ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

TECHNOLOGY

Shape Matching Technique vs Bounding Volume Technique in 3D Object Picking

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

[*1]Lecturer, Department of MCA,Sarah Tucker College (Autonomous), Tirunelveli– 627007, Tamilnadu, India

[2] Principal, Sadakathulla Appa College (Autonomous), Tirunelveli– 627011, Tamilnadu, India

merriliance@gmail.com

Abstract

In this paper we present an analytical study to pick 3D objects based on their shape pointing device in a 3D environment. We adapt a 2D picking metaphor to 3D selection in 3D environments by changing the projection and view matrices according to the position and orientation of a pointing device and rendering a bounding volume to an off-screen pixel buffer. This makes it difficult for users to perform tasks that require them to select objects that have a small visible area, since small targets require high levels of precision. Both an analytical evaluation based on a 3D object picking and bounding volume technique demonstrate that progressive refinement selection can be better than shape matching technique. In this paper I focused the advantages of Shape matching technique over bounding volume technique in 3D object picking. The usefulness and effectiveness of the proposed evaluation measures are shown by reporting the performance evaluation of two techniques. We then compare the application of both techniques 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.

Keywords: object picking, 3D environment, bounding box, shape matching, 3D selection.

Introduction

3D objects have surfaces, edges and vertices. When picking objects, we need to map the 2D space onto the 3D objects in the scene. In 3D terms, this is referred to as picking. The transformation of the 2D mouse position to a location in the 3D world is an important process in picking. Because computer display is really a regenerated 2D view of the underlying 3D world.2D view is referred as the viewing plane. Picking is the act of identifying objects in the 3D scene, usually with a pointing device, such as the mouse. Picking can also be used to implement simple collision detection and response within a 3D scene. A pick in 3D is usually carried out as a ray pick. The ray is defined by the virtual camera position and the 2D mouse pointer on the object plane. By intersecting the objects in the scene with the ray, it is determined which one is picked? Not only objects but also their topological elements, i.e. faces, edges, vertices, can be picked, which is especially important in CAD. In virtual environments 3D picking and/or grabbing is typically performed by bounding box checks or collision detection taking the position of a virtual hand and the objects into account. Commonly known approaches to collision detection can be performed in real time only when applied to facetted models. CAD models are typically non-polyhedral, so that collision detection does not apply very well in this application context. Another modeling on is, that collision detection mainly focuses on sparse environments with many small moving objects. In contrast to that, in a part modeling application, the scene is mainly build up by one complex CAD model with a moving 3D cursor. With the 6 degrees-of-freedom input device, the 3D echo can be rotated, so that the beam points into the desired direction. The beam has a radius to enable picking of tiny objects. Combining 3D input devices with ray picking in these applications contradicts the idea of 3D input devices and direct 3D interaction 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 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. 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. There are several tasks that could benefit from techniques that allow accurate picking object in 3D environments without requiring users to be precise.

3D Object Picking

Picking an object is the combination of handling a mouse click or movement and Mapping a 2D screen coordinates into the 3D world .A 3D mouse or an interactive glove can be used to explore and interact with any of the objects that have been assigned dynamic properties. For instance if the VE contains a car, the car door can be given dynamic properties and constrained to allow them to rotate about their hinges through a specified angle. The user opens a door by moving the icon of the 3D mouse towards the position of the doors handle. When the user activates a button on the mouse, the selection of the door is confirmed. This is called picking. Each 3D scene has the database(s) to maintain its details. When the user interacts with the scene the bounding box sphere is created by calling the module bounding volume technique. Finally the realistic object is displayed with its attributes. Things which are inside the Virtual Environment are known as objects. The objects computer-generated stereo objects are projected onto the surface of the workbench .Interaction means objects in the scene can be manipulated. On top of this basic level we implemented operations on objects' topology and geometry, such as removing and adding vertices of objects, tweaking of vertices, and choosing and moving around objects. For example pick one object from the 3D scene. Common approaches that use a 3D cursor combine it with a usual ray pick. In the field of 3D it is important that the pick generates precise information using the accurate model and supports the identification of topological entities such as faces, edges and vertices. Using a mouse to select objects in 3D is a little tricky because the mouse gives only 2D pixel coordinates which must be somehow converted to 3D coordinates. In fact, the mouse location on screen represents an infinite number of points in world space which are projected on to a single point in screen space. In a 3D environment, there may be more than one object under the mouse pointer when it is clicked. Normally, the user's intention is to select the object which is visible at this point. The general approach will be to use the mouse coordinates to generate corresponding points on the near-plane and far-plane in world coordinates. These points will form a ray. The ray will be compared against every object. For intersection If more than one object is intersected, the object nearest the viewer is selected. 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. 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 .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. Use Object picking to identify the objects on the screen that appear near the cursor. To use Object picking, the software must be structured so that the picture can be regenerated on the screen whenever picking is required. As there was nothing special about the notation used for2D shape picking, the same approach can be used for 3D object picking.

