MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

WORKING PAPER 126 August 1976

TWO SIMPLE ALGORITHMS FOR DISPLAYING ORTHOGRAPHIC PROJECTIONS OF SURFACES

Robert J. Woodham

ABSTRACT

Two simple algorithms are described for displaying orthographic projections of surfaces. The first, called RELIEF-PLOT, produces a three-dimensional plot of a surface $z = f(x,y)$. The second, called SHADED-IMAGE, adds information about surface reflectivity and source illumination to produce a grey level image of a surface $z = f(x,y)$.

Both algorithms demonstrate how a systematic profile expansion can be used to do hidden surface elimination essentially for free.

Work reported herein was conducted at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defence under Office of Naval Research Contract number N00014-75C-0643

1. INTRODUCTION

This paper describes two simple algorithms for displaying orthographic projections of surfaces. The first, called RELIEF-PLOT produces a three-dimensional plot of a surface $z = f(x,y)$. Figure 1-1 illustrates RELIEF-PLOT applied to the (polar) function $f(R,\theta) = K_1 \cdot SIN(R)/R$. The second, called SHADED-IMAGE, adds information about surface reflectivity and source illumination to produce a grey level image of a surface $z = f(x,y)$. Figure 1-2 illustrates SHADED-IMAGE applied to the (polar) function $f(R,\theta) = K_2 \cdot R \cdot SIN(N\theta)$ (case $N = 7$).

Both algorithms use a simple profile expansion technique to do hidden surface elimination. The key word here is simple. In fact, the principle motivation behind my initial implementation of the RELIEF-PLOT algorithm was to demonstrate that hidden surface elimination could be done very cheaply.

The idea underlying the approach taken to hidden surface elimination is not new. It follows as a direct consequence of the approach to hidden surface elimination described in [1]. Basically, in addition to computing the image coordinates (u,v) of an object point (x,y,z), one also retains the depth value (ie. distance from the viewer) associated with that point. Then, the decision whether or not to update a particular image point depends upon whether or not the associated depth value is less than the depth value associated with any previous use of the image point.

RELIEF-PLOT and SHADED-IMAGE use a uniform profile expansion technique to eliminate the need to explicitly calculate depth. By systematically expanding profiles away from the viewer, the decision to display a particular image point (u,v) reduces to a simple comparison of its row coordinate v with the largest row so far displayed in column u.

Subsequent to my implementation, I was pointed to a Japanese paper[2] which produces a three-dimensional contour plot of a surface $z = f(x,y)$ using a similar technique for hidden surface elimination.

2. RELIEF-PLOT

The key to the RELIEF-PLOT algorithm is the ability to do hidden surface elimination essentially for free. How this comes about will follow from the presentation of the algorithm itself. Let me begin the discussion by establishing the appropriate conceptual framework.

Suppose we wish to generate a two-dimensional display of a surface $z = f(x,y)$. In order to calculate the image point (u,v) which corresponds to a particular surface point (x,y,z), we can think of transforming the point (x,y,z) from its object space representation into an eye space representation (x',y',z'). (The eye coordinate system has its origin fixed at the viewpoint and its positive Z' axis pointed in the direction of view.) Second, we can generate two-dimensional display coordinates by projecting the eye coordinate point (x',y',z') onto the plane of the display. If we define the transformation as:

