www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242

Volume 3 Issue 12 December 2014, Page No. 9721-9725

Evaluating Effectiveness of 3D object picking algorithms in Non immersive virtual world

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

[1]Principal, Sadakathulla Appa College (Autonomous),
Tirunelveli– 627011, Tamilnadu,India
mmdsadiq@gmail.com
[2]Lecturer, Department of MCA,
Sarah Tucker College (Autonomous),
Tirunelveli– 627007, Tamilnadu, India
merriliance@gmail.com

Abstract:-. As '3D object picking in non immersive virtual world' is an essential interaction technique in virtual environment, it is performed using a 3D mouse or an interactive glove to explore and interact with any of the objects. In this paper we focused the advantages of picking using bounding box and pick cube algorithm over picking using bounding sphere and pick sphere algorithm .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 algorithms with related work to demonstrate that they are more suitable. These analytical studies provide distinct advantages in terms of ease of use and efficiency because they consider the tasks of object picking effective application-independent picking technique for various input devices. In this paper, cube and sphere are used to represent the intersection point of the ray coming from the user input and the object to be picked. By intersecting the objects in the scene with the ray with a pick cube at the intersection point, it is determined which one is picked. An object is selected when this pick cube intersects the object's bounding box.

Key words: -- Virtual environment (VE), pick cube, picks sphere, volume of intersection (VI), picked object bounding volume, object picking.

1. Introduction

Virtual information on real world is interactive and spatial. Visualization is the ability to present complex data as multidimensional images and combined with direct control over that data in virtual environment. In a VR environment the data is presented truly in 3D and users can interact with the data directly in the 3D space[1]. The advance of computer graphics knowledge and technology, itself tied to the enormous increase in processing power and decrease in cost, together with the development of relatively efficient and unobtrusive sensing devices, has led to the emergence of participatory immersive virtual environments, commonly referred to as "virtual reality. A key component in most virtual reality systems is the ability to perform a walkthrough of a virtual environment from different viewing positions and orientations. The walkthrough requires the synthesis of the virtual

environment and the simulation of a virtual camera moving in the environment with up to six degrees of freedom. 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 model from the real world[2]. 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. Object picking in a virtual environment is a dream that has not yet turned into reality. The focus of this work is on the development of new object picking algorithms for non immersive virtual environment. Picking, which involves the specification of one or more objects by the user, is one of the fundamental tasks in virtual environments. The goal of the work presented in this paper is to determine whether object picking algorithms are useful. In other words, we want to know when it makes sense to sacrifice the simplicity of immediate selection in order to improve speed and accuracy. There are several tasks that could benefit from algorithms that allow accurate picking in non immersive virtual environments without requiring users to be precise[3]. Object picking is one of the fundamental tasks in 3D user interfaces and the initial task for most common user interactions in a VE. In this comparative study, we review two major 3D interaction algorithms intended for 3D object picking to improve user performance in terms of task completion times and error counts. In this paper we describe about PICK CUBE and PICK SPHERE both are used to represent the intersection point of the ray coming from the user input and the object to be picked. So at the cursor position we can be able to see the pick cube or pick sphere while we are picking the object. By experimental approach, I find that PICK CUBE is better than PICK SPHERE for picking 3D object in non immersive virtual environment.

2. Object picking using pick cube Algorithm

The choice is highly dependent of the shape of the objects to be bounded. For elongated objects, possible solutions include bounding ellipsoids and cylinders. Thus this technique is too simple, more efficient and it helps in easy interaction with the virtual world, hence it makes the Virtual world user friendly. We have described how to find objects that are inside the pick cube. However, we do not know the exact contact point on the selected object which is often required in order to perform further actions or to provide visual feedback. We can exploit the structure of our picking implementation to avoid the computation of actual intersections[4]. In fact, it is sufficient to save the three dimensional vertex positions of the objects to the off-screen buffer additionally to the object ID. First of all the bounding boxes around each object and all its topological elements' geometry are determined and stored. If the user initiates a pick, the procedure starts with checking the pick cube against the object bounding boxes. This is done by calculating the volume of the intersection between the bounding boxes with the pick cube. So we have to find the intersected object with the pick cube. In principle, a Pick action has to be performed with every cursor movement against each and every object, to detect if the cursor has to be intersected to an object[5]. If more than one object intersected in the pick cube, minimum intersected volume object will be picked. Thus, the number of these calculations has to be further minimized beyond the level already reached with the bounding box checks to enable real-time snapping on moderately complex and precise models(see figure: 1). The overlap of two cubes is still a 'box'. If two corner points of an intersected object A are (x,y,z) and (x',y',z') (x'>x,y'>y,z'>z) and two corner points of a pick cube B are (a,b,c) and (a',b',c') (a'>a,b'>b,c'>c) then the volume of the overlap is defined in (1)

