# Line Art Rendering Using a Modified A-Buffer

**Mayur Patel**
Industrial Light + Magic
patelm@acm.org

There are many different technologies for producing line art. Analytical methods, or object-space methods, can give very complete line information but are not as easy to implement as scan-conversion methods.[1] Hardware techniques are very fast, but may render with aliasing, and they may not render crease lines.[2] Using edge detection on geometry maps has become very popular, but it also requires supersampling and filtering to avoid aliasing.[3]

The A-buffer is a robust anti-aliasing hidden-surface evaluator.[4] The technology uses a bitmask as an occlusion map