Volume 1, No. 3, Sept-Oct 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Analysis of Object Picking Algorithms in Non Immersive Virtual Environment

K.Merriliance*
Lecturer in MCA Department,
Sarah Tucker College,
Tirunelveli, India.
merriliance@gmail.com

Dr M.Mohamed Sathik
Associate Professor in Computer Science Dept,
Sadakathullah Appa College,
Tirunelveli, India.
mmdsadiq@gmail.com

***Abstract*:** A defining feature of virtual reality is the ability to manipulate virtual objects interactively, rather than simply viewing a passive environment. 'Picking' is an important interaction technique in graphics applications. Object picking is typically performed by bounding box checks or collision detection, taking the position of a virtual hand and the objects in account. We present an adaptation of object picking algorithm in virtual environments using shapes. Such picking algorithms have been widely used for selecting objects under a 2D mouse cursor. In this paper, we propose and analyze a method of various object picking algorithms in non immersive virtual environment and the discussion of comparing these algorithms and provide the usability and a picking mechanism is essential for any direct-manipulation application. A user study of these was performed which revealed their characteristics and deficiencies, and led to the development of a new class of techniques. 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 algorithm for various input devices. The key idea is to represent the signature of an object as a shape distribution sampled from a shape function measuring global geometric properties of an object.

***Keywords*:** Virtual Environment, object picking, Pick ray, positioning, pointing.

I. INTRODUCTION

The challenge of VR is to make those objects appear convincingly real in many aspects like appearance, behavior, and quality of interaction between the objects and the user environment. VR is the technology that provides almost real and believable experiences in a virtual way. So it uses the current multimedia technologies such as image, video, sound and text, as well as newer and upcoming media such as e-touch, e-taste, and e-smell. 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. The manipulations of virtual objects and meetings with synthesized collaborators have been proposed as special human computer interfaces for the scientific visualization process. In the field of VR-like systems, conventional ray-picking is still widely used. The user can then interact with virtual objects by using traditional tools, such as a pen, or specifically designed physical icons, which are tracked on the surface using a variety of sensing techniques. In the earliest stage Physical objects are tracked on the table by using markers attached to them. An overhead camera and computer-vision techniques enable us to estimate the objects' 2D positions on the table. The physical objects can then be used for interactions on the table, e.g. by manipulating them, we can select and move virtual objects. The object position follows the mouse cursor position closely, while the object always stays in contact with other surfaces in the scene. In contrast to existing techniques, the movement surface and the relative object position is determined using the whole area of over lap of the moving object with the static scene. The resulting object movement is visually smooth and predictable, while avoiding undesirable collisions. The technique also utilizes the fact that people easily recognize the depth-order of shapes based on occlusions. Finally, the evaluation of the new technique with a user study shows that it compares very favorably to conventional techniques in any interactive graphics application. When the object moves in the virtual world its system of coordinates(x, y, z) moves with it. Therefore the position and orientation of object vertices in the object. System of coordinate remains invariant, regardless of the object position in the scene.

A. Related work

A number of authors have investigated the performance of object manipulation with 3D input devices, such as a spaceball or a six degree-of-freedom tracker. Such devices enable direct interaction with a 3D scene and are typically used in VR systems. One of the first authors to pick an object is typically performed by bounding box checks, taking the position of the pick cube and the objects in account so that the algorithm is made simpler and the computation time of the algorithm is also reduced. Pick cube can be activated to perform more actions during interactions than other shapes. And also states to Use Object picking to identify the objects on the screen that appear near the cursor. Subsequently many other researchers studied the creation and manipulation of 3D environments in VR.

Users can interact with the scene using tracked pinch-gloves, and the predefined object behaviors facilitate 3D object manipulation. The problem of determining the similarity of two shapes has been well-studied in several fields. The vast majority of work in shape matching has focused on characterizing similarity between objects in 2D images. Another popular approach to shape analysis and matching is based on comparing high-level representations of shape. For instance, model-based approaches first decompose a 3D object into a set of features, and then compute a dissimilarity measure between objects based on the differences between their features and their spatial relationships. Interacting with virtual Environment is basically used for selecting objects in the scene graph. It is not directly connected to any input device. In addition, a VR system might want to use different metaphors for selection in different modes

II. Object Picking

Picking is an important interaction technique in graphics applications. Things which are inside the Virtual Environment are known as objects. The objects computer-generated stereo images 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 picking one object from the 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.

