MEDZINÁRODNÁ KONFERENCIA
CA.. TECHNOLÓGIE V STROJÁRENSTVE
CATS
Bratislava, 11.-12.október 1993

Raster Space Solution of Hidden Line Problem for Projected Grid Surfaces

T. Hrúz[1]and I. Považan[2]

This paper describes a new approach to the hidden line problem in computer graphics. We assume that a 3D visible grid surface scene is given. A raster space (image space) method for effective solving of hidden line problem is defined for this well established instance of hidden line problem. The solution of the visibility problem relies mainly on a special Bresenham-like linear interpolator and on a new enumeration of edges resp. facets.

1 Introduction

The hidden line problem is one of the major algorithmic problems in computer graphics. The roots of this problem can be followed to the early sixties (see [3, 1]). One of the well established instances of this problem is a hidden line elimination in projected grid surfaces. Grid surfaces arise when we want to render the graph of a function defined on a rectangle in 3D Euclidean space. An example of such surface is in figure 1. The el-

[2]Tomáš Hrúz, Slovak Technical University, Faculty of Mechanical Engineering, Department of Automatic Control and Measurement Námestie Slobody 17, 812 31 Bratislava, Slovak Republic, Phone: +42-7-3594-571, 497193 e-mail: hruz@vm.stuba.sk Fax: +42-7-495315

[2]Ivo Považan, Institute of Control Theory and Robotics, Slovak Academy of Sciences, Dubravská cesta 9, 842 37 Bratislava, Slovak Republic, Phone: +42-7-3782985, e-mail: utrrpova@savba.savba.cs

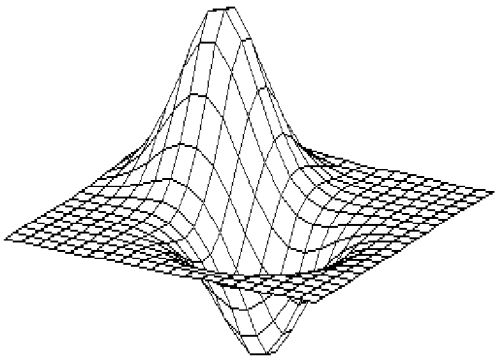

Figure 1:

The projected grid surface.

ements from which the picture consists are called *facets*. Facets are the images of grid elements of the function in 3D.

As was many times emphasized there are two fundamental approaches to the visible surface determination. The first approach is called object space approach and relies in various ways to represent objects in the scene by discrete combinatorial representation. The basic objects used are geometrically defined subsets of Euclid space R^3 (i.e. points, edges, polygons, polyhedra e.t.c.). The brute force method of this kind compares every object with other objects and finds the visible resp. invisible parts.

The second fundamental approach (the image space approach) is to solve the visibility in every pixel in image space. A straightforward way relies in examining all objects and determining which of them is the closest to the viewer on the projector passing through a given pixel. Image space methods are often realized in hardware. The Z-buffer method is a widely used representation of this class but it has a drawback that when the number of objects and the pixel resolution is large the computational cost can be very high.

The above classification is only a basic pattern. Many algorithms combine above approaches. Our algorithm is in essence an image space algorithm but the structures we use are more rich compared to the Z-buffer.

The classical work solving the hidden line problem for projected grid surfaces is that of Wright [5]. A more recent important contribution was due to Anderson [6]. This method works in object space and achieves linear complexity with respect to the number of facets. The complexity is established only experimentally. We think that it could be derived from more recent works of Overmars (see for example [2]).

Our method works in raster space. It also achieves linear speed with respect to the number of facets. Recently we have learned that similar algorithm was independently discovered by Wang and Staudhammer [7]

Our contribution consists mainly in the enumeration of facets and also in the method how we use interpolation algorithm.

The article is organized in the following way.

In section 2 we define the basic terminology. Section 3 overviews the algorithm and section 4 comments our development and suggests some further directions of research.

2 Grid Surfaces

Let we have a single-valued function f which maps from R^2 to R of the form $z = f(x, y)$. The cartesian product of two sequences of points $(x_1, \dots, x_n)$, $(y_1, \dots, y_m)$ define a grid in x-y plane of 3D Euclidean space. In every grid-point (x_i, y_j) we have a value of the function f: $z_{ij} = f(x_i, y_j)$. An image under function f of the elementary surface in x-y plane is a surface in 3D space. The projection of this surface in 3D space to the viewing plane is called *facet*. To define projection plane we must have a viewpoint $VP = (VP_x, VP_y, VP_z)$ with a viewing direction $VA = (VA_x, VA_y, VA_z)$. The points in 3D are projected to the projection plane. Finally the picture data are scan converted (approximated) from the projection plane coordinates to the device coordinates. The device coordinates resp. *raster space* is an integral lattice which models concrete raster devices as raster displays etc.

3 The algorithm overview

Our solution use an old idea of contour (floating horizon) but in a raster space sense. To obtain an effective algorithm we use two contours (horizons) in the raster space. They are represented with a simple data structure consisting from two vectors. The first maintains y-coordinates of the up growing contour in raster space and the second maintains y-coordinates of the down growing contour.

When the contour is defined in the raster space it is possible to test the contour during the interpolation. In our algorithm the Bresenham interpolator is modified so that during a point generation the visibility is tested against the contour and subsequently the contour is modified. This is another basic idea of our approach.