*VI=max(min(a',x')-max(a,x),0) * max(min(b',y')-max(b,y),0)* max(min(c',z')- max(c,z),0)---------(1)*

The overlap starts on the X axis at the maximum of the two coordinates x and a and ends at the minimum of a' and x'. If a' < x (i.e. a < a' < x < x') then there's no overlap and what happens is that max (a,x) = x > min(a',x') = a', so the difference becomes negative and the volume is zero. The overlapping area is described by a width of S1.xmax - S2.xmin, and a height of S1.ymax - S2.ymin. To determine the order of subtraction requires a couple of if tests.

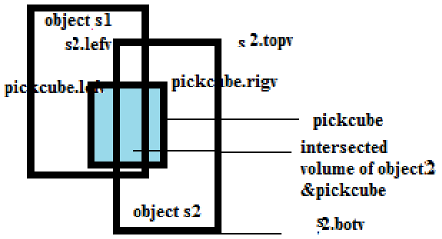

Figure: 1 *object picking using pick cube*

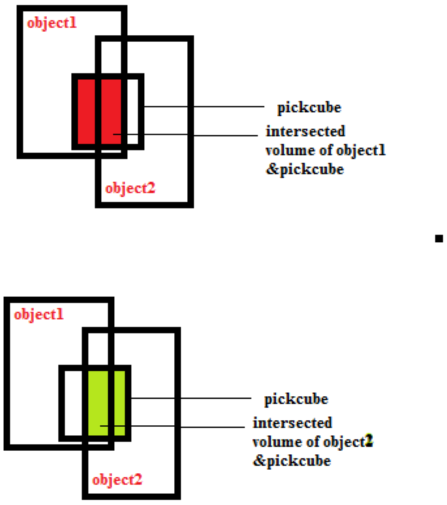

Figure2:*3D object picking based on volume pick*

cube

```
Algorithm
Input :     list of  all 3D objects in non immersive
virtual environment(vo),
            Pick cube;
Output:   picked object i;

Step 1: for each  i in vo do
            {
step2:  if pickcube intersect i
            {
step3:  Vol(i)= calcvol (i,pickcube);
            }
            }
Step4:  picked object= $max_{i \in vo} vol(i)$
step5:  maximum volume object to be assigned as
picked object
```

If volume of the intersection of the object1 and the pick cube is greater than the volume of the intersection of object2 and pick cube then object1 will be picked Else object2 will be picked. PICK cube works very well for picking small objects near larger objects, but it does not work well if the user points away from the center of an object.

Rotating a bounding box causes it to no longer be axis aligned. Because of this, if you rotate a model being bounded, you will need to recreate the bounding box. Doing so can be slow, since all the points in an object are iterated through to get the bounding box. If the model being bounded is not aligned to the axis, the bounding box will have some empty space. Empty space in the bounding box can result in false positives when checking for collision.

3. Object picking using pick sphere Algorithm

Bounding sphere is a hypothetical sphere that completely encompasses an object. It is defined by a 3D coordinate representing the center of the sphere, and a scalar radius. Checking for collision between two objects can be a very complicated task, involving tests between the many faces of each object. To perform this task quickly, simplified volumes are often used to represent each object, allowing fast tests for collision. Unlike a bounding box, a bounding sphere doesn't need to be recreated if the model rotates. If the model being bounded rotates, the bounding sphere will still be large enough to contain it[6]. Pick sphere is used to represent the intersection point of the ray coming from the user input and the object to be picked. So at the cursor position we can be able to see the pick sphere while we are picking the object. We have described how to find objects that are inside the pick sphere. We can exploit the structure of our picking implementation to avoid the computation of actual intersections. In fact, it is sufficient to save the name of the objects to the array additionally to the object ID. First of all the bounding sphere around each object and all its topological elements' geometry are determined and stored[7]. If the user initiates a pick, the procedure starts with checking the pick sphere against the object bounding spheres. This is done by calculating the volume of the intersection between the bounding spheres with the pick sphere. So we have to find the intersected object with the pick sphere[8]. In principle, a Pick action has to be performed with every cursor movement against each and every object, to detect if the cursor has to be intersected to an object. If more than one object intersected in the pick sphere, minimum intersected volume object will be picked[9]. Thus, the number of these calculations has to be further minimized beyond the level already reached with the bounding sphere checks to enable real-time snapping on moderately complex and precise models(see figure: 3).In the algorithm, the volume of intersection (VI) of objects and pick sphere as defined in (2)

