An Introduction to Different Image Space Methods for Hidden Surface Removal Algorithms: An Overview

Narinder Kumar
Instructor,
S.B.S.B.M. University College Sardulgarh,
Punjabi University Patiala,

ABSTRACT

In the real world we see different realistic objects. When we transfer these real objects on computer or 2D coordinate systems, there are parts of objects which are hidden or can say not visible directly to human eyes. We remove these hidden surfaces with the help of different methods or algorithms. In this paper we will discuss those different types of algorithms which are used as per the requirement of the different types of application and purpose. I found it very interesting while working on it and challengeable. The basic purpose of this paper is to provide basic terms and knowledge of hidden surfaces removal methods for learner.

I. INTRODUCTION

Hidden surface are those which are not visible directly. In real world when we saw objects and there is required to project it coordinates system. Then to display this realistic scene, the major concerns to remove those parts of objects which are not visible from a chosen viewing position. Normally when we see objects, the parts which are backside or at the rear end are not visible. We can see only the front part. The part which is not visible is called hidden surfaces or hidden edges. The algorithms or methods are chosen on different aspects such as constraints, complexity of scene, available instruments etc. Hidden surface removal algorithms are also known as visible surface detection methods and can be classified into two categories which are given below:

![Hidden surface removal methods](image)

Fig. Types of Hidden Surface Removal Methods

1. **Object Space Methods**

These types of algorithms are implemented in physical coordinate system in which the object is defined. In object space methods the main focus on comparisons between those parts which are visible from specific position and which are not. Back face Removal algorithms, Binary Space Position (BSP) Tree, area Sub-Division (Quad trees and Octrees)) etc is included in this type of algorithms.

2. **Image Space Methods**

Screen coordinate system is used to implement by image space method. Each point is considered for the detection of visible surface on the projected plane. This is type of discrete method. Display resolution effects the accuracy calculation. If the display resolution is changed then re-calculation is required. Implement aspects which are used in hidden surface removal methods are sorting and coherence.
Coherence: It is the result of local similarity. Efficiency of sorting is increased using coherence. Advantage of regularity in the scene is taken by coherence.

b. Sorting: To compare the distance between different surface from the view plane.

The different image space algorithms are given below:

i. Z-buffer Algorithm (Depth Buffer)
ii. A-Buffer Algorithm
iii. Scan Line Algorithm
iv. Painter's Algorithm (Depth Sorting)

II. DIFFERENT TYPES OF ALGORITHMS

Here we will discuss the different algorithms can be used to remove hidden surface. First of all we will discuss Z-Buffer algorithm.

1. Z-Buffer Algorithm

Z-Buffer algorithm is also called Depth Buffer algorithm and it is developed by "Edwin Catmull " in 1947. The approach which is used by this algorithm is image space approach. This algorithm can be implemented on both hardware and software. It is easiest and simplest method. Each pixel position is compared on projection plane in this algorithm. In this Method Object's surface is processed indivisibly. For both types of surface Planer and non planer is suitable this algorithm but it is best for planer surface than non planer, because depth value can be calculated easily. The depth value is measured along z-axis; due to this name of the algorithm is Z-buffer algorithm. The value of z coordinate is zero on screen coordinate system 2D (x, y) while views as 3D (x, y, z). Buffers which are used are given below:

a. Depth Buffer: Z-Value or depth of each pixel is stored in this buffer. Initially value is set to be zero for each.
b. Refresh Buffer: Intensity value of each pixel (x, y) is stored in this buffer.

![Fig. Plane at different depth along a same project line](image)

**Steps in Algorithm:** Steps which are followed in Z-Buffer algorithm are given below

1. Initialize the depth buffer for all position
   Depth (x, y) = 0
2. Initialize the refresh buffer for all pixel
   Refresh (x, y) = Intensity of background.
3. Calculate the depth value (Z-Value) for each pixel (x, y)
4. Compare the depth value of all objects with already stored value. If newly calculated (Z value) is greater than previously stored value then update both buffer otherwise do nothing
If \( Z \)-value > depth \((x, y)\) then
Set depth \((x, y)\) = \( Z \)-value
Set Refresh \((x, y)\) = Intensity \((x, y)\)
End if

**Depth Calculation:** The equation which is used for depth calculation is
\[
Ax + By + Cz + D = 0
\]
\[
Z = \frac{-Ax - By - D}{C}
\]

For \( x+1 \),
\[
Z' = \frac{-A(x+1) - By - D}{C} = Z' = Z - \frac{A}{C}
\]

2. **A- Buffer Algorithm**

A- Buffer algorithm is extended version of Z-Buffer algorithm with additional information. It Stores the intensity for each pixel of multiple surface using link list. A-Buffer algorithm is also known as area – averaged, anti-aliased, and accumulation buffer method. Each position in the A-Buffer has two fields.

a. Field of Depth: Field of depth stores the +ve / -ve values.

b. Field of Intensity: Field of Intensity of each pixel for all surfaces.