In the following paragraphs we overview the main steps of the algorithm.

1. The point with the smallest y-coordinate in the projection plane is found from the four corner points of the rectangular area $(x_1, y_1), (x_n, y_1), (x_n, y_n), (x_1, y_n)$ on which the function is defined. The algorithm has four branches according to the point which is at the bottom. This point is the starting point for the enumeration of edges resp. facets.

2. The direction for the enumeration of edges is determined according to the relation between the angles α_1, α_2 (see figure 2). The enumeration always follows the smaller angle. In the situation in figure 2 the enumeration follows the numbering of the edges. This method is expressed in the main loop

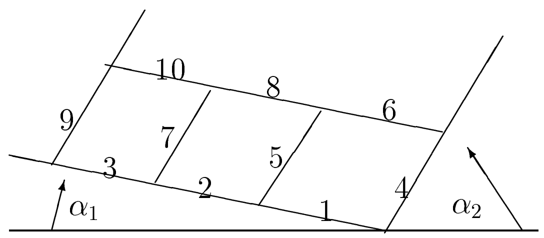

Figure 2:

The figure illustrates how the facets resp. edges are enumerated.

of the algorithm.

3. When the appropriate edge to be drawn is identified the edge is projected to the projection plane and the end-points are scan-converted to device coordinates. Then the algorithm use a modified Bresenham interpolator. This interpolation is done two times. Firstly with respect to the upper contour and then with respect to the lower contour. In figure 3 we give an algorithmic primitive for such interpolation in one octant.

4 Conclusions

In this section we comment our development and suggest further direction for research.

- It is intuitively straightforward to see that the presented algorithm is correct. But to prove this assertion would need further investigation.

- Because the algorithm visits every edge resp. facet only once the complexity is linear with respect to the number of facets. In the case of raster space algorithm it is worthwhile also to express the complexity as a function of two variables: number of facets and resolution. We suggest that this function $C(s, N) < O(s.N)$ is strictly bounded with $N.s$ and equals $O(s.\sqrt{N})$.

- The algorithm was implemented in optimized version for image processing system and it proves to be very useful and efficient. We have used it also for animation of projected grid surfaces.

References

[1] James D. Foley, Andries Van Dam, Steven K. Feiner, John H. Hughes, Computer Graphics, Principles and Practise, second edition, Addison Wesley, Reading, Mass., November 1991.

[2] M. Sharir, M. H. Overmars: A Simple Output-Sensitivity Algorithm for Hidden Surface Removal, ACM Transactions on Graphics, Vol. 11, No. 1, January 1992, pp. 1-11.

```
10if(abs_dx > abs_dy)
20  {
30  inc_1=2abs_dy;
40  inc_2=2(abs_dy-abs_dx);
50  dd=2abs_dy-abs_dx;
60  abs_dx++;
70  if(x_point>d_x)
80      {
90      exchange=y_point;
100     y_point=d_y;
110     d_y=exchange;
120     exchange=x_point;
130     x_point=d_x;
140     d_x=exchange;
150     }
160 if(y_point < d_y)
170     {
180     while(abs_dx) /* x+1 y+1 */
190         {
200         point(x_point,y_point)
210         x_point++;
220         if(dd < 0)
230             {
240             dd+=inc_1;
250             }
260         else
270             {
280             dd+=inc_2;
290             y_point++;
300             }
310         abs_dx–;
320         }
330     }
340 }
```

Figure 3:
The algorithmic primitive for interpolation in the first octant. The function point(..) in line 200 is different for the up-growing contour and the down-growing contour. Those forms are in figure 4 resp. in figure 5.

```
point(x_point,y_point)
{
        if (y_point ≤ contour_up[x_point])
            {
            contour_up[x_point]=y_point;
            point_xy(x_point,y_point);
            }
}
```

Figure 4: The algorithmic primitive for the function point() which tests the visibility and updates the contour for the up-growing contour.

```
point(x_point,y_point)
{
        if (y_point ≥ contour_down[x_point])
            {
            contour_down[x_point]=y_point;
            point_xy(x_point,y_point);
            }
}
```

Figure 5: The algorithmic primitive for the function point() which tests the visibility and updates the contour for the down-growing contour.

[3] I. E. Sutherland, R. F. Sproull, and R. A. Shumacker: A characterization of ten hidden-surface algorithms, Computer Surveys, Vol. 6, No. 1, March 1974,pp. 1-55.

[4] J. M. Bajkovskij, V. A. Galaktionov, T. N. Michajlova: Grafor. Grafitcheskoje rozshirenie fortrana, Moskva, Nauka, 1985.

[5] T. J. Wright: A Two-Space Solution to the Hidden Line Problem for Plotting Functions of Two Variables, IEEE Transactions on Computers, Vol. c-22, No. 1, January 1973,pp. 28-33.

[6] D. P. Anderson: Hidden Line Elimination in Projected Grid Surfaces, ACM Transactions on Graphics, Vol. 1, No. 4, October 1982,pp. 274-288.

[7] S. CH. Wang and J. Staudhammer: Visibility Determination on Projected Grid Surfaces, IEEE Computer Graphics & Applications, July 1990,pp. 36-43.