Bounding Volume Technique in 3D Object Picking

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. 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 [6].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. Because bounding volumes are chosen to have much simpler topology and geometry than the original objects, checking the intersection between bounding volumes can be performed with a lower computational cost. Bounding Volume is a 3D object that encloses an object. Different types of bounding volumes may be considered, each of them having their own strong points and weaknesses. We will be making it so that you can "pick up" and move objects after you have placed them. We would like to have a way for the user to know which object it's currently manipulating. You can use the showBoundingBox method to create a box around objects. Our basic idea is to disable the bounding box on the old current object when the mouse is first clicked, then enable the bounding box as soon as we have the new object. The 3D input devices adopted by these systems allow for direct 3D interaction, thus to completely support 3D interaction. This approach, which utilizes other structures in the scene, typically uses a ray from the eye point through the current pixel to identify the first intersection point with the scene. This intersection is then used to compute the position of the 3D object. However, this approach suffers from severe problems in complex scenes. As an example for a heuristic approach we list the idea of using a library of predefined objects with predefined movement behaviors. These behaviors are then used to constrain objects to particular places in a scene. A ray along the current mouse position is then used to find the places in the scene where the constraints are fulfilled and the object is close to the cursor position. Therefore, we keep the pick ray connected to the object, but gradually straighten the ray every time the movement of the user's hand decreases the angle to the object, whereas the object's position is unchanged.

Hand movements not decreasing that angle drag the selected object as in single-user manipulation. This way the pick ray gets unbent in a continuous and transparent way, which intuitively resolves the issues of the feedback of multi-user interaction. The ability to navigate through a world seen only on your computer screen, or through a special headset or visor, opens the door for an incredible variety of experiences. It adds the ability to navigate through a virtual environment or the capability of picking up objects, or otherwise interacting with objects found in the virtual environment, and the basis for the enthusiasm for the technology becomes readily apparent. 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. There are two types of Bounding Volumes. They are (i) Bounding Boxes (ii) Bounding Spheres. The bounding boxes are usually axis-oriented, described by two opposite corner vertices, and the bounding spheres are described by the center and the radius. A Bounding Box for an object is just a rectangular box in three dimensional space, with sides parallel to the coordinate planes, that contains the object More complicated bounding volumes may be considered for efficient bounding when a small number of bounding primitives are required. Such volumes use more parameters in their description, allowing a wider range of shapes in optimizing their filling efficiency and trading away some of their computational simplicity. 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. Here, we do not look at objects on a polygonal basis anymore, but at the bounding spheres surrounding them. A Bounding Sphere is the smallest sphere possible including all vertices of the object (see fig. 1). Therefore, the necessary description of an obstructing object is reduced to the center coordinates of its Bounding sphere and its radius.

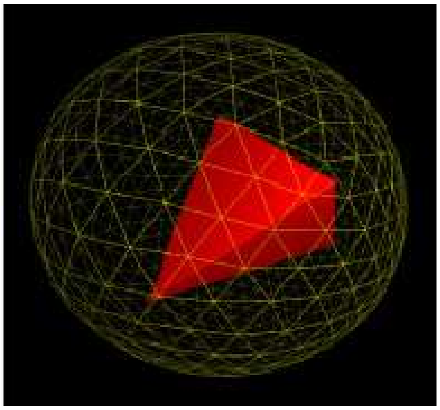

Fig.1: Determination of obstruction using Bounding spheres

Both conditions can be verified using the 3Dobject picking method, replacing the object itself with its Bounding volume. If there is an intersection of Bounding sphere and viewing volume, both conditions are fulfilled. The degree of obstruction v for the respective object (d) can be calculated from

$$v = 1 - \frac{d}{r}$$

If there is more than one acoustically relevant object located in the sound propagation path, again attenuation coefficients of all objects between listener and current sound source are accumulated and an obstruction value total is set. Main drawback is that the objects are registered in a fairly inaccurate way, because Bounding spheres are per se always bigger than the original

ISSN: 2277-9655
Impact Factor: 1.852

object's dimensions they represent. This might lead to the technique assuming obstruction although the sound source is still entirely visible. Still, in most cases the error introduced is negligible. Only if the object's extension into one dimension is much bigger than into the other two, i.e. the Bounding sphere's volume is only partly filled by the object, the effect is obvious.