RELIEF-PLOT of $f(r,\theta) = K_1 \frac{\sin(r)}{r}$

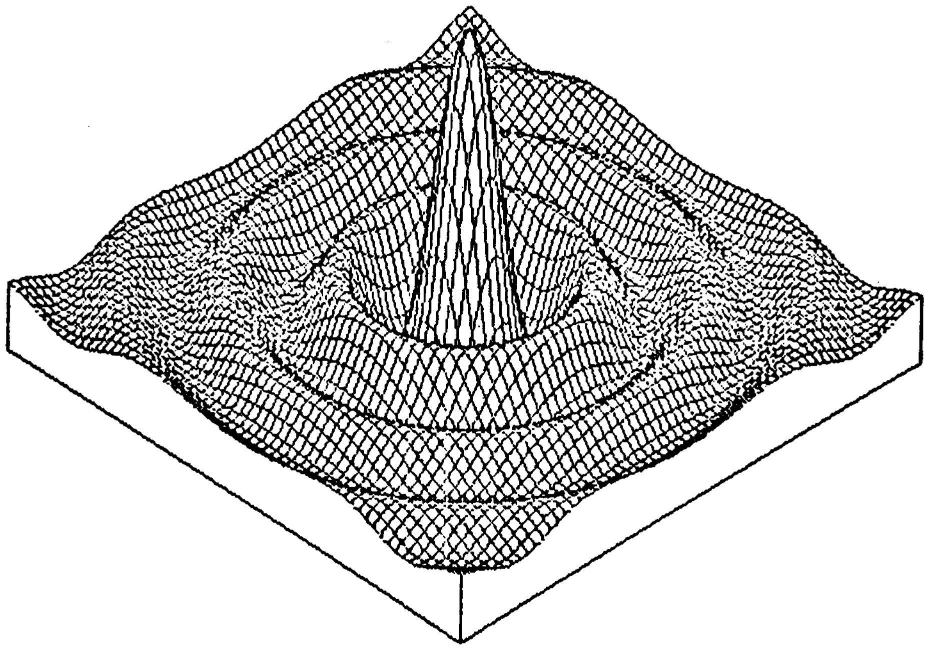

figure 1-1

SHADED-IMAGE of $f(r,\theta) = K_2\, r \sin(n\theta)$ (case $n=7$)

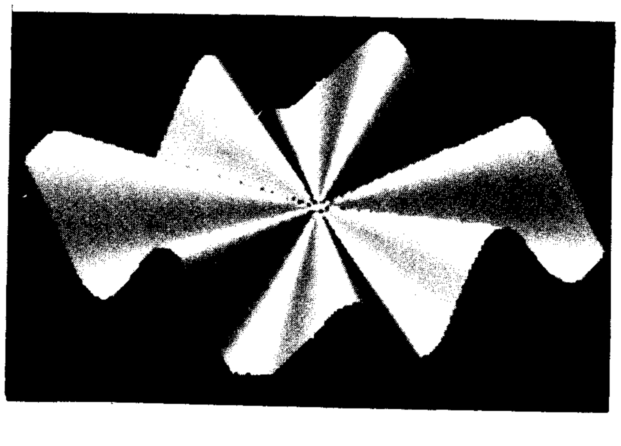

figure 1-2

Suppose we have image coordinate axes U and V. For simplicity, let the origin in object space map into the origin in image space and align the Z axis with the V axis as shown in figure 2-1. The projection from object space to image space is characterized by the angles θ_1 and θ_2 (figure 2-1) that the projected X and Y axes make respectively with the U axis. There are still two degrees of freedom here. RELIEF-PLOT requires that $\theta_1 = \theta_2$. We can drop subscripts and simply call this angle θ. The correspondence between the image space angle θ and the object space angle ϕ is then given by:

$$\mathrm{SIN}(\phi) = \mathrm{TAN}(\theta)$$

(where $0 \leqslant \phi \leqslant \pi/2$ so that $0 \leqslant \theta \leqslant \pi/4$)

Now, in object space, the viewing direction is given by the vector:

$$(1,-1,\sqrt{2}\cdot\mathrm{TAN}(\phi))$$

The projection transformation we have defined is given by:

$$u = (x + y)/\sqrt{2} \qquad v = z\cdot\mathrm{COS}(\phi) + (y - x)/\sqrt{2}\cdot\mathrm{SIN}(\phi)$$

The reader can verify that this transformation defines an orthographic projection.

If we uniformly scale the U and V axes by the constant $\sqrt{2}$, we can simplify the computation even further. The actual projection transformation used in RELIEF-PLOT is given by:

$$u = x + y \qquad v = \sqrt{2}\cdot z\cdot\mathrm{COS}(\phi) + (y - x)\ \mathrm{SIN}(\phi)$$

Now, $\sqrt{2}\cdot\mathrm{COS}(\phi)$ and $\mathrm{SIN}(\phi)$ can be precomputed so that, given an object space point (x,y,z), the projection transformation requires only three addition/subtraction operations and two multiplications. Note that we go directly from object space coordinates to image space coordinates. We need not explicitly perform a coordinate transformation from object coordinate space to eye coordinate space. More importantly, we need not explicitly calculate a depth value for each point (x,y,z).

RELIEF-PLOT expands profiles of $z = f(x,y)$ over a finite support plane. Denote this support plane by W where,

$$W = \{(x,y) \mid x_{min} \leqslant x \leqslant x_{max}\ ;\ y_{min} \leqslant y \leqslant y_{max}\}$$

The algorithm then is as follows:

1. Create a one-dimensional array of length the width of the display screen. Call it THRESHOLD. Initialize it to zero.

2. Initialize the THRESHOLD array by expanding the two initial boundary profiles $B_1 = \{(x,y_{min}) \mid x_{min} \leqslant x \leqslant x_{max}\}$ and $B_2 = \{(x_{max},y) \mid y_{min} \leqslant y \leqslant y_{max}\}$.
   That is, for all (x,y) in B_1 and B_2, let $\mathrm{THRESHOLD}(x + y) = \sqrt{2}\cdot f(x,y)\cdot\mathrm{COS}(\phi) + (y - x)\ \mathrm{SIN}(\phi)$
   There is no hidden surface problem here since both boundary profiles are guaranteed to

