Implementation of Edwin Catmull's hidden Surface Algorithm with anti-aliasing with Results

Fariba Khan
khanfari@oregonstate.edu
Oregon State University

1 Introduction

In this project, part of the paper titled ”A hidden Surface Algorithm with anti-aliasing” authored by Edwin Catmull[Catmull 1978] has been implemented. In this paper, an anti aliasing technique addressing hidden surface in image has been introduced to reduce symptoms of aliasing effectively when data for the pixel is complicated. To display an image which will always have much higher resolution than corresponding display, it need to be sampled at discrete points corresponding to pixels. These image space objects have sharpness and details (higher frequencies) that cannot be possibly reproduced due to lower sampling rate than required (Nyquist-shannon theorem). It is the attempt to sample that detail at discrete points in the image that causes the problem. Hence, the image should be filtered to filter these fine details before it can mess up attainable lower frequencies as aliases after which sampling can be done.

One simple filter that is easy to implement analytically is two dimensional Fourier(box) window. Convolving with this filter in frequency domain is equivalent to integrating i.e. taking average visible intensities over the area of each pixel in spatial domain. To correctly integrate intensities of only visible objects at a pixel,a hidden surface algorithm at every pixel is also required. Therefore, what we need is an analytic continuous solution to both hidden surface algorithm and the filter convolution and this paper provides an algorithm for that.

Keywords: aliasing, clipping, computer graphics, filtering, hidden surface removal, sampling

2 Related Work

The removal techniques of hidden surface have improved in the last couple years before this paper was published. [Sutherland et al. 1974] paper provides coherence to the development. Several new algorithms have come along [Myers 1975][Hamlin and Gear 1977][Weiler and Atherton 1977], each adding new insight into the ways that we can take advantage of coherence for some class of objects to facilitate display.

On the other hand, until that time, progress in anti-aliasing has been slower. The more obvious effects of aliasing, like jagged edges, could be fixed up with ad hoc techniques. Methods for anti-aliasing have been presented in [Catmull 1974][Crow 1977][Shoup 1973]. Frank Crow's dissertation was devoted to the topic and the results were published in [Crow 1977].

3 Proposed Technique

The first step of the technique presented here is to find surfaces of all polygons lying on each pixel. This pixel surface extraction algorithm requires a clipping algorithm as well. Then, the intensity of that pixel is calculated as an area weighted average of the only visible surface.

First, we need to find all polygons that overlap a particular scanline and clip away everything that doesn't overlap it. Since the scanline has the width of one pixel we are left with a list of very narrow horizontal polygons. The next step is to clip off pieces of those narrow polygons on the scanline that overlap a particular pixel. For efficiency, if a piece comprises several pixels, this algorithm treats middle part as solid run and clipped off two ends and find if they are solid or irregular. If the closest polygon completely covers the pixel then its intensity value can be taken as full intensity, otherwise area of each visible pieces need to be calculated to compute area weighted average intensity for that pixel.

This paper proposes three algorithms for Finding Visible Surfaces, Clipping and Complicated Pixel Intensity Integrater (for more than two intersecting polygons inside a single pixel) respectively. I have implemented first two algorithms in this project which works for up-to two intersecting polygons inside a pixel and only one of these intersecting polygons can be irregular (not covering the entire pixel);

3.1 Finding Pixel Surface

The Hidden-Surface algorithm [Catmull 1978]

Input: List of all polygons in image

Output: Intensities for all pixels in display

- Step 1:Sort all polygons on highest y value.
- Step 2: Initialize active polygon list to be empty.
- Step 3: Repeat for each scanline:
  1. Add polygons from y-list that enter this scanline to active polygon list
  2. Initialize the x-bucket to be empty the scanline array to background
  3. Loop through each polygon in active polygon list
     (a) Call Clipping algorithm to clip off of each polygon the piece that lies on the current scanline.
     (b) Replace polygon in list with polygon that has piece clipped off
     (c) Call Clipping algorithm to clip off two end polygons at the ends at the pixel boundaries. The two end polygons are called irregular pieces if not covering the whole pixel and solid pieces if so. The centers are called solid pieces.
     (d) The pieces are sorted into the x- bucket according to the leftmost pixel covered.