Virtual environments maintain a certain degree by using only a ray with appropriate starting and ending point. We have to provide the eye direction, choose two end points of the pick segment. Picking the correct 3D object is to find the direction of the picking shape and the virtual world coordinates of the starting point. The user clicks on the screen at point A. The object that is picked is to determine as there is only one along the projected vector indicated by the dashed line. A simple casting of the vector into the scene graph will reveal what object has been selected and our pick system will return a reference to it. A glove or a 3D mouse can be used to pick virtual objects and move them to different positions. In virtual environments, basically Object picking is performed by bounding box checks or collision detection, and taking the position of a virtual hand. Use Object picking to identify the objects on the screen that appear near the cursor. 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 image plane.

Commonly known approaches to collision detection perform in real time only when applied to facetted models. 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 alternative approach to 3D is to register virtual objects on the surfaces, using either overhead or back projection. The user can then interact with virtual objects by using traditional tools, such as a pen, or specifically designed physical icons, which are tracked on the surface using a variety of sensing techniques. The physical objects can then be used for interactions on the table, e.g. by manipulating them; we can select and move virtual objects. By linking multiple projection surfaces, and using traditional computer devices, for example laptop computers, to interact with virtual objects.

For an interaction between a virtual human and a real one, there is no possibility of transferring data structures, and image understanding methods are required to provide the virtual human with a perception of the real human's behavior. 3D devices like a Data Glove allow the real human to communicate any geometric information to the virtual one. A transparent pick cylinder with a variable radius is a straightforward solution. It is perfectly symmetric in 3D. Since the pick cylinder can intersect several topological entities of one or different object, simply selecting the nearest one usually does not conform to the user's expectation and makes it nearly impossible to pick elements with a small extent, i.e. vertices with respect to edges and with respect to faces.

Since this default behavior does not always fit to the user's intention, there is an additional pick filter, which allows the user to specify which topological element type should be considered by the pick procedure. If the pick ray is still bent while all but one user released the object we slowly transition back into the normal single-user manipulation mode, i.e. into a straight pick ray. It is quite clear that straightening the pick ray, keeping the object selected, would cause an unintended movement of the object. Deselecting the object automatically seems not useful. Picking can be used for more than just mouse selection. The pick shapes can be located anywhere and can be moved around within your scene. For example, you could use a pick shape in front of a person to detect potential obstacles and stop them from bumping into things.

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.

B. Picking object using pick cube

Picking may be either one checks for collisions between the cube and the topological entities geometry or one calculates the point of the topological entity's geometry that is next to the center of the pick cube and checks if the distance between the nearest point and cube center is smaller than the current pick distance .Only the second approach promises exact results, but requires time-consuming nearest-point calculations. To minimize calculations, a multi-level bounding box check is introduced and described in the following. Whenever we pick or click one object which is inside the bounding box, it enlarges the bounding box with the radius of the pick cylinder and checking if the center of the pick is inside the enlarged bounding box(see Fig 1). This trick is used to avoid the calculation whether the pick cube intersects or touches the bounding box.

The bounding volume must be much simpler an object than the object it bounds. Cylinders and prisms seem to be simple. However, they are seemed to be too useful, even with static scenes, probably because their geometry is already too complicated. For curved surfaces, such as splines, curved bounding volume such as spherical shells seems to be a good choice. Three-dimensional bounding volume is a useful tool of accelerating various geometric calculations including intersection detection. 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. 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 spaces, with sides parallel to the coordinate planes, that contains the object. The choice is highly dependent of the shape of the objects to be bounded. For elongated objects, possible solutions include bounding ellipsoids and cylinder. 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.

For an interaction between a virtual human and a real one, there is no possibility of transferring data structures, and image understanding methods are required to provide the virtual human with a perception of the real human's behavior.

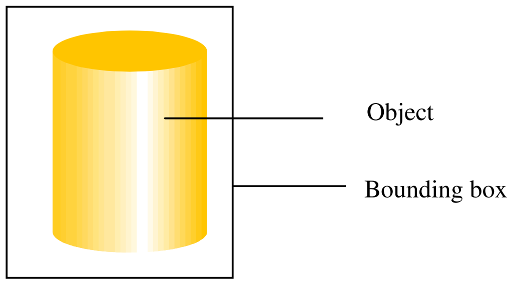

Figure 1: object within a bounding box

If the center of the cube is inside the enlarged object bounding box all object faces are checked(see Fig 2). Again, the procedure starts with enlarging the face bounding boxes by the pick radius and checks if the cursor center is inside the enlarged box.

The vertices are checked by simply comparing there position with the cursor position and the pick radius. 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. The shape of the virtual object is determined by their 3D surface, which can be described in many ways. Virtual objects have their surface composed of triangle meshes.