Shape Matching Technique in 3D Object Picking

In 3D graphics the most common way to define 3D models is by a list of points in 3D space called vertices. The vertices are used to define triangles that can be displayed on a screen. To make it much more efficient it is common to use a list of indices which defines in what order to draw the vertices making it possible for triangles to share vertices reducing the number of vertices in each model. For more about vertices an d indices see or any basic book about 3D graphicsFigure 2 illustrates this selection phase. Sphere-casting avoids the precision issues of ray-casting, and also allows selection of occluded objects. Upon completion of the first phase, all objects that were inside or touching the sphere are evenly distributed among four quadrants on the screen, without regard for the spatial locations of the objects in the 3D environment. In this approach we can measure the similarity between shapes and exploit it for object recognition. The measurement of similarity is preceded by solving for correspondences between points on the two shapes and using the correspondences to estimate an aligning transform between 3D objects.

Corresponding points on two similar shapes will have similar shape contexts, enabling us to solve for correspondences as an optimal assignment problem. Given the point correspondences, the estimation may be the transformation that best aligns the two shapes; the dissimilarity between the two shapes is computed as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transform. The most critical and time consuming part in the bin picking process is object localization. With the advances in 3-D picking technologies efficient and robust techniques for geometric models are needed in much research. There are two steps for object localization. If the position and orientation of these objects are roughly known, the pose refinement the exact match of the object. This reduces its complexity by comparing only the reduced representation of an object model to a scene data set. The obvious advantage is an increasing performance. Many techniques, methods and object representations exist for surface registration problems. Especially in computer graphics object location and pose estimation is a common task. There exists a huge variety of applications in architectural, medical, industrial and scientific 3D-visualization, 3D-modeling, reproduction, reverse engineering and 3D-imageprocessing. The main disadvantage of the technique is the computational complexity. In most cases this technique cannot be used to process huge data sets in real time. Many improvements were made in the past to speedup the registration process. Our goal is to reduce this search time by decreasing the number of iterations and the number of corresponding points in each iteration step.

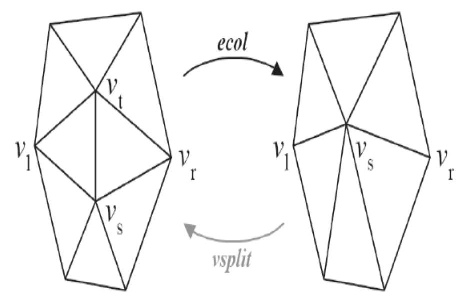

Fig2: Closest faces are checked for similarity

The vertex vt and vs are unified into the new vertex vs. This results in a reduction of the vertices and the number of faces in the mesh for every simplification step. Assuming we have a mesh M with n vertices, this mesh can be simplified by applying an edge collapse transformation until the base mesh M0 is reached. The vertex split is the inverted operation to the edge collapse transformation. Given the base model, we can add new vertices incrementally to reconstruct the original model.

IF abs(D1−D2) _ Thresholddist AND abs(V1−V2) _ Thresholdvol THEN SIMILAR

ELSE

NOT SIMILAR

END IF

To solve the problem of the outliers at the first box and also to be able to check if there exists a partial match in the hierarchies, the modified technique is to be checked if there are matches within the hierarchy. Just matching parts of the hierarchies for fitting resulted in quite poor result as it very often found good matches even if the box decompositions were not the same. This was caused by that no angles were used and that small parts easily can be very similar. Approaches like always beginning at the top neither gave good enough results. Suppose that each object class is represented by its

 ISSN: 2277-9655
Impact Factor: 1.852

features. As above, let us assume that the jth feature's value for the ith class is denoted by iij. For an unknown object the features are denoted by Uj.

$$S_i = \sum_{j=1}^{N} w_j s_j$$

The similarity of the object with the ith class is given by where Wj is the weight for the jth feature. Objects can have infinite complex shapes and thus creating an technique that takes all its parts into account to find a good grip will probably be too time-consuming for an autonomous robot which is expected to react within seconds from a command is given. Mean Square Error (MSE) is commonly used as a quality predictor to compute time needed to pick 3D objects. MSE is defined as:

$$MSE = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \frac{(P_0(i,j) - P_c(i,j))^2}{MN}$$

where p0 is the original 3D scene consist of set of 3D objects, Pc is the picked object, M and N are the width and height of the object respectively.Bounding volume technique was much more accurate than Shape matching technique and Bounding volume technique was faster than Shape matching technique with small targets and less 3D environments. The time to complete the task in the bounding volume technique found significantly lower compared to the time needed with a shape matching technique in Object picking process. Although Bounding volume technique converges within even fewer iterations, the overall time is slightly more because one iteration takes more computation due to the additional parameter estimation. 3D picking problem can be reduced to a problem of determining the object that intersects at a given point the eye-ray fired from the center of projection through the pixel's center into the unprojected scene.

