The following figure shows the architecture of our VR object picking algorithm. VR database contains all the details of the VRML scene. VR scene can be created by referring the database. After that any user can walk through the scene and select any object using 3D input devices.



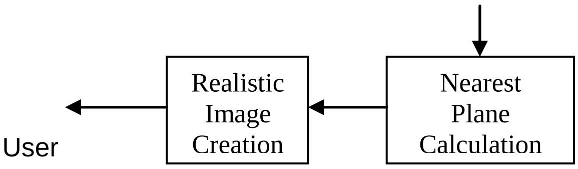


Figure 2: System Overview

When the user interacts with the scene, bounding box created and for only the objects within the boundary nearest plane calculation has been made. At last the realistic image of the top most planes is displayed to the head mounted display of a user. This algorithm reduces its complexity and run time by avoiding the nearest plane calculation for all the objects in the scene.

4 Conic volume technique in Object picking

This method works for triangulated as well as volume rendered objects, no explicit geometric representation is required. Their method uses bounding boxes to calculate conic volume intersections. Select the object that corresponds to the pixel at the mouse cursor position. All selectable objects are rendered twice. The second render pass assigns unique object identifiers instead of colors. To adapt this method to 3D selection we change the projection and view matrices of the second render pass to correspond to the position and orientation of a pointing device. Hence, the perspective view frustum of the pointing device represents a conic selection volume. The opening angle of the selection volume can be adjusted by changing the field of view of the perspective projection[9]. Visible pixels of the objects within the selection volume will be recorded by their identifiers in the off-screen buffer. State-of-the-art scoring schemes according to pixel based object statistics can then be applied to choose the active object. The user can select objects that are located within spotlight's cone. To select among more objects contained in the cone, it is possible to use a conic volume technique (Figure 3). It is modification of ash light technique where the selection of objects within the cone is done using a collision plane. The conic volume is determined by the position and orientation of the pointer’s tip, it’s orientation and a spread angle. First, we determine which objects are within the range of this volume. For every object in the scene, every frame, we perform an "inside"-test of the object and the conic volume. In our current proof-of-concept we limit our test to the bounding box of an object, which is typical for the objects in our user tests. We transform the object’s center point position and radius to the co-ordinate system of the pointer. In this co-ordinate system we can easily derive the projection distance on the ray (dperp) and the distance from this projected point to the pointer’s position (dpro j). Figure 2 shows a three-dimensional schematic overview of object P inside the conic volume. Point P lies on a two-dimensional cross section of the cone. This is a circular surface, with the ray defining both its center point and normal, while the radius is determined by the cone spread angle and distance dpro j .

dperp = dist= sqrt((cx-bpx[i])2 + (cy-bpy[i])2 + (cz-bpz[i])2) dpro j = z (2)

If the angle defined by these two distances, is smaller than the o to be inside of the conic volume. For these objects a scoring value will be calculated, while other objects will be ignored in the following stage. pening angle of our cone, we consider the object The Picking behavior enables selecting objects in the scene. A pick in 3D is usually carried out as Ray Pick.[11] The virtual camera position and the 2D mouse pointer on the image plane define the ray. By intersecting the objects in the scene with the ray, it is determined which one is picked

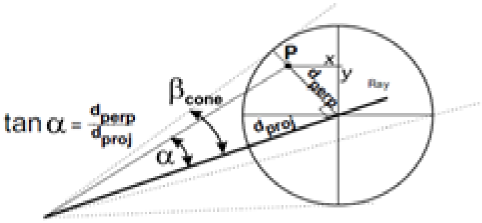


Figure 3:.selection volume test

Objects available in the second phase basic interaction task corresponds to an indivisible unit of information that is meaningful in the context of the application. In the case of interactive 3D applications they are: positioning, identification, entering text, and entering numeric quantities. Composite interaction tasks are built on top of the basic ones, actually being combinations of them and performing a unit of operation. Typical composite tasks for interactive 3D applications are model construction and direct manipulations on this model. This component provides two basic functionalities: the maintenance of selection state and the identification mechanism. The identification occurs when a user “clicks” a pointing device over the output of a camera. This process first detects whether that action occurred on top of some graphic primitive. If that happened, then the element that contains such primitive is located. Finally, the state of Bounding Box. The bounding box dragger allows direct interaction with its eight vertices, twelve edges, and six faces. In implementing our bounding box manipulator, which is inspired by [4], we assigned scaling to vertices, rotation to edges, and translation to faces. Two consecutive positions of a vertex are mapped into a scaling factor. Two consecutive points over an edge are mapped into a rotation about an axis, being parallel to the selected edge and passing through the center of the bounding box.

