Artigo

FASTER PIXEL BIT MASK GENERATION EXPLOITING POINT-TO-POINT COHERENCE OF SCANLINES

José Mário De Martino
(UNICAMP / Haus der Graphischen Datenverarbeitung – Darmstadt, Alemanha)

SIBGRAPI'91
IV Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens

Página em branco na versão original impressa.

Faster Pixel Bit Mask Generation Exploiting Point-to-Point Coherence of Scanlines

José Mario De Martino

State University of Campinas, Brazil
Haus der Graphischen Datenverarbeitung
Wilhelminenstraße 7 - 6100 Darmstadt - FRG
mario@zgdvda.uucp
mario%zgdvda.uucp@unido.bitnet

ABSTRACT

Everyone who is willing to generated high quality images is faced with aliasing problems. Prefiltering of the image is a recognized method to soften the symptoms of aliasing. One simple filter is the integration of the visible intensities over the area of a pixel. An exact pixel integrater has to clip all polygons found in a pixel against each other and the pixel window. Pixel bit masks is a convenient approximation to represent the visible area of polygons at a subpixel level. These masks reduces the full polygon clipping to simple boolean exclusive-or operations. The efficient generation of such pixel bit masks plays an important role in the time performance of the synthesis process. The idea of pixel bit masks is not new, but exploiting the point-to-point coherence of scanlines, it was possible improve the performance over the original algorithm .

1. Introduction

Jagged edges, flickering of edges and flashing off and on of small objects in animated sequences are some common symptoms of not properly anti-aliased images. As pointed out in [Crow 77], these undesirable symptoms come from the undersampling of the image at a rate lower than the Nyquist frequency (the highest frequency of the a bandlimited signal).

Fundamentally, there are two methods to soften the aliasing problem. The first is to increase the sampling rate, generating the image at a higher resolution. The display of such an image can be accomplished by increasing the resolution of the raster device (shifting the problem, without solving it) or by low-pass filtering the image back to the lower resolution, using some kind of averaging. The second method of anti-aliasing is to limit the bandwidth of the signal by prefiltering (low-pass). The results presented in [Crow 81] suggests that the second method has advantages, producing acceptable images in acceptable time. The prefiltering process, however, has to be efficient and simple enough to maintain these advantages.

One simple filter, that satisfy the above mentioned conditions, is the integration of the visible intensities over the area of each pixel. Adaptive applying the filter only in the regions where abrupt intensity variations (high frequencies) are surely to be found, e.g. at the polygon edges, gives rise to an extra gain in performance. The Bresenham's algorithm with anti-aliasing [Pitt 80] could be mentioned as an example of such a simple filtering. However, this solution has two major drawbacks. First, it fails to cope with small objects. Second, the coverage alone does not supply enough information for composing the final intensity of the pixel [Port 84]. Without the geometric information about eventual overlap of polygons in a pixel, we are always incurring in rough approximations.

Catmull devised a strategy to do an exact area sampling [Catm 78]. Although elegant, the solution proposed, based on the Weiler-Atherton polygon clipper [Weil 77], is at least questionable for complex scenes. The problem arises not only because the exhaustive polygon-against-polygon clipping but also due to the calculation of the resulting clipped areas. A practical alternative to Catmull's algorithm is the utilization of bit masks to represent the polygons at a subpixel level, as devised independently by [Fium 83] and [Carp 84]. This solution has the advantage that the clipping of polygons is reduced to a simple boolean exclusive-or operation and the calculation of the resulting area to a summation. A potential disadvantage of the method is its accuracy. However, this can be controlled by the size of the pixel bit mask. Because the construction of the pixel bit masks plays a importance role in the overall performance of the rendering process, the question is how to build up efficiently these masks.

2. Constructing Pixel Bit Masks

For sake of completeness, it is reproduced here the basics of building up the masks as proposed in [Carp 84]. The construction of a pixel mask begins with a polygon that has been clipped to a pixel boundary. Each polygon edge defines a trapezoid bounded by the edge itself, the right side of the pixel, and by the extension, parallel to x, of the ends of the edge toward the right side of the pixel. The edge mask is constructed by or'ing together row masks taken from a table indexed by the quantized location of the intercepts of edge with subpixel scanlines. The pixel mask of a polygon is constructed xor'ing its edge masks. It was found through experimentation that 4x8 bit masks provides an acceptable image quality. Unfortunately, Carpenter did not explain how to generate the indices of the row mask table. In our implementation of Carpenter's algorithm, we used a modified Bresenham's algorithm, which generate the index of a subpixel when the area of the subpixel right to the edge is equal of bigger than half of the total subpixel area. The subpixel position of the end points of a edge are calculated by rounding.

We claim here that, in scanline oriented hidden-surface elimination algorithms, the full clipping of the polygons against the pixel window is avoidable. Exploiting the point-to-point coherence of a scanline, it is possible to build up the pixel bit masks only clipping against the pixel window those edges of a polygon that are effectively present in the pixel.