They are preferred because they use shared vertices and are faster to render. The Virtual Environment can contain all kinds of quite physical objects. To find the depth value, for each object, we can compute the maximum Z value within the area. If the maximum z value of one of these virtual objects is less than the minimum z values of the object obscures all other objects within the area boundary.

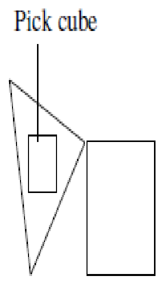

Figure 2. Triangle will be picked

Note that the nearest point over all checked faces is determined in addition. 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 pick cube position.

C. Pick Ray

It is the most basic picking shape and will pick an object in the same way as a penetrating a ray or radiation. The picking ray extends in the scene infinitely in a specific direction and an object intersected will be picked. Here we obtain local mouse and eye position from the view plane to 3D coordinate. The normalized ray direction that projects infinitely into the scene is then calculated. The closest intersected object is retrieved. In cases where the hand is used as a mouse, the data glove serves as an input source for a real-time, animated computer graphics model of the hand. The virtual hand moves around in response to the user moving his or her hand and may intersect with objects or may project a selection ray from an extended finger. The objective is to grab an object by pointing a finger at it.

The computer projects a ray from the extended finger; if the ray intersects the object it is captured. They constructed a system using data gloves which allowed users to generate three-dimensional curves for all applications. The user's extended index finger position was processed by corner detection and decimation algorithms to produce a series of piecewise cubic Bezier curves. In their earliest application, they treated the hand representation as a direct mouse equivalent—the user selected from control panels and from menus of B-splines by moving the hand analog until it intersected the desired choice. Generally speaking, using the hands as 3D mice is an extension of the direct-manipulation approach to interfaces. In this paper, we perform the exact object-space-based ray-cylinder intersection test in a geometry shader by taking advantage of its geometric processing capability.

D. Ray intersection Test in the cylinder

The ray tracing is the calculation of intersection between a ray and objects in the scene. All of this equation begins with the parametric equation for a line. A line from the point A=(x1, y1, z1) to point B=(x2, y2, z2).A is the starting point of the ray which represent view point location and B represent the pixel location and (i, j, k) is the direction the line is going. The point of intersection between the picking ray is defined by an origin point p and a normalized vector v and the equation is, p(t)=p+v*t and the cylinder, (p(t)-o)*d^2 where o is point on cylinder core, D is direction of cylinder and r is radius satisfies the equation

r^2*((B-A).(B-A))
(1).

As a result, the intersection information is obtained. Here the scalar, t, is a variable that is used to generate different points on the ray, where t-values of greater than zero are said to lie in front of the ray origin and so are a part of the ray and negative t-values lie behind it. Based on the distance from p<0 or |B-A|>0. Also, since the ray direction is normalized, a t-value generates a point on the ray that is t distance units away from the ray origin As the value of the t goes from 0 to 1 these equations generate the location of the ray. And the intersection point on the initial ray will always be greater than 1 if the entire object rendered is beyond where the image plane has been placed. The pick tool checks to find the nearest object in the picked region and have a result. If match occurs the exact object is retrieved. The intersection test is conducted in the view space and if it is passed, the x- and y- coordinate are 0 because the render target used in our algorithm is only one-pixel in size. The z-component is the depth value which is obtained by transforming the distance value into the projection space. This method maintain a certain degree by using only a ray with appropriate starting and ending point. We have to provide the eye direction, choose two end points of the pick segment. Picking the correct 3D object is to find the direction of the picking shape and the virtual world coordinates of the starting point. The user clicks on the screen at point A. The object that is picked is to determine as there is only one along the projected vector indicated by the dashed line. A simple casting of the vector into the scene graph will reveal what object has been selected and our pick system will return a reference to it. The object space based ray-cylinder intersection test is implemented in a highly parallelized geometry shader. When we applying the occlusion queries, only a small number of objects are rendered in subsequent layers, which accelerates the picking efficiency. Our algorithm can be easily integrated into existing real-time rendering systems.

The algorithm is as follows
Input: Cursor point and Virtual scene
Output: Ray intersection point, object id

Step 1: When we click on the screen, compute the picking ray intersection point and direction in the view coordinate system.

Step 2: Set the depth states to false. After the view, the bounding boxes consist of the visible objects. We pass a query for each object.

Step 3: The bounding boxes of all sub-objects whose corresponding query returns true. Again we issue a Boolean query for each sub-object during this rendering pass. then set the depth state to true.

Step 4: Render the actual cylinders whose queries have been returns true then pass query for all cylinder shape objects.