$$A = (r*r) * \cos (((d*d)\} + (r*r) –(R*R)) / (2 * d * r)).$$

$$B= (R*R) *\text{arccosine} (((d*d) – (r*r) + (R*R)) / (2* d * R)).$$

$$C = 0.5 * (\text{sqrt} ((r + R - d) * (r - R + d) * (R - r + d) * (R + r + d))).$$

$$VI = A+B+C \text{ --------------------------------------- } (2)$$

Where r is the radius of the pick sphere, R denotes the radius of the intersected object and d denotes the distance between the centre point of the pick sphere and the intersected object.

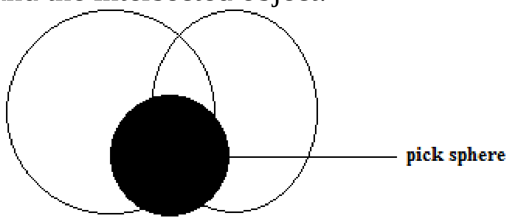

Figure:3 *bounding sphere with pick sphere*

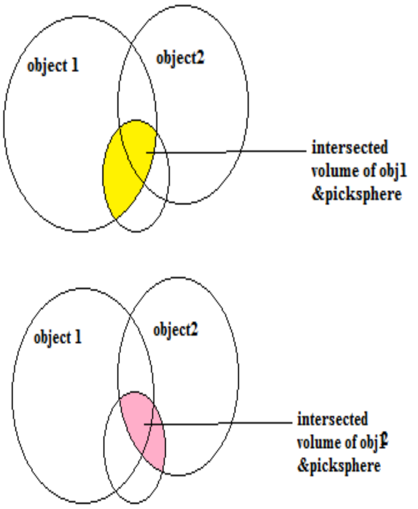
object 1
object2
intersected volume of obj1 &picksphere
object 1
object2
intersected volume of obj2 &picksphere

Figure:4 *3D object picking based on volume using pick sphere*

If volume of the intersection of the object1 and the pick sphere is greater than the volume of the intersection of object2 and pick sphere then object1 will be picked Else object2 will be picked. PICK SPHERE works very well for picking small objects near larger objects, but it does not work well if the user points away from the center of an object[10].

Algorithms	Test model 1	Test model 2	Test model 3	Test model 4
Pick sphere technique	0.12	0.16	0.1391305	0.165217
Pick cube technique	0.06	0.08	0.069565217	0.095652

Algorithm

The algorithm is used to pick a 3D object from non immersive virtual environment. The detailed steps are listed as follows:

```
Input : list of  all 3D objects in non immersive virtual
environment(vo),
        Pick sphere;

Output: picked object, i;

step1: for each  i in vo do
              {
Step2:  if  pick sphere intersect  i
            {

Step3: Vol(i)=calcvol (i, pick sphere);
            }
     }
```

Step4: picked object= $max_{i\,€\,vo}\ vol(i)$
Step5: maximum volume object to be assigned as picked object

We conducted a short experiment to demonstrate and evaluate how well the algorithms work for picking a desire object. We used the volume of intersection test to guide our choice of the test cases for the experiments.

4. Performance evaluation

In this study, we have focused on the most standard 3D interaction tasks allowing one to explore non immersive virtual environments or to pick objects. We have described a lot of picking algorithms for interaction with virtual environments under various levels of cursor positions[11]. In all possible comparisons, it provides a good exposure to many of the more common algorithms currently in use. We understand their strengths and weaknesses. The first performance measure was error rate from the centre of the target object at the end of a trial. The second measure of performance was Accuracy analysis to complete a selection**. T**he results are shown in Table 2. The last measure was the number of misses. We plan to continue this analytical study by performing a more formal experiment using the best of these algorithms on several specific tasks. It will also be important to determine the differences between these Virtual Environments which have sets of usability Characteristics. The same possible factors were used to create a predictive linear model of the error in pointing. The trials of each test models asked the subjects to perform a sequence of picking over a range of test cases. the results are shown in Figure 5.