In our approach an edge mask is fundamentally constructed in the same way as proposed by Carpenter. However, the final pixel bit mask of a polygon is constructed by x-or'ing together the masks of the edges that are present in the pixel and a initial mask. This initial mask is empty at the begin of the scanline and updated for each new processed pixel. The update is done by extending the right most column of the last pixel bit mask to the right side of the present pixel. In order to keep continuity of the masks in a scanline, we apply a floor function to find the subpixel x coordinates of the end points of a edge. The y coordinate is calculate, as above, by rounding. Moreover, the generation of the indices for the row mask table is accomplished using a DDA algorithm.

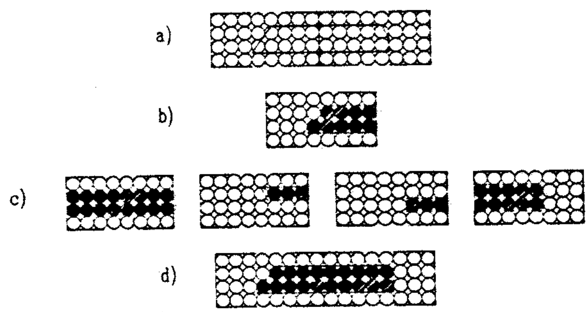

fig 1 - Bit Mask Generation

Figure 1 shows an example. that helps to clarify the idea of the algorithm. Figure 1a shows a small object that covers partially two pixels of a scanline, that otherwise is empty. Beginning from the left, the mask of the edge that is present in the first pixel is constructed (fig. 1b). In this case the initial mask is empty. Figure 1c shows all masks associated with the second pixel. The left most is the initial mask, constructed by extending the last column of the bit mask of the first pixel. The two other subsequent masks are the edge masks of the left side of the small object. The exclusive-or of these three masks results in the final bit mask of the pixel; also shown in figure 1c. Both resultant pixel bit masks are shown in Figure 1d.

3. Results

For comparison purposes, we implemented both strategies, as explained above, to construct the bit pixel masks. The exploitation of point-to-point coherence in scanlines showed a reduction of the processing time of about 50 percent. Of course, the gain in performance floats according to the degree of coherence that a given scene has and with the complexity of the primitives (polygons). We measured the performance of the algorithm with specially generated scenes, which tried to emphasize the drawbacks of both algorithms, and with sequences of films that has been produced at the "Haus der Graphischen Datenverarbeitung". As best result it was possible to measure a gain of about 80 percent; as worst result 30. The average was about 50 percent.

Figure 2 shows a test scene. The scene is composed of xx cylinders, each with xxx polygons. The cylinder on the left shows the polygonal structure of the object. The right one and those composing the wheel were rendered applying the geometric smoothing proposed by Phong [Phon 75]. The illumination model used contemplates only the ambient and diffuse components.

Figures 3 and 4 show a detail of the test scene (xx times zoomed) generated with Carpenter method and with the point-to-point coherence method, respectively. Taking the generation time of the Carpenter method as unit the point-to-point coherence method showed a performance of 0.41.

4. Conclusion

In this paper we presented a strategy to construct pixel bit masks that exploits point-to-point coherence of a scanline. The results show an improvement in performance of about 50 percent over an implementation that does not exploit this kind of coherence. The improved is mainly due to the saving of the full polygon clipping against the pixel window. In the point-to-point coherence method is only necessary to clip the edges a polygon that effectively pass through the pixel.

In a higher level, we claim that pixel bit masks offer a better strategy then the simple alfa-channel strategy (coverage) [Port 84]. We base this observation on the fact that pixel bit masks do not only provide the percentage of the pixel that is covered (coverage information) but also the geometric information of how/where the pixel is covered. This information can be used to compose more accurately the final colour of pixels that are partially covered by more than one polygon. Another interesting aspect of pixel bit masks is the easy that one can increase the accuracy of the method. This is achieved only increasing the size of the mask. Our experiments showed that a 4x8 is not always satisfactory. In our implementation, it is possible to choose between 4x8, 8x8, 16x8 or 16x16 masks.

References

[Pitt 80] Pitteway, M., Watkinson, D.; Bresenham's Algorithm with Grey Scale; *Comm. ACM*, Vol. 22, No. 11; November 1980, pp. 625-626.

[Carp 84] Carpenter, L.; The A-Buffer, an Antialised Hidden Surface Method; *Computer Graphics* (ACM SIGGRAPH), Vol. 18, No. 3; July 1984, pp. 103-108.

[Catm 78] Catmull, E.; A Hidden-Surface Algorithm with Anti-Aliasing; *Computer Graphics* (ACM SIGGRAPH), Vol. 13, No. 3; August 1978, pp. 6-11.

[Crow 77] Crow, F.C.; The Aliasing Problem in Computer-Generated Shaded Images; *Comm. ACM*, Vol. 20, No. 11; November 1977, pp. 799-805.

[Crow 81] Crow, F.C.; A Comparison of Antialiasing Techniques; *IEEE Computer Graphics and Application*, Vol. 1, No. 1; January 1981, pp. 40-49.