4. Initialize the z-list to be empty.
5. Repeat for each pixel across the scanline:
   (a) Sort every entry at the current x position of the x-bucket into the z- list.
   (b) Evaluate the z-list if not empty:
      i. If a solid piece, get its color else if an irregular piece is in front of a solid piece then find the area of the irregular piece over the pixel to weight the two colors else call the pixel integrater (not implemented in this project) to get color
      ii. Write the color into scanline array.

3.2 Clipping

The Clipping algorithm [Catmull 1978]

Input: List of all polygons in image

Output: Intensities for all pixels in display

- Step 1: A polygon is list of points P1, P2, ...Pn.
- Step 2: Call Pn the previous point. Determine which side it is on.
- Step 3: Loop though each point, called current point.
  1. If current point on side A then,
     (a) If previous point on A side then, copy current point to polygon A.
     (b) If previous point on B side then,
        - Calculate intersection of line with edge formed from current point and previous point.
        - Copy calculated point to A and B polygons.
        - Copy current point to A polygon.
  2. If current point B side then,
     (a) If previous point on B side then, copy current point to B polygon.
     (b) If previous point on A side then,
        - Calculate intersection of line with edge formed from current point and previous point.
        - Copy calculated point to A and B polygons.
        - Copy current point to B polygon.
  3. Call the current point the previous point.

There is also a pixel integrater algorithm for complicated pixels that I haven't implemented.

Results

All shapes are rendered with scanline conversion without depth test. Clipping algorithm works very effectively to clip away part of the polygons that are not on the particular scanline and also to clip off of each polygons the piece that is inside a pixel. With a sorted z list for all pieces of polygons lying on a particular pixel, it can determine visible part of each pieces thus eliminating completely occluded polygons pieces. Pixel intensity is then area weighted average intensities which blurs the jagged edges giving the smooth illusion. If we look at Figure 1 and 2, we can observe this blurred effect along the edges of rectangles and triangles. Also, in pixels with intersecting polygons, red (farthest polygon's color) doesn't show through when it's completely occluded by the green, otherwise red and green color is weighted by their respective area inside that particular pixel. In figure 3, after eliminating aliasing effect, edges look very nice without any saw tooth effect like before. Implementation of this algorithm works perfectly for removing hidden surfaces of irregular shapes as well (Figure 4).

References

CATMULL, E. 1974. A subdivision algorithm for computer display of curved surfaces. *Technical report UTEC-CSs-74-133 University of Utah*.

CATMULL, E. 1978.

CROW, F. 1977. The aliasing problem in computer- generated haded images. *CACM* (Novemenber).

HAMLIN, G. J., AND GEAR, C. 1977. Raster- scan hidden surface algorithm techniques. *Siggraph proceedings*.

MYERS, A. J. 1975. An efficient visible surface program. *Ohio State University, Computer Graphics Research Group, report to NSF* (July).