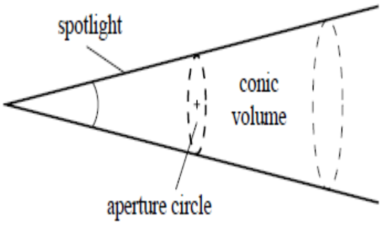


Figure 4:. Conic volume technique in Object picking

Each pixel position contains a value that is either zero (no object) or an integer number (ID) that corresponds to an object. Since we set up the projection and transformations according to the center pixel position c corresponds to the center of the selection cone. A popular scoring metric for instance used by the projected distance d between the cone center and an object at pixel position p which is

$$d = ||c- p||2.$$

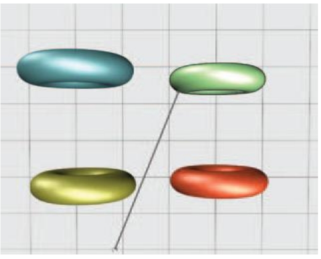


Figure 5:A pick ray selects the green torus.

Picking visual object using conic type of shape will be advantageous if the object has a cross section or size that is too small to be picked or intersected by a line. To illustrate the use of pick cone ray which has a picking shape that corresponds to a cone with infinite length. Smaller object to be easily picked. The main benefits of this method are:

(1) It is easy to implement. Additional implementation code is kept to a minimum.

(2) It reduces selection ambiguity.

3) Bounding volumes can significantly reduce occlusion problems.

4) This technique allows the easy integration of selection of non-transparent or segmented parts of volume rendered objects in the same way as triangulated geometry by using a different transfer function in the selection render pass.

5 Bounding box technique in Object picking

Three-dimensional bounding volume is a ubiquitous tool of accelerating various geometric calculations including intersection detection. In virtual environments, 3D picking is typically performed by bounding box checks or collision detection, taking the position of a virtual hand and the objects in account. As a basic idea for reducing the collision detection complexity, preliminary collision tests can be performed using simplified bounding volumes that contain the complex objects to be tested. In checking whether two objects intersect or not, bounding volumes make this intersection test more efficient, especially when the objects do not intersect most of the time. A bounding volume approximates an object by another simpler object that contains the original. A rectangular path always displays four small anchor points. For any object, you can select its bounding box—a rectangle that represents the object's horizontal and vertical dimensions. The bounding box is also called a container. The bounding box makes it possible to quickly move, duplicate, and scale the object without having to use any other tool. For paths, the bounding box makes it easy to work with an entire object without accidentally altering the anchor points that determine its shape. When you select one or more objects with the Selection tool, you see a bounding box that indicates the size of each object. If you don't see a bounding box when an object is selected, you may have selected the object using the direct method we will learn here is something called "Bounding Volumes". When we check for collisions or picking, we first check the bounding volumes of the objects, and if the bounding volume has been intersected, then we can move on to check the actual object the bounding volume is bound to. Scenes becoming more complex usually means more triangles to test against when checking for collision or picking[10]. So, we can learn here how to minimize the amount of triangles that need to be tested each frame. Figure 6 illustrates this picking phase using bounding box. To find it, we would iterate through all the vertices in the object, and distance between current cursor position and the array of boundary points of the object

To find the bounding box, we will need to create two points, the minimum and the maximum, which we will use to find the size of our bounding box, to make sure it tightly surrounds our object.

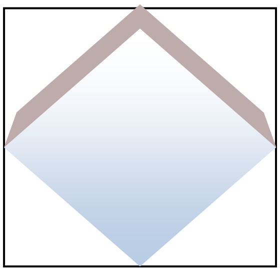


Figure 6: 3D object within a bounding box

To find the minimum and maximum points, we will iterate through each vertex in the object. After giving a rough overview of our proposed implementation technique we will describe the algorithm in more detail.