[Fium 83] Fiume, E., Fournier, A.; A Parallel Scan Conversion Algorithm with Anti-Aliasing for a General-Purpose Ultracomputer; *Computer Graphics* (ACM SIGGRAPH), Vol. 17, No. 3; July 1983, pp. 141-149.

[Phon 75] Phong, Bui Toung; Illumination for Computer Generated Pictures; *Comm. ACM*, Vol. 18, No. 8; August 1975, pp. 311-317.

[Port 84] Porter, T., Duff, T.; Composing Digital Images; *Computer Graphics* (ACM SIGGRAPH), Vol. 18, No. 3; July 1984, pp. 253-259.

[Weil 77] Weiler, K., Atherton, P.; Hidden Surface Removal Using Poligon Area Clipping; *Computer Graphics* (ACM SIGGRAPH), Vol. 11, No. 3; July 1977, pp. 214-222.

Acknowledgements

I gratefully acknowledge the support of Prof. Dr. J.L. Encarnaçao and the staff of the "Haus der Graphischen Datenvearbeitung", FRG. I also acknowledge the financial support of the "CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico", Brazil.

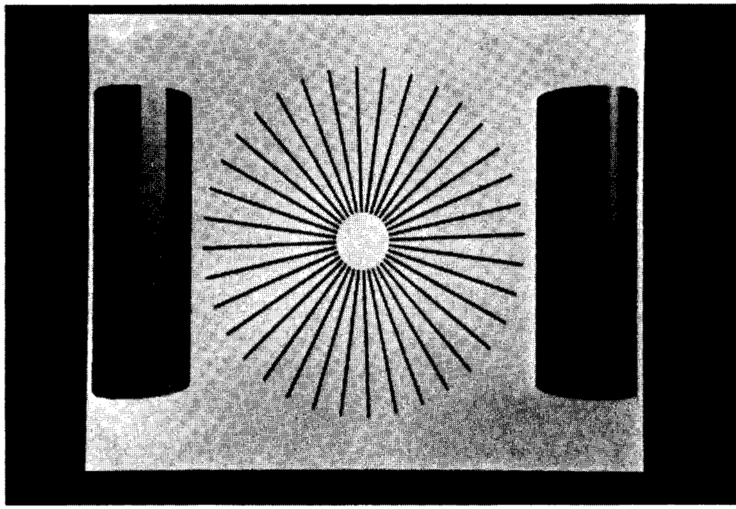

fig.2 - Test Scene (proposed method)

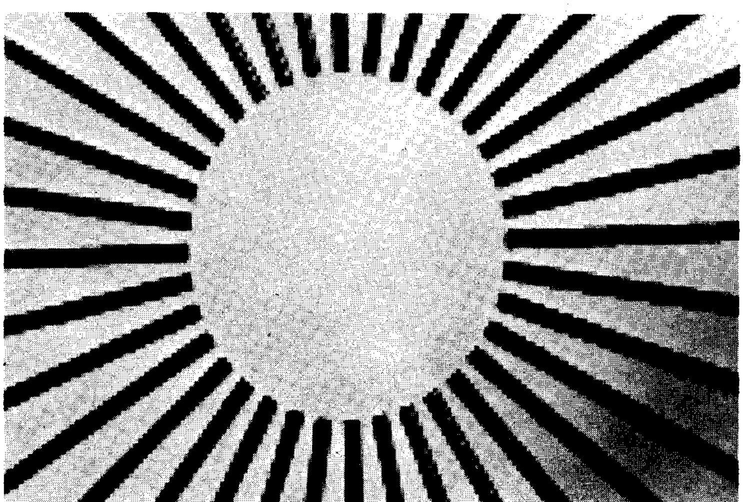

fig.3 - Carpenter's Method

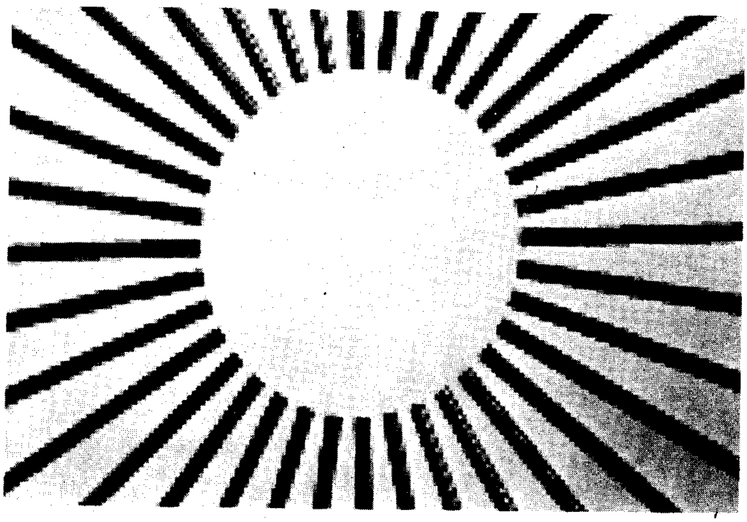

fig.4 - Proposed Method