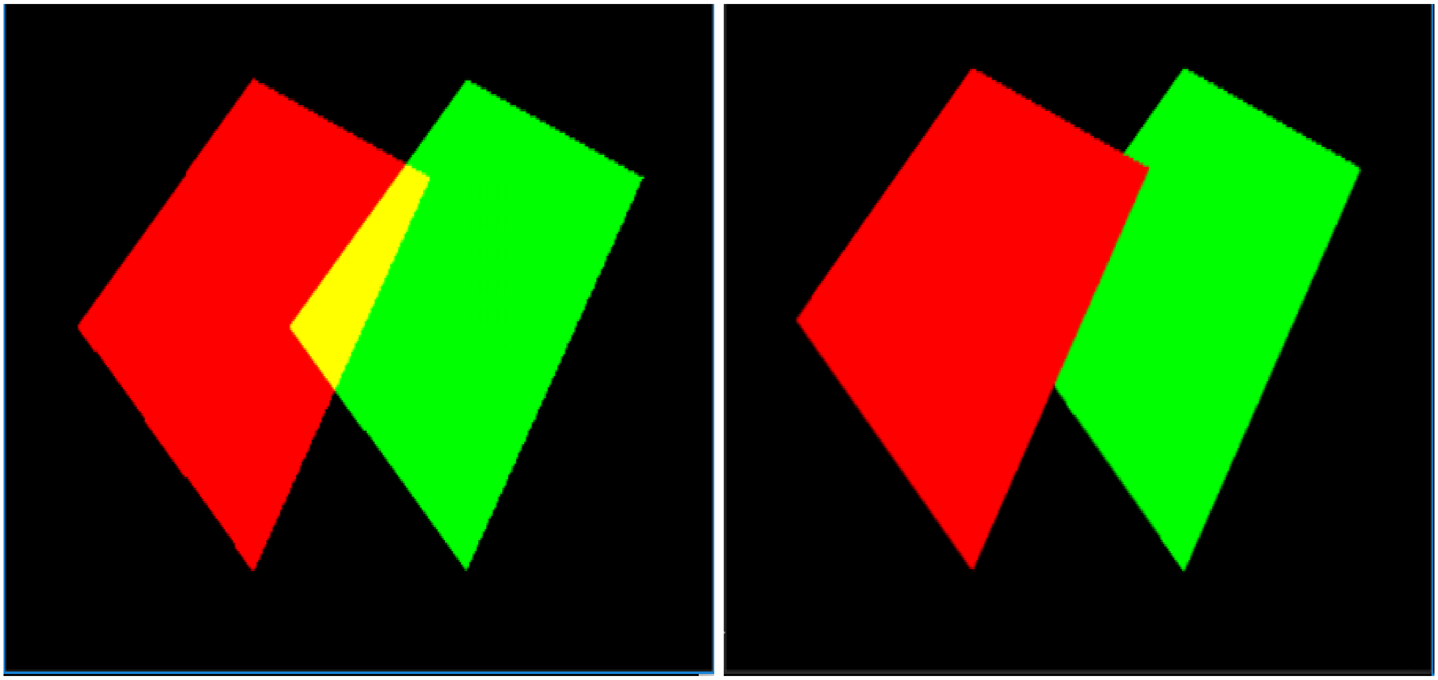
SHOUP, R. 1973. Some quantization effects in digitally-generated pictures. *Society for In- formation Display*.

SUTHERLAND, I., AND G.W., H. 1974. Reentrant polygon clipping. *CACM* (January).

SUTHERLAND, I., SPROULL, R., AND SCHUMACKER, R. 1974. A characterization of ten hidden surface algorithms. *ACM Computing Surveys* (March).