5.1 Algorithm for creating an Object Bounding Box(OBB)

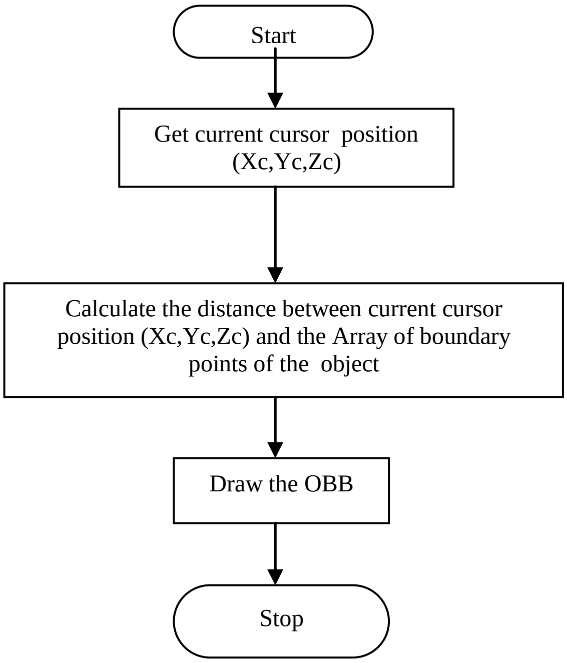
This module creates an object bounding box around the mouse clicked position with respect to the screen dimension.

Steps :

1. Get the coordinates of the current cursor position from the Scene.
2. Keep it as a center point (Xc,Yc,Zc) for the OBB which is to be drawn.
3. Calculate the distance between current cursor position and the Array of boundary points of the object
4. Draw the Bounding box at the center point with distance and depth.

First of all the each object's topological elements' geometry are determined and stored. If a pick is initiated by the user, the procedure starts with creating the object bounding box (OBB) around the selected position. Access the properties like position, orientation, scale, center of the bounding box etc by the mouse motion sensor and store in the buffer. Then connect to the database and find all the objects which come inside the bounding box using the accessed properties. This trick is used to avoid the accessing of all the objects details from the database to find which object is selected. Now it is enough to do the nearest plane calculation only for the objects within the bounding box[11]. For all the objects within the bounding box, converts its original data into the format suitable for shape and appearance. From the transformed data format build the 3 D part of an object[13]. For all the surfaces in the 3D part, determine whether the mouse lies on the plane and if lies on, set the found flag as true and store its distance to eye in the buffer else go to the next surface. Finally sort all the objects by its distance in descending order and display the top most object and its details after converting its local coordinates into screen coordinates.

5.2 Flowchart for creating an Object Bounding box (OBB)



Start
Get current cursor position (Xc,Yc,Zc)
Calculate the distance between current cursor position (Xc,Yc,Zc) and the Array of boundary points of the object
Draw the OBB
Stop

Figure 7: Flow Chart for Creating OBB

The distance between the center of the cursor and the array point can be calculated by using the equation (1)

distance= sqrt((cx-bpx[i])2 + (cy-bpy[i])2 + (cz-bpz[i])2)---(1)

where cx, cy, cz – Centre of the cursor position and bpx[ ], bpy[ ], bpz[ ] specifies the Array of boundary points of the object. This might not be the best method to use for large or irregularly shaped objects, which will have very large bounding boxes that might overlap with smaller objects nearby. With rectangular objects, it can be difficult to tell the difference between the object's bounding box and the path of the object itself. A bounding box always displays eight large hollow anchor points. Edges are handled similarly to faces, only on the vertex level the bounding box check is not needed, the vertex position is directly compared with the cursor position

If the center of the cube is inside the enlarged object-bounding box, all object faces are checked. Again, the procedure starts with enlarging the face bounding boxes by the pick distance and checks if the cursor center is inside the enlarged box. Actually, the enlargement is done component-wise in the bounding box check to speed up the process. If the center of the cube is inside the bounding box, the nearest point of the face geometry is calculated. The vertices are checked by simply comparing their position with the cursor position and the pick distance. Finally, if all these checks failed, an is-inside check is performed to see if the cursor is in the interior of an object without intersecting any of its topological elements. Access the properties like position, orientation, scale, center of the bounding box etc by the moue motion sensor and store in the buffer. Then connect to the database and find all the objects which come inside the bounding box using the accessed properties. This trick is used to avoid the accessing of all the objects details from the database to find which object is selected. Now it is enough to do the nearest plane calculation only for the objects within the bounding box. For all the objects within the bounding box, converts its original data into the format suitable for shape and appearance. From the transformed data format build the 3 D part of an object [12].

