Journal Homepage: - www.journalijar.com
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)
ISSN NO. 2320-5407
Article DOI: 10.21474/IJAR01/3022
DOI URL: http://dx.doi.org/10.21474/IJAR01/3022

RESEARCH ARTICLE

A REVIEW: COMPARISON OF LINE CLIPPING ALGORITHMS IN 3D SPACE.

Nisha.

Manuscript Info

Manuscript History

Received: 29 November 2016
Final Accepted: 27 December 2016
Published: January 2017

Key words:-
3D geometry, Line Clipping, Computer Graphics, 2D geometry.

Abstract

In computer graphics there are many line clipping algorithms which can be extended to 3D space. This paper is a review on Cohen-Sutherland and Liang-Barsky line clipping algorithms. Mostly all of the 3D line clipping algorithms involves basic three steps to check either a line segment lies completely inside the clipping volume or lies completely outside the clipping volume or crossing calculations when it is not entirely inside or outside. [1]

Introduction:-

In context of computer graphics, it is very essential to clip an area or a volume of interest which is to be displayed on the computer monitor. This division of interest is normally a rectangle or a general polygon in 2-Dimension and known as the clipping window. When it comes to 3-Dimensional clipping, a volume is used to extract a part from a 3-Dimensional scenario. Broadly the lines are clipped by clipping volume and it is known as polyhedron. And these Cuboids are extensively used as the clipping volume. 3-Dimensional clipping is one of the most essential processes such as video games, medical applications, computer aided design and many various applications. Occasionally it is important to dispose the data that is not within the visible range to avert the overflow of the internal registers of the display device.

Moreover, less memory consumption can be acquired by loading only a certain part o f a scenario to the memory by clipping unnecessary parts. Therefore, bettering the performance of the clipping algorithm has a better impact on performance of the comprehensive graphics system.

Cohen-Sutherland line clipping algorithm, Liang-Barsky line clipping algorithm, Cyrus-Beck line clipping algorithm and Nicholl-Lee-Nicholl line clipping algorithm are the traditional line clipping algorithms. The Cohen-Sutherland and the Liang-Barsky algorithms can be continued to 3-Dimensional clipping. Nicholl-Lee-Nicholl algorithm achieves lesser comparisons and divisions which makes it faster than others. Nevertheless, it is difficult to expand for 3-Dimensional clipping. [1]

Cohen-Sutherland Line Clipping Algorithm in 3D Space:-

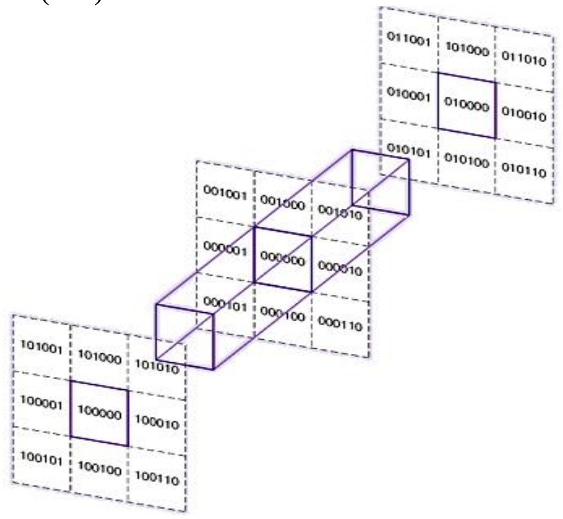
Cohen- Sutherland algorithm is one of the elementary and widely used clipping algorithms in computer graphics. This algorithm works very quickly in situations like the line segment is completely inside or outside of the clipping window. When the line segment is not lies inside or outside, this algorithm becomes ineffective due to repeated calculations. This algorithm can be easily extended to 3-Dimensional clipping, but space is required to divide into 27 mutually exclusive volumes, when the clipping volume is a cube or a cuboids. Again each volume is assigned a region code. Throughout the clipping process those region codes should be reserved in the memory. So in terms of

Corresponding Author:- Nisha.

space and time complexity Cohen – Sutherland algorithm lacks results for complex problems. According to Hearn and Baker [3] all most all the 3D line clipping algorithms involve three steps:

1. For a given line sector check if it lies entirely inside the clipping volume.
2. If it does not fall between check whether it lies completely outside the clipping volume.
3. Diversely, perform intersection calculations with one or more clipping planes.