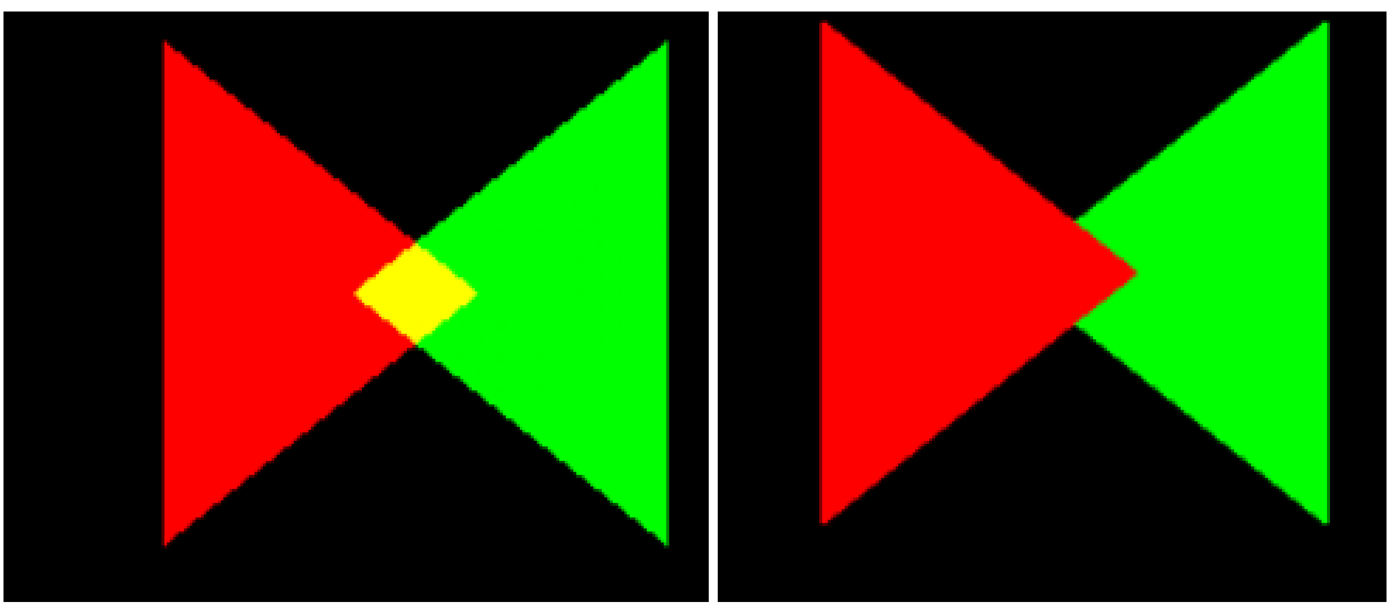
SUTHERLAND, I.E., O. . 1973. Polygon sorting by subdivision: A solution to the hidden-surface problem,. *Unpublished manuscript, Also public lecture at University of Utah*.

WEILER, K., AND ATHERTON, P. 1977. , hidden surface removal using polygon area sorting. *Siggraph proceedings*.



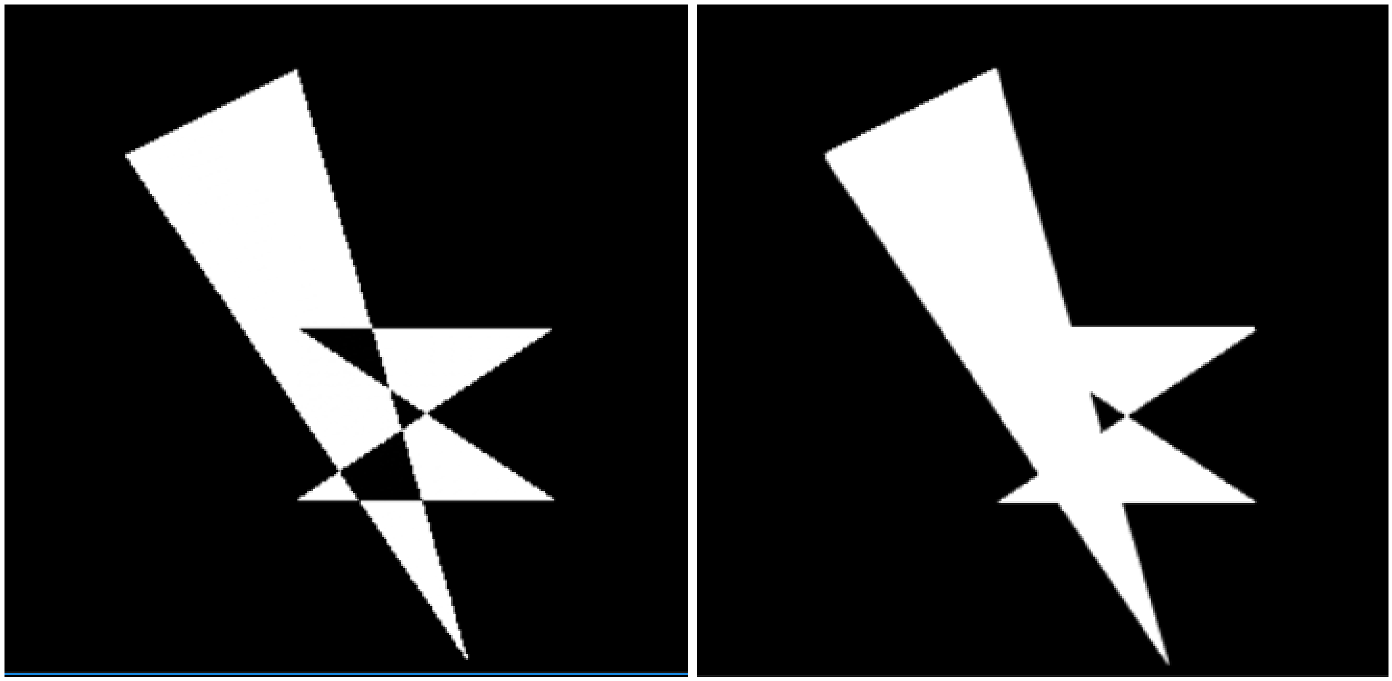
$(a) renderedwithout proposedmethod$ $(b) renderedwithproposedalgorithm$