If the value of depth is greater than zero \((d>0)\) means there is no link list but if there is negative value of depth \((d <0)\) means multiple surface intensity is exists.

3. **Scan Line Algorithm**

This is another type of image space algorithm. In this method full line is scanned rather than a pixel like z-buffer. Before process the next line we have to group the polygons which are overlapping on intersecting at given scan line. Z- Value is calculated during scanning a line for each pixel to determine nearest value. Tables which are used in this algorithm are edge table (ET), Polygon Table (PT), Active Edge table (AET) and flag value to on or off.
a. Edge Table (ET): All Non Horizontal edges of all polygons which are projected on the view plane are stored in edge table (ET). Horizontal edges are ignored. Information stored in this table are
   i. X- Coordinates
   ii. Y- Coordinates
   iii. x- increments
   iv. Polygons identification numbers.

b. Polygons Table (PT): Surface material property, coefficients of plane, other surface doter, points of edge table etc re contained in this table.

c. Active Edge Table (AET): Information of those edge which are crossing the scan line is stored on active edge table (AET) to keep the track of active edge. Values are stored in ascending order of X.

d. Flag: Flag value is used to on or off when line move left to right or right to left respectively.

Steps in algorithm for scan line:

1. Establish the required data structure such as Polygon table (PT), Active Edge Table (AET), Edge Table (ET), Flag, Intensity depth etc.

2. Repeat for all scan lines
   a. Update AET by sorting ET against scan line y values.
   b. Scan across, using background color until an IN flag on.
   c. When two or more flags are on, do depth sort and use intensity for surface n with minimized z value.
   d. Use coherence of plane to repeat for next scan line.

4. Painter’s Algorithm

   It is also type of image space approach. The painter algorithm is also known as depth sorting algorithm and priority fill algorithm. Like painters, the area is painted which seen farthest, background colors and then second farthest and so on, due to this it is called painter’s algorithm. In this method depth is sorted thus various surfaces are from inside to outside due to this, it is known as depth sorting algorithm. The reason behind to call priority fill algorithm is that priority is given to the farthest visibility order or depth priority list of polygons is created using polygons depth sort.

   Painter’s algorithm uses both operations of image space and object space method. The function performed by depth sorting or priority fill or painter's algorithm are given below:
   a. Surfaces are sorted in descending order of depth.
   b. Surfaces are scan converted.
Various surfaces/polygons may overlap then comparisons are needed, different methods are available for this, the commonly used method is minimax method. Different test can be applied for each surface that overlapped which is given below. If any test is true then reordering required. Tests are listed in ascending order of difficulty.

1. The boundary rectangles in the XY plane for the two surfaces do not overlap.
2. Surface S is completely behind the overlapping surface relative to the viewing position.
3. The overlapping surface is completely in front of S relative to the viewing position.
4. The projection of the two surfaces onto the view plane does not overlap.

**Steps in Algorithm:**

1. Sort the all surface in decreasing order according to the z value.
2. Apply scan convert in all surfaces start with greatest depth.
3. Compare depth of overlapping surfaces to determine visible surface.
4. Update refresh buffer.
5. Repeat the process for all surface.
6. If there exist overlapping apply the test.

**Reverse painter Algorithm:**

In this method nearest object is painted after that second nearest and so on. This technique is opposite of painter's method.
III. CONCLUSION

In above we have discussed various image space method's algorithm for display a 3D view on 2D coordinate systems with removing hidden surfaces. The simplest technique is Z-buffer for identify visible surface with two buffers. A-buffer provides additional information. The importance of the method is depend on the particular application. Depth sorting is suitable for few surfaces. Scan line is suitable where several surfaces are exists. It is possible to combine various visible surface detection methods. Special hardware are required to implement these algorithms.

REFERENCES

- http://www.viz.tamu.edu
- http://www.netgraphics.sk
- http://www.tutorialpoint.com
- Computer Graphics: Theory into Practice by Jeffrey J. McConnell
- Introduction to Commuter Graphics by Krishnanmurthy
- Computer Graphics And Multimedia By A.P.Godse, D.A.Godse
- Introduction To Computer Graphics & Multimedia, 2E By Arup Chattpadhyay Anirban Mukhopadhyay