The Cohen-Sutherland algorithm extends to 3D very simply. The outcode for 3D clipping is a 6-bit code. For a parallel canonical view volume the bits represent:
bit 1 - point above view volume (Y>1)
bit 2 - point below view volume (Y<-1)
bit 3 - point to the right of view volume (X>1)
bit 4 - point to the left of view volume (X<-1)
bit 5 - point behind view volume (Z<-1)
bit 6 - point in front of view volume (Z>0)



Nine line clipping regions of Cohen-Sutherland in 3D space

As in the 2D case, the line may be trivially accepted (both outcodes=000000), or trivially rejected ((code 1 AND code 2) not 000000). Otherwise line subdivision is used again. The points of intersection with the clipping edges are found using a parametric representation of the line being clipped. Consider the general line from P0 (x0, y0, z0) to P1 (x1, y1, z1). Its parametric representation is:

$$x = x0 + t(x1 - x0) \tag{1}$$
$$y = y0 + t(y1 - y0) \tag{2}$$
$$z = z0 + t(z1 - z0) \tag{3}$$
$$0 <= t <= 1$$

As t varies from 0 to 1 these equations give the coordinates of all the points on the line between P0 and P1.
EXAMPLE: To calculate the intersection of a line with the y=1 plane, y=1 is substituted into equation (2). Rearranging this gives:

$$t = \frac{1 - y0}{y1 - y0} \tag{4}$$

Substituting (4) into (1) and (3) yields:

$$x = x0 + \frac{(1 - y0)(x1 - x0)}{y1 - y0} \tag{5}$$

$$z = z0 + \frac{(1 - y0)(z1 - z0)}{y1 - y0} \tag{6}$$

As all the values in equations (5) and (6) are known, these two equations can be solved to find the x and z coordinates of the point of intersection. Using this information the line can be divided in two. [6]

Liang-Barsky Line Clipping Algorithm in 3D space;-

Consider the parametric definition of a line:-

- $x = x_1 + u\Delta x$
- $y = y_1 + u\Delta y$
- $\Delta x = (x_2 - x_1)$, $\Delta y = (y_2 - y_1)$, $0 \leq (u, v) \leq 1$

Mathematically, this means

- $x_{min} \leq x_1 + u\Delta x \leq x_{max}$
- $y_{min} \leq y_1 + u\Delta y \leq y_{max}$

Rearranging, we get

- $-u\Delta x \leq (x_1 - x_{min})$
- $u\Delta x \leq (x_{max} - x_1)$
- $-v\Delta y \leq (y_1 - y_{min})$
- $v\Delta y \leq (y_{max} - y_1)$

In general: $u * p_k \leq q_k$. There are many cases, if $p_k = 0$ then line is parallel to boundaries and if for the same k, $q_k < 0$, reject the line otherwise accept the line. If $p_k < 0$ then line starts outside this boundary and $r_k = q_k / p_k$, $u_1 = \max(0, r_k, u_1)$. If $p_k > 0$ then line starts outside this boundary and $r_k = q_k / p_k$, $u_2 = \min(1, r_k, u_2)$. And if $u_1 > u_2$, the line is completely outside. It can easily extend to 3D by adding equation for $z = z_1 + u\Delta z$ and 2 more p's and q's. [5]

Conclusion:-

In this paper I defined Cohen-Sutherland and Liang-Barsky line clipping algorithms in 3-Dimensional space. Three-dimensional objects are clipped against a bounding volume as opposed to a clipping rectangle. Specifically, when clipping in 3D space we need to consider the intersections of lines and planes or polygons and planes as opposed to the clipping of lines against lines, which is the case with 2D space. A simple technique often employed to reject objects completely outside the clipping volume and to accept objects entirely inside it is to create a bounding sphere for the object and to subsequently test its edges against that of the view volume. A bounding volume, or bounding box, is simply a volume containing a geometric object. Cohen-Sutherland is the simplest line clipping algorithm but the Liang-Barsky algorithm is more efficient, since intersection calculations are reduced. [2]

References:-

1. R. Kodituwakku, K. R. Wijeweera, M. A. P. Chamikara, "An Efficient Line Clipping Algorithm for 3D Space", International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 5, May 2012.
2. Abhishek Pandey1, Swati Jain2, "Comparison of Various Line Clipping Algorithms for Improvement", International Journal of Modern Engineering Research (IJMER) Vol.3, Issue.1, Jan-Feb. 2013.
3. Donald Hearn, and M. Pauline Baker, "Computer Graphics, C Version", 3 edition, pp. 226-230, December 2004
4. Rogers DF. "Procedural elements for computer graphics". New York: McGraw-Hill, 1985. P.111–87.
5. https://en.wikipedia.org/wiki/Liang%E2%80%93Barsky_algorithm
6. https://www.cs.helsinki.fi/group/goa/viewing/leikkaus/lineClip.html