Figure 1: Results of proposed method implementation for two intersecting rectangles with different depth (red one being closest)



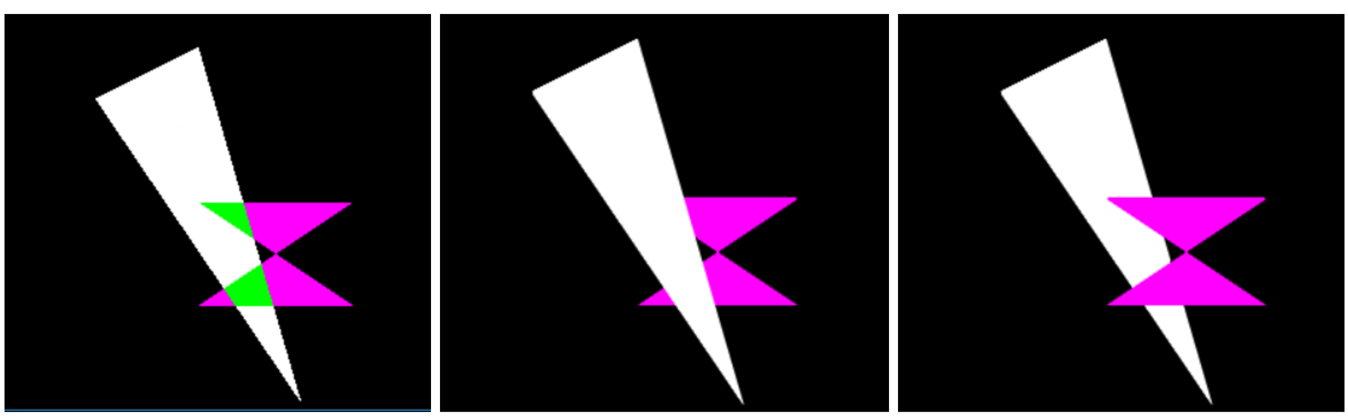
$(a) renderedwithout proposedmethod$ $(b) renderedwithproposedalgorithm$

Figure 2: Results of proposed method implementation for two intersecting triangles with different depth (red one being closest)



(*a*)*renderedwithoutproposedmethod* (*b*)*renderedwithproposedalgorithm*

Figure 3: Results of proposed method implementation for two monochrome intersecting shapes with different depth



(*a*)*withoutproposedmethod* (*b*)*withproposedalgorithm* (*c*)*withproposedalgorithm*

Figure 4: Results of proposed method implementation for two intersecting shapes with different depth (a)without proposed method magenta shape behind (b) with proposed algorithm magenta shape in front