Thus this technique is too simple, more efficient and it helps in easy interaction with the virtual world, hence it makes the 3D world user friendly. That is, the robot can add its target objects without constraint of shapes or types except one constraint that the object has some texture on its surface for object modeling in our framework. Based on the results of a series of user's studies, we presented a list of guidelines for techniques to pick objects in 3D scenes. Depending on the application, it can be advantageous to provide visual feedback to the user on the state of the semantic pointing functionality. When we analysis these two techniques in the view of selection and cost of performing mean square error test , picking using bounding volume is always greater than the average cost along the other 3D picking techniques. In the future, we would like to extend and to apply our technique to the generic collision detection field.

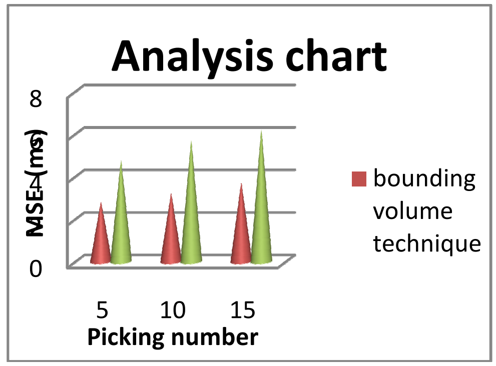

Fig. 3: An analytical chart between bounding volume technique Vs shape matching Technique

From this result, we recommend Bounding volume technique for more robustness, and provides faster performance.Fortunately, evaluation errors in the results are small and do not change the overall property of the 3D scene. It is therefore important to note that bounding volume technique should only be used to estimate perceptual quality of objects for which geometry is an important component of perceived shape.

Conclusions

In this paper, we present a set of design guidelines and strategies to aid the development of picking techniques which can compensate for 3D environment density and target visibility. We discussed an implementation of the proposed techniques. Based on these guidelines, we present new forms of the bounding volume and shape matching techniques, which are augmented with positioning, selection and average cost of performing intersection test feedback, to support selection within dense and occluded 3D target environments. The results provide an initial understanding of how these factors affect picking 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. Using the bounding volume technique will further increase the robustness and reduce the computational costs. The comparison technique performs maximization of the posterior parameters of all known objects. 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 techniques were compared, it proves that the bounding volume technique can work better in 3D object picking than shape matching technique and provides the user with an easy and precise way to pick the desired object, independent of its size, location or orientation. 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 The performed user study implies that picking 3D object can be performed faster in bounding volume technique without loss of precision.

References

[1] John Vince, Virtual Reality Systems, Published by Pearson Education, 2002.

[2] Kiyoshi Kiyokawa, Haruo Takemura, Yoshiaki Katayama, Hidehiko Iwasa and Naokazu Yokoya,VLEGO: A Simple Two-handed Modeling Environment Based on Toy Blocks, Nara Institute of Science and Technology (NAIST), JAPAN.

[3] Rogier van de Pol, William Ribarsky, Larry Hodges and Frits Post, Interaction in Semi-Immersive Large Display Environments.

[4] Fadel, G.M., Crane, D., Dooley, L., Geist, R. " Support Structure Visualization in A Virtual Environment" Presented at the sixth International Rapid Prototyping Conference, Dayton, OH, 1995.

[5] Weimar, The Projector-based, Desktop Reach-In Virtual Environment, Eurographics Symposium on Virtual Environments/ Immersive Projection Technology , July 2007

[6] M. Moore, J. Wilhelms, Collision Detection and Response for Computer Animation, 1988.

[7] K. Shimada, J. Cagan and S. Yin, Geometric Representations for Intersection Detection in Intelligent Packaging, 1998.

[8] Rolf Klein, Thomas Kamphans, The Minkowski Sum of two arbitrary polygons , Institut für Informatik , 2001.

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

[10] 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.

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

[12] 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.

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

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

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

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

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

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

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