TABLE 1: Error Rate Analysis

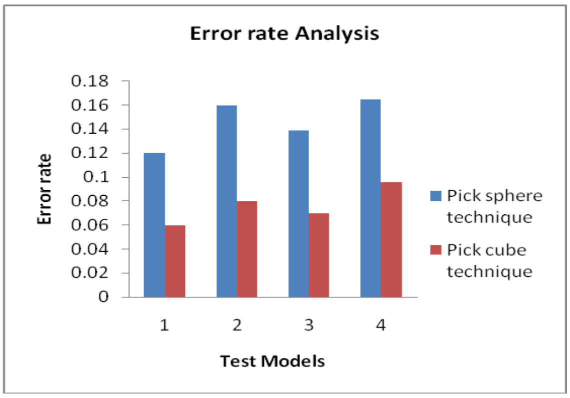

Figure 5: *The mean picking error rate per each picked object.*

TABLE 2: Accuracy Analysis

Algorithms	Test model 1	Test model 2	Test model 3	Test model 4
Pick sphere	0.88	0.84	0.8608695	0.834783
Pick cube	0.94	0.92	0.930434783	0.904348

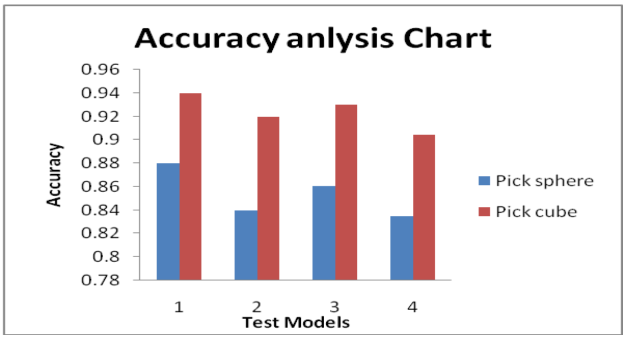

Figure 6: *The accuracy (mean percentage of correct selections)by the two picking algorithm*

Post comparisons showed that pick cube (error rate=0.076304 and average accuracy=0.923696) were significantly faster than the pick sphere algorithm (error rate=0.146087 and average accuracy=0.853913).

5. Conclusion

Various object picking algorithms have been developed for interactive virtual environments.In this paper, we present a set of design guidelines and strategies to aim the development of picking algorithms which can compensate for non immersive virtual environment. Based on these guidelines, we present new forms of the volume with positioning, selection and average cost of performing intersection test feedback, to support picking object within the virtual environments. We discussed an implementation of the proposed algorithms based on the understanding of how these factors affect performance. Using the bounding volume technique increases the robustness and reduces the computational costs. Therefore, it is important to examine the comparative study that are available and determine those that are best suited for the tasks that need to be accomplished. At last, in the two algorithms were compared, it proves that the bounding box with pick cube technique can work better in 3D object picking than bounding sphere with pick sphere technique and provides the user with an easy and precise way to pick the desired object, independent of its size, location or orientation. The performed user study implies that picking 3D object can be performed faster in bounding box with pick cube technique without loss of precision. We hope that this survey can aid both researchers and developers in having a clearer overview of the topic and in particular can be useful for practitioners and researchers that are new to the field of 3D object picking technique.

References

[1] Frederick P. Brooks, Jr ., What's Real About Virtual Reality?, IEEE Computer Graphics and Applications, 1999.

[2] Cecilia Sik Lanyi, Zoltan Geiszt, Peter Karolyi, Adam Tilingerand Viktor Magyar, Virtual Reality in Special Needs Early Education, The International Journal of Virtual Reality, 2006.

[3] Andre Stork, An Technique for Fast Picking and Snapping using a 3D Input Device and 3D Cursor, Fraunhofer Institut fur Graphische Datenver

[4] AMO-VILLANI N., WRIGHT K.: SMILE: "Effects of platform (immersive versus non-immersive) on usability andenjoyment of a virtual learning environment for deaf and hearing children". ACM Proceedings of SIGGRAPH 2007.

[5] Ivan Poupyrev Interaction Lab, Sony CSL: Beyond VR: "3D interfaces in Non immersive Environment". (2001).

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

[7] George C.Robertson: "Non Immersive Virtual reality".(2005)

[8] "Interaction for VR". Year of Publication: 2003. ISBN:1-58113-578-5.

[9] ISSAC"A META CAD system for virtual Environment"Computer-Aided Design, Volume 29, Issue 8, August 1997,Pages 547-553 Mark R Mine.

[10] "The simple virtual Environment library User's Guide"-www.cc.getech.edu.

[11] Elsivier."Special issue on Virtual Environment interaction". Journal of Visual Languages & Computing · Volume 10, Issue 1, February 1999.

[12] M.Mine."Virtual Environment Interaction Technique" (1995).