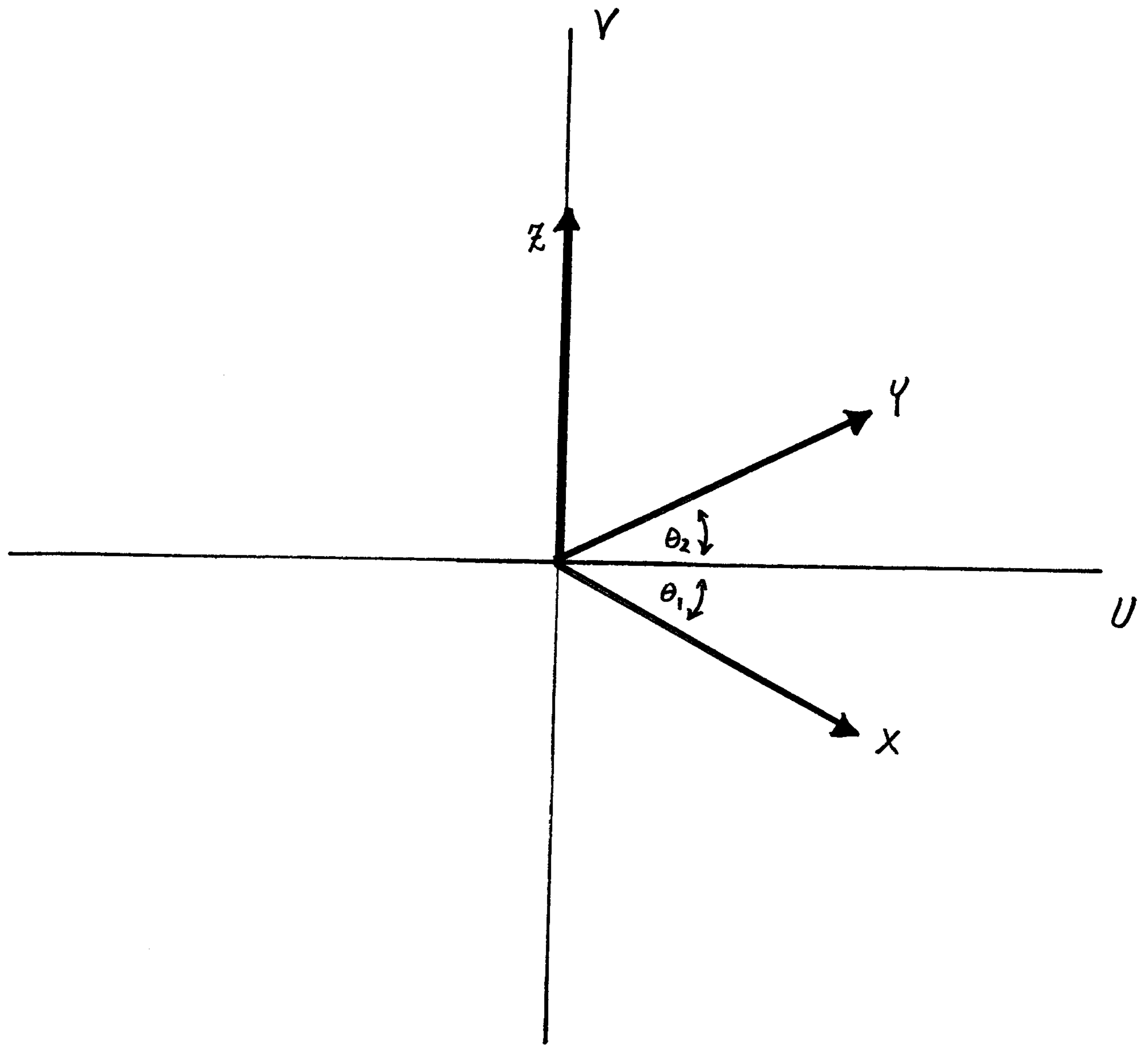

figure 2-1

be visible.

3. Choose some expansion resolution (call it RES). Then, expand profiles increasing in Y and decreasing in X. That is:

```
(DO X XMAX (- X RES) (< X XMIN)
    (DO Y YMIN (+ Y 1) (> Y YMAX)
        (PROG (U V)
              (SETQ U (+ X Y)
                    V (ROUND (+$ (*$ ROOT2-COS-PHI (F X Y))
                                 (*$ SIN-PHI (FLOAT (- Y X))))))
              (COND ((> V (THRESHOLD U))
                     (DISPLAY-POINT U V)
                     (STORE (THRESHOLD U) V))))))
```

As described above, RELIEF-PLOT simply point plots the appropriate image coordinates. The algorithm could easily be modified to connect adjacent display points with a straight line. In fact, this is how figure 1-1 was generated.