For all the surfaces in the 3D part, determine whether the mouse lies on the plane and if lies on, set the found flag as true and store its distance to eye in the buffer else go to the next surface. Finally sort all the objects by its distance in descending order and display the top most object and its details after converting its local coordinates into screen coordinates.The measured time was analyzed with a paired t-test.

Table 1: Time needed to complete the task and precision of the resulting models.

Matching type	*Mean time(min)*	*S.D*
Conic volume technique	8.50	4.32
bounding box technique	5.38	3.77

The time to complete the task in the Conic volume technique in Object picking found significantly lower compared to the time needed with a bounding box technique in Object picking. The mean time was 8.50 minutes (S.D. 4.32) in the conic volume technique compared to 5.38 minutes (S.D. 3.77) at the bounding box technique.

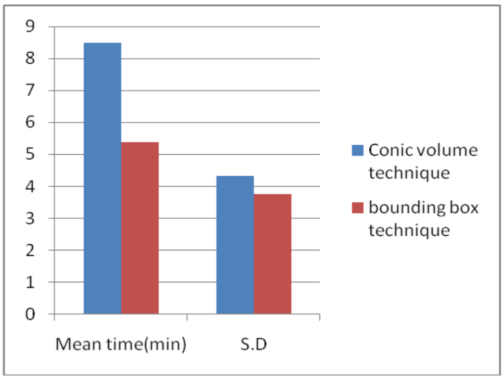


Figure 8: An analytical chart between conic volume technique Vs Bounding box Technique

6. Conclusion

The performed user study implies that picking 3D object can be performed faster in bounding box technique without loss of precision. Furthermore the study has shown that even users without a strong virtual reality background can achieve good reliable results in non immersive virtual environments. Using the bounding box technique will further increase the robustness and reduce the computational costs. The results provide an initial understanding of how these factors affect selection performance. Furthermore, the results showed that our new techniques adequately allowed users to select targets which were not visible from their initial viewpoint. Our analysis indicated that our introduced visual feedback played the most critical role in aiding the selection task. Study results have shown the performance of the Bounding box Technique was to be slightly better than the performance of the conic volume technique.

References

[1] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In Proc. SI3D 1997, page 35ff., 1997.

[2] Tobias Rick GPU Implementation of 3D Object Selection by Conic Volume Techniques in Virtual Environments

[3] Taylor II, R., et al., "The Nanomanipulator: A Virtual-RealityInterface for a Scanning Tunneling Microscope", ComputerGraphics Proceedings, ACM Siggraph, pp. 127-133, 1993.

[4] Sebastian Knodel." Navidget for Virtual Environments" Proceedings of the 2008 ACM symposium on Virtual reality software and technology.

[5] Williams, G., McDowell, I. E. and M. T. Bolas. Human scale interaction for virtual model displays: A clear case for real tools. In Proc. Of The Engineering Reality of Virtual Reality.

[6] Wu, 2002. shin - ting, marcel abrantes, daniel tost, and harlen costa batagelo ”picking for 3d objects”.

[7] Wu, X. 1992. A linear time simple bounding volume algorithm. In Graphics Gems III, David Kirk, Ed., chapter VI, pages 301–306.Academic Press, San Diego, CA.

[8] Foley, J.D., van Dam, A., Feiner, S.K., and Hughes, J.F. Computer Graphics: Principles and Practice.Addison-Wesley, 1990.

[9] Neider, J., Davis, T., and Woo, M. Open GL Programming Guide: The Official Guide to Learning OpenGL, release 1. Addison-Wesley, 1993.

[10] Mine.”Virtual Environment Interaction Technique”

[11] Herman: “Virtual reality a new technology: A new tool for personal selection”. Journal of neurosurgery, 2002.

[12] A. Steed. Towards a general model for selection in virtual environments. In 3DUI ’06: Proceedings of the 3D User Interfaces, pages103–110, 2006.

[13] A. Ulinski, C. Zanbaka, Z. Wartell, P. Goolkasian, and L. Hodges.Two handed selection techniques for volumetric data. In 3DUI ’07:Proceedings of the 3D User Interfaces, pages 107–114, March 2007.