Edge Inference with Applications to Antialiasing

Jules Bloomenthal

Computer Graphics Laboratory
New York Institute of Technology
Old Westbury, New York

Abstract

An edge, when point-sampled for display by a raster device and not aligned with a display axis, appears as a staircase. This common aliasing artifact often occurs in computer images generated by two- and three-dimensional algorithms. The precise edge information often is no longer available but, from the set of vertical and horizontal segments which form the staircase, an approximation to the original edge with a precision beyond that of the raster may be inferred. This constitutes a smoothing of the staircase edge.

Among other applications, the inferred edges may be used to reshade the pixels they intersect, thereby antialiasing the inferred edges. The antialiased inferred edges prove a more attractive approximation to the real edges than their aliased counterparts.

Presented here are algorithms for the detection and smoothing of edges and the filtering of an image in accordance with the inferred edges.

General Terms: Algorithms.

Keywords and Phrases: Antialiasing, Edge Inference, Filtering.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism--visible line/surface algorithms; I.4.1 [Image Processing]: Digitization--sampling; I.4.6 [Image Processing]: Segmentation--edge and feature detection.

1. Introduction

In 1961 Herbert Freeman published "Computer Processing of Line Drawings" in which he introduced the chain coding scheme for representing and processing edges [Freeman 1961; Freeman 1970]. Subsequent research with line representation has been applied widely to the problems of artificial intelligence [Duda 1973; Rosenfeld 1969] but the ability to construct edges from an image has rarely been applied to computer graphics.

One such application was presented in a 1978 dissertation by Garland Stern; as part of an animation system, outlines of animated figures were extracted from an image and then smoothed.

In 1974, Bruce Baumgart, in a seminal study of geometric modeling, extracted and smoothed line drawings from real world images of an object as an initial step in creating a computer model of the object [Baumgart 1974]. Eight years later, Scott Roth, using some of Freeman's techniques, extracted chain-encoded line structures from a rastered description of three-dimensional objects and, after processing the chain-codes, produced quite convincing line drawings of the objects [Roth 1982].

The extraction of edges may also be used to modify the original image. In 1977, Lance Williams demonstrated a technique ("dekink") for estimating edges of a two-dimensional region and then filtering the horizontal and vertical segments of the edges in two separate iterations [Williams].

It is not difficult to imagine other applications such as deriving closed contours from computer images in order to interpolate images or convert image data into a contour form (such as topographical maps).

Techniques for extracting edges from images are presented by Rosenfeld and Kak in their text Digital Picture Processing [Rosenfeld 1969]. Approaches for smoothing the edges are suggested as well.

2. Edge Inference and Antialiasing

Aliasing of objects (manifested by staircase edges, disappearing objects, or disconnected lines) has been minimized by a number of techniques including super-sampling [Crow 1981], adaptive super-sampling [Whitted 1980; Roth 1982], and analytical integration for certain kinds of objects [Catmull 1978]. There are also techniques which address aliasing in texture [Blinn 1976; Dungan 1978; Feibush 1980; Williams 1981], and highlighting [Whitted 1980; Roth 1982].

All of these techniques filter before sampling the image, but, with the exception of the computationally expensive super-sampling technique, none of the techniques works well with z-buffer based programs which render multiple types of objects (such as polygons, quadrics, and patches). Neither do they work well with "paint" programs or other programs which generate images in which edge information is not explicitly available.

Thus, the most intriguing use of edge inference may be to minimize the aliasing artifacts near object edges in the point-sampled computer generated images typically produced by z-buffer algorithms or paint programs. Filtering the image along inferred edges enjoys the advantages of post-processing flexibility and the concentration of processing along edges, where aliasing is usually most apparent.

The process of edge smoothing will be described below in two sections. The first describes a method for finding and tracking those rastered edges created by the point-sampling process. The second section discusses the smoothing of the rastered edges. Filtering an image along the smoothed edges is presented in a third section.