4. If the cross-hash pattern of figure 1-1 is desired, repeat steps 2 and 3 above but, this time, expand profiles decreasing in X and increasing in Y. That is, replace step 3 by:

```
(DO Y YMIN (+ Y RES) (> Y YMAX)
    (DO X XMAX (- X 1) (< X XMIN)
        (PROG (U V)
              (SETQ U (+ X Y)
                    V (ROUND (+$ (*$ ROOT2-COS-PHI (F X Y))
                                 (*$ SIN-PHI (FLOAT (- Y X))))))
              (COND ((> V (THRESHOLD U))
                     (DISPLAY-POINT U V)
                     (STORE (THRESHOLD U) V))))))
```

3. SHADED-IMAGE

The same simple profile expansion technique used in RELIEF-PLOT can be trivially extended to generate a grey level image of a surface $z = f(x,y)$. Basically, all that has to be done is to add a calculation of an appropriate intensity value for each image point (u,v) displayed by RELIEF-PLOT.

It is not the point of this paper to discuss details of possible shading algorithms. Suffice to say that one can model surface reflectivity by a function relating incident, emergent and phase angles (see Horn[3]). The object space representation of these angles is easy to determine. We have already shown that the viewing direction is given by the vector $(1,-1,\sqrt{2}\cdot\mathrm{TAN}(\phi))$. The direction of source illumination can be arbitrarily assigned. Surface normals are determined from the $z = f(x,y)$ representation.

The only point of concern is the fact that, in expanding profiles over the XY plane, there is no guarantee that an intensity value will be computed for each appropriate image point. In fact, this will most assuredly not be the case. As illustrated by figure 1-1, many surface points are "missed" in the expansion.

Expressing the problem more precisely, it will often be the case that a new computed row value v will bump the value of THRESHOLD(u) by more than one. The problem then becomes what to do about the image points in between. A simple interpolation of intensity values doesn't do the trick since the points between which one interpolates typically cross real depth discontinuities.

One might try to expand the original profiles at such a high resolution that each value in the THRESHOLD array never gets updated by more than one image point at a time. At best, this solution would be so computationally wasteful as to defeat the simplicity of the technique.

It turns out that one can solve this problem quite effectively. Suppose that in addition to updating the THRESHOLD array with each new row value, we keep track of the (x,y) point that gave rise to that value. Then, when we find a new row value, v_{new}, that bumps the old value, v_{old}, by more than one, we can interpolate <u>in the XY plane</u> between (x_{old},y_{old}) and (x_{new},y_{new}). This is guaranteed correct since all image points between (u,v_{old}) and (u,v_{new}) must arise from points between (x_{old},y_{old}) and (x_{new},y_{new}).

SHADED-IMAGE implements a quick binary interpolation algorithm for computing the appropriate intermediate intensity values. Basically, the idea is to compute the row value at the midpoint of the line between (x_{old},y_{old}) and (x_{new},y_{new}). The intensity corresponding to this row value is stored in image point (u,v). Now, if this intermediate row value bumps the old value by more than one, the interpolation algorithm is recursively applied to solve for intensity values between (u,v_{old}) and (u,v). Similarly, if this intermediate row value is more than one less than the new one, the interpolation algorithm is recursively applied to solve for intensity values between (u,v) and (u,v_{new}). In this way, intensity values are calculated for image points between (u,v_{old}) and (u,v_{new}).

A sufficient condition to guarantee that this algorithm computes the correct intermediate intensity values is that there be no more than one local extremum in $z = f(x,y)$ between (x_{old},y_{old}) and (x_{new},y_{new}) (ie. that the original profile expansion samples f(x,y) at least as often as the Nyquist rate).

In conclusion, I simply wish to point out that SHADED-IMAGE makes no attempt to handle effects due to mutual illumination or shadows. Mutual illumination is difficult to handle quantitatively (see HORN[3] for some specific examples). On the other hand, finding surface points that are shadowed by other portions of the object, is equivalent to solving the hidden line problem from the point of view of the light source.

4. REFERENCES

[1] Catmull, E., A Subdivision Algorithm for Computer Display of Curved Surfaces, UTEC-CSc-74-133, University of Utah, December 1974.

[2] Murai, S., Ohbayashi, S., Tateishi, R., "Three Dimensional Representation of Landscape", Journal of Institute of Industrial Science, Vol 27 No 5, Univerisity of Tokyo.

[3] Horn, B. K. P., "Image Intensity Understanding", AI-MEMO 335, AI Laboratory, MIT, August 1975.