Step 5: If the occlusion query returns true, the picking information returns the one-pixel-sized target data; otherwise, no object is picked. Cylinder outside the view ray is discarded, and only the closest cylinder is needed.

Step 6: If the query passes, the cylinder with the minimal distance from the eye-point is picked and its intersection information can be retrieved from the target.

After this algorithm performs the object-space-based ray intersection test in the geometry shader, output a point with picking information if the cylinder is intersected. (see Fig: 4). The x- and y-components of the intersection point are set to 0, and the z-component is assigned as the depth value of the point. Output the picking information directly in the pixel shader. When the user clicks the mouse, the screen coordinates of the cursor are transformed through the projection matrix into a view-space ray that goes from the eye-point through the point clicked on the screen and into the screen.

Figure 3: Example of non immersive virtual environments and the cylinder shaped objects will be picked

A 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 un projected scene. This problem can be solved by determining all the objects that intersect the ray and performing point-in-solid inclusion tests to find out which object contains the specified 3D point. It may reduce the problem to a one-dimensional problem by determining the intersection intervals along the eye-ray for each object. Then, the object that is pointed at can be obtained with a point-in-segment inclusion test this is because when no bounding box intersects with the picking ray, our approach will not render the actual cylinders and return false directly.

For handling a cylindered shape, the problem is reduced to the computation of the nearest point on the shape, most of which require potentially time consuming point-and-shape or the ray-and-shape intersection algorithms. It allows accurate fine positioning in 3D-space.Reduced to a simple rounding of the device position to a point of a specified grid.

Advantages

- This algorithm is used to decrease the number of objects for testing the depth range.
- It will be totally application-independent.
- It can be easily developed by the user. The algorithm is also robust, since it does not depend on reading z-buffer values to determine which object contains the 3D cursor.
- Thus, it may be widely used in applications that represent complex polyhedral structures, making the work all easier
- It is used to decrease the cost of performing intersection tests between the surface and the eye-cursor ray, yet improving points matching.
- Reliability is very good.

When we analysis these two concepts in the view of selection and cost of performing an intersection test(see Fig:4) , picking using pick ray using shape algorithm is always greater than the average cost along the 3D picking algorithm using bounding volume.

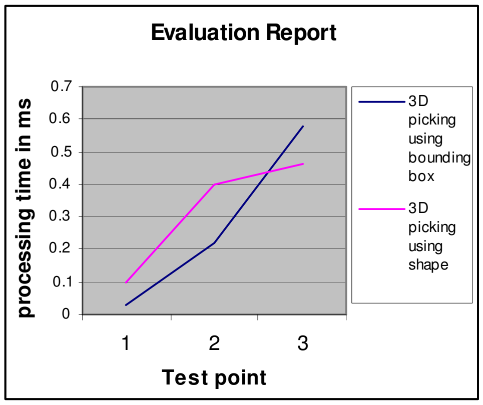

Figure 4. Average cost of performing an intersection test

III. CONCLUSIÓN

Although this has not been an exhaustive depiction of all possible interaction techniques, it provides a good exposure to many of the more common techniques currently in use. It is important to examine the interaction techniques that are available and determine those that are best suited for the tasks that need to be accomplished. We have presented an efficient algorithm for intersection tests between a picking ray and multiple objects in non immersive environment. The investigation resulted in new design guidelines that will allow for more usable design of non-immersive desktop, photo-realistic virtual environments. Furthermore, the study provides some new areas for future developments of usability evaluation methods. We discussed an implementation of the proposed algorithms, based on these guidelines, we present new forms of the pick ray casting and 3D picking using cylinder shape algorithm 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 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. In this paper, we perform the exact object-space-based ray-cylinder intersection test in a geometry shader by taking advantage of its geometric processing capability. The overall approach makes no assumptions about the object's motion and can be directly applied to all cylinder models.

IV. REFERENCES

[1] Cleber S.Ughini, Fausto R.Blahco “3D interaction Technique for accurate object selection in Immersive Environment”. Proceedings of the 1997 symposium on Interactive 3D

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

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

[4] M.Mine.”Virtual Environment Interaction Technique” (1995)

[5] Rolf Klein, Thomas Kamphans, The Minkowski Sum of two arbitrary polygons Institution fürInformatik2001.

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

[7] 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.

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

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

[10] Fosner, R. Real-Time Shader Programmin Morgan Kaufmann, 2002.

[11] T. Möller and B. Trumbore, “Fast, minimum storage Ray triangle intersection,” in Proceedings of the 32nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '05), Los Angeles, Calif, USA, July-August 2005.

[12] Alan wexelblatan “Approach to Natural Gesture in Virtual Environments”

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