3. Edge Tracking

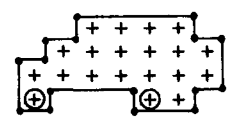
The tracking process is performed upon a frame of discrete, single-valued regions. It is usually a simple matter to create such a frame during the computer generation of an image. Alternatively, rastered images may be processed into a collection of regions of interest by various segmentation methods [Rosenfeld 1969].

A scan-line method similar to one developed by J. W. Butler [Butler 1963] was used to produce starting points for all the regions in a frame. A scan-line edge-detector could also be used [Agrawala 1977].

3.1. Tracking a Region

The "region" containing pixel (x,y) may be defined as the set of pixels connected to pixel (x,y) through a four-connected path of pixels, all the same value as pixel (x,y). A "four-connected path" is a connection or a series of connections from a pixel to any of its perpendicular neighbors.

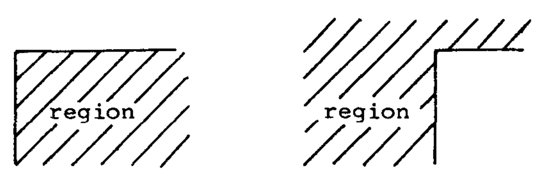
In this discussion, "starting point" means a pixel located at a lower left corner of a region (see Figure 1). Given any point within a region, the determination of a starting point requires only a small amount of processing. Once the starting point is known, a region's boundary may be tracked, forming a closed circuit when the starting point is again reached. Rather than the chain coding representation of Freeman, the boundary is represented here as a sequence of corner locations in frame buffer coordinates (Figure 1). This has the advantage of readily accessible positional data for the boundary.



+ represents a pixel center
. represents a corner
o is a potential starting point

Figure 1: A 4-Connected Region

The tracking process is presented by the flowchart in Figure 3. The tracking proceeds in a clockwise direction about the region; the ability to determine the concavity or convexity of a corner is required (see Figure 2). The algorithm ensures that any two corner points share either their x or y coordinate and that no three corners are collinear. A detail of the flowchart is given in Figure 4.



a convex corner a concave corner

Figure 2: Two Types of Corners

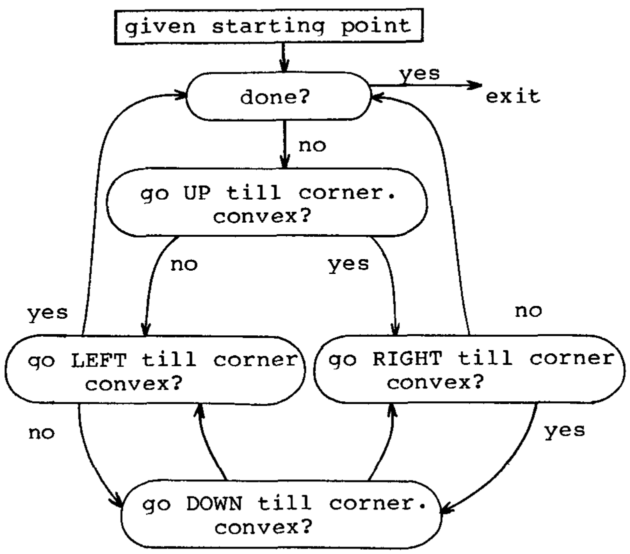


Figure 3: Flowchart for Edge Tracking

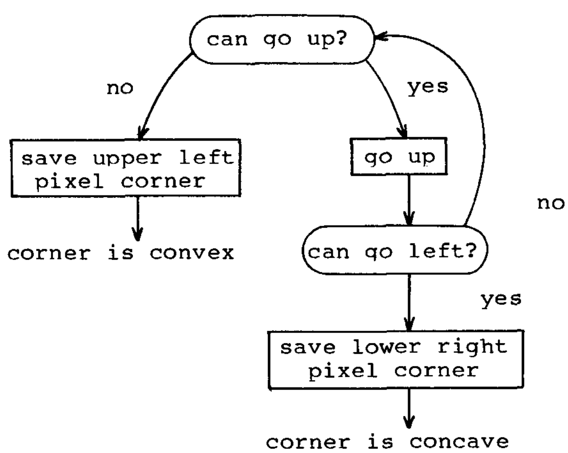


Figure 4: Detail for Figure 3

4. Edge Smoothing

Edges of surfaces in real space, when transformed to screen space, become aliased if the edge pixel values are not properly computed. This aliasing is manifested by "jaggies", sudden pixel-sized shifts in an otherwise straight line. Edge smoothing must disambiguate intended changes in direction from changes produced by jaggies.

A sudden pixel-sized shift, or "jag" is determined from four sequential corner points when the direction from point 1 to 2 is the same as from 3 to 4 and the distance from point 2 to 3 is one pixel unit. From a repeated jag pattern, Figure 5, left, a straight edge may be inferred with reasonable certainty. In Figure 5, right, the jag pattern changes midway and a change of slope is inferred.

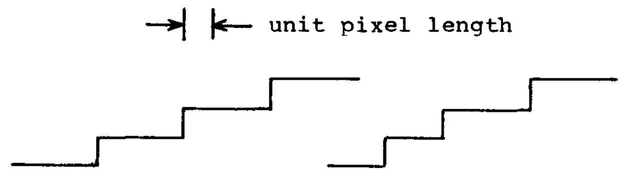


Figure 5: Jags

The edge inference algorithm cycles through each corner of a region. A first-in, first-out buffer of nine points is maintained which, for a given corner point, provides the positions of the four preceding and four following points, as well as the length and direction of the segments connecting the points. A series of tests is performed on this information to determine if and where a new endpoint should be created for the smoothed edge.

It is important to note that the inferred edge endpoints are given in floating point coordinates resolved beyond the raster. Further, the inferred edges are constrained to be within one pixel of the original, rastered edges.

5. Image Filtering

In this section "pixel" is used to mean the unit square surrounding the pixel center. Given the inferred boundary of a region, those pixels in the image which are intersected by the inferred edges are reshaded. In the present implementation, the percentage of the pixel covered by the region, as defined by the inferred edges, is first computed; it serves as a weighting coefficient for averaging the surrounding pixels. This is equivalent to a simple box filter; more complicated filters may be implemented as look-up tables with the table index a function of the distance from the inferred edge to the pixel center [Feibush 1980; Turkowski 1982].

Once the weighting coefficient is estimated, the pixel is reshaded as a linear combination of surrounding pixels averaged according to whether or not they belong to the region.

5.1. Computing the Estimate

Pitteway and Watkinson developed a method similar to Bresenham's line drawing algorithm which computes pixel area covered by an edge [Pitteway 1980].

Referring to Figure 6, one observes that the area covered in Pixel B is equal to that covered in Pixel A plus the shaded area. Thus, given an initial area, the areas of the remaining intersected pixels may each be computed with a single addition. At Pixel D, however, it is necessary to subtract the area of the black triangle from the new total. The area of the black triangle is then the initial area for Pixel E. This incremental technique is complicated by the need to accommodate endpoints not on a pixel boundary, or edges which re-enter a pixel.

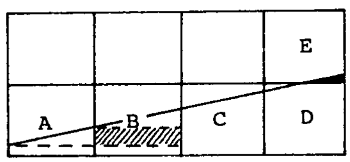


Figure 6: Computing Pixel Area

6. Conclusion

Proper antialiasing of computer generated pictures requires that the image be filtered before it is sampled for display on a raster device [Crow 1977]. Unfortunately, a high precision representation of the image is not always possible or practical with some hidden surface algorithms. Certain programs and input devices also produce aliased images.

Inferring, and then smoothing, an edge from an aliased image can produce convincing results, as shown in Figure 7. Figure 8 demonstrates one result of the post-process technique as well as a comparison with an image which exhibits no aliasing artifacts. Figure 9 presents magnified views of Figure 8. The post-process technique exhibits errors at some of the vertices as well as on the silhouette of the sphere. Further, the technique does not address other forms of aliasing such as small objects disappearing and thin, continuous lines breaking up.

It is worth noting that all the algorithms presented here operate in linear time with respect to the length of an edge. Figure 8, a 512 by 486 resolution image, was processed in 17 seconds with a VAX/780.

Some aspects of the technique are not limited to inferred edges. Filtering may occur along the true edges calculated by a hidden line eliminator program. This increases the accuracy of the edge position but usually limits the program to images generated using polygonal models. (Most hidden line eliminators work with polygonal surfaces only.) Conversely, antialiasing is not the only application for the inference of edges.

7. Acknowledgements

Lance Williams offered consistent support for this research, as well as many insightful concepts. Paul Heckbert provided critical reading of this paper as well as the model descriptions for Figure 8.

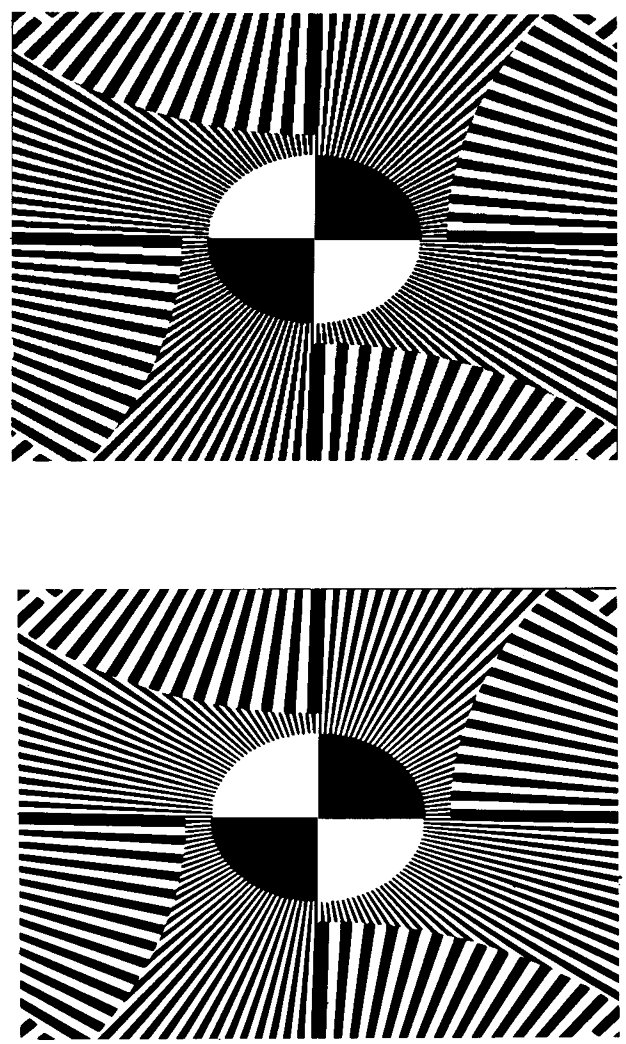
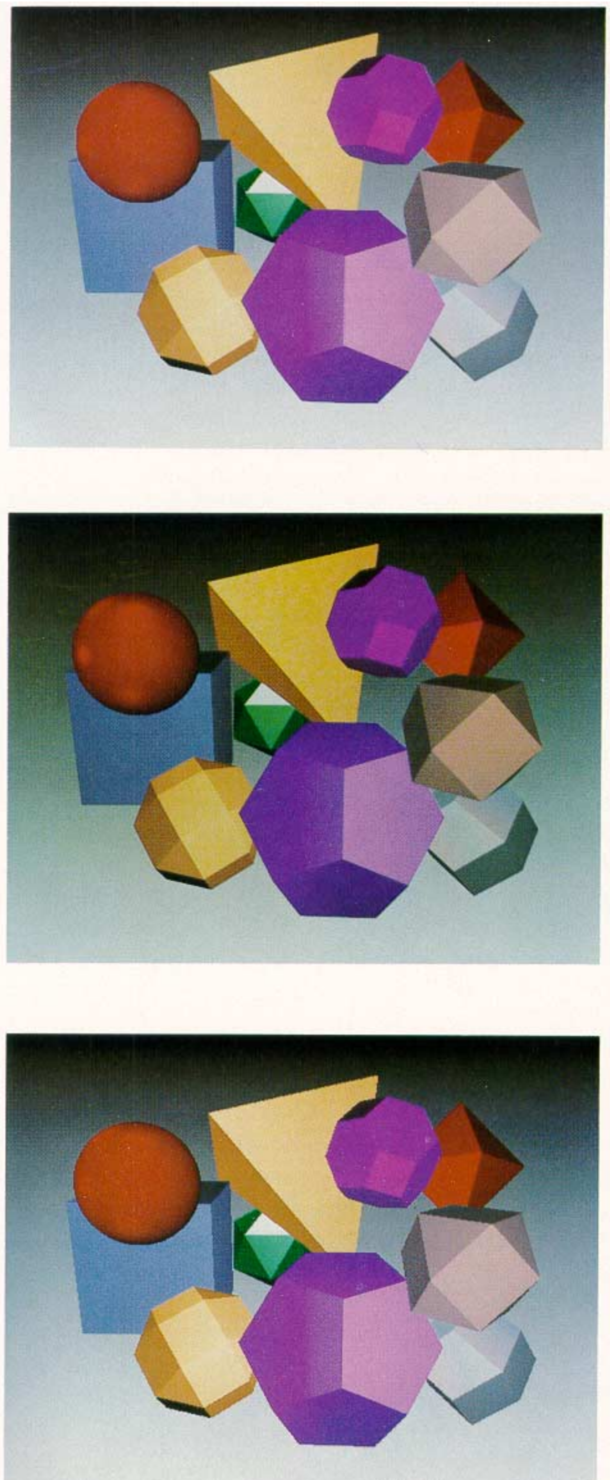
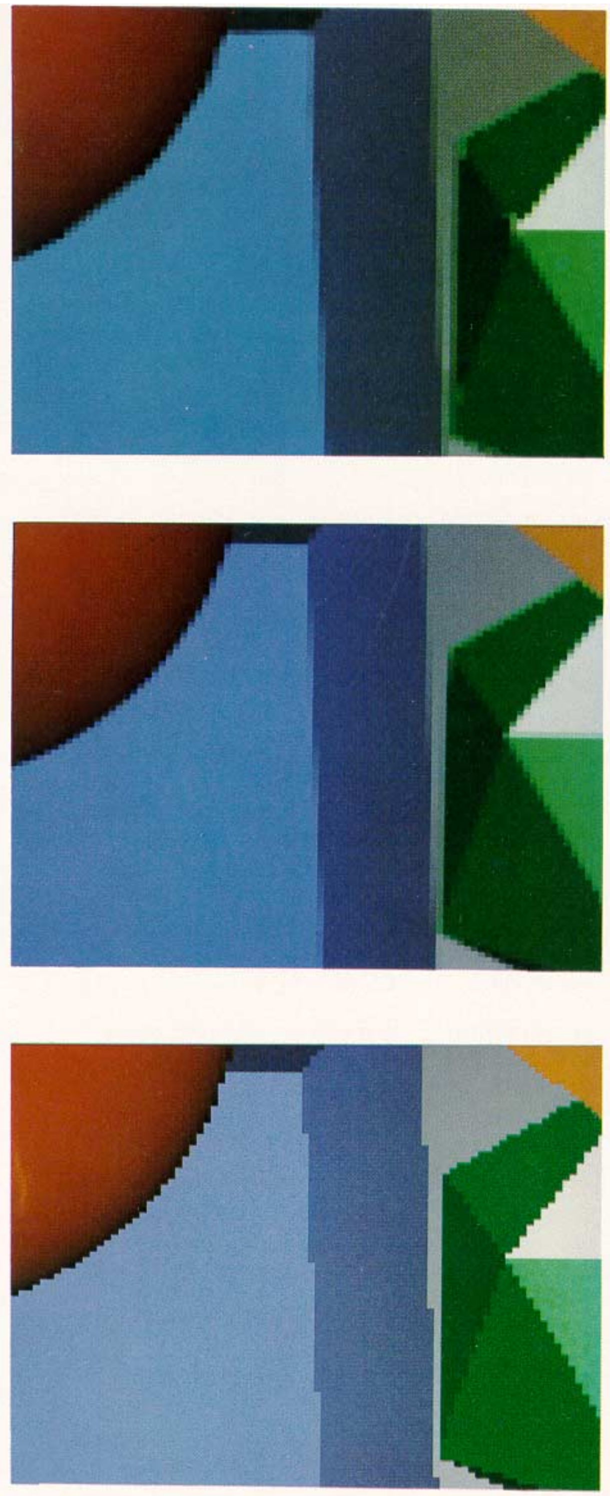


Figure 7: Test pattern before (top) and after (bottom) processing





Top: Antialiasing as a Post Process
Middle: Correctly Antialiased
Bottom: Without Antialiasing

Figure 8: Rendered Images, 512 Resolution

Figure 9: Rendered Images, Magnified

8. References

Agrawala, A. and Kulkarni, A., "A Sequential Approach to the Extraction of Shape Features", Computer Graphics and Image Processing, v.6, December 1977, pp. 538-557.

Baumgart, B.G., "Geometric Modeling for Computer Vision", Report STAN-CS-74-463, Stanford Artificial Intelligence Laboratory, Stanford University, 1974.

Butler, J.W., Butler M.K., and Stroud A., "Automatic Analysis of Chromosomes", Data Acquisition and Processing in Biology and Medicine, v.3, pp. 261-275, Pergamon, Oxford, 1963.

Catmull, E.E., "A Subdivision Algorithm for Computer Display of Curved Surfaces", Ph.D. Thesis, University of Utah, Computer Science Department, 1974.

Catmull, E.E., "A Hidden-Surface Algorithm with Antialiasing", Computer Graphics, v.12, August, 1978.

Crow, F.C., "The Aliasing Problem in Computer Generated Shaded Images", Communications of the ACM, v.20, November, 1977.

Crow, F.C., "A Comparison of Antialiasing Techniques", IEEE Computer Graphics and Applications, v.1, January, 1981.

Duda, R.O. and Hart, P.E., Pattern Classification and Scene Analysis, John Wiley and Sons, New York, 1973.

Dungan, W., Stenger, A., and Sutty, G., "Texture Tile Considerations for Raster Graphics`, Computer Graphics, v.12, August, 1978.

Feibush, E.A., Levoy, M., and Cook, R.L., "Synthetic Texturing Using Digital Filters", Computer Graphics, v.14, July, 1980.

Freeman, H., "On the Encoding of Arbitrary Geometric Configurations", IRE Trans. EC-10, pp. 260 - 268, June, 1961. lp Freeman, H., "Boundary Encoding and Processing", Picture Processing and Psychopictorics, B. Lipkin and A. Rosenfeld, editors, Academic Press, Inc., NY, 1970.

Pitteway, M. and Watkinson, D., "Bresenham's Algorithm with Grey Scale," Communications of the ACM, v.23, November 1980, pp. 625-626.

Roth, S.D., "Ray Casting for Modeling Solids", Computer Graphics and Image Processing, v.18, February, 1982, p. 109-144.

Rosenfeld, A. and Kak, A.C., Digital Picture Processing, Academic Press, New York, 1969.

Stern, G., "A System for Computer-Aided Keyframe Animation", Ph.D. Thesis, University of Utah, Computer Science Department, 1978.

Turkowski, K., "Antialiasing Through the Use of Coordinate Transformations", Computer Graphics, v.16, July, 1982.

Whitted, T., "An Improved Illumination Model for Shaded Display", Communications of the ACM, v.23, June, 1980.

Williams, L.J., "Pyramidal Parametrics", Siggraph notes, Advanced Image Synthesis Seminar, August, 1981; see also Computer Graphics, v.17, July, 1983.

Williams, L.J., "Brute Force in Image Space", Ph.D. Thesis (unpublished), University of Utah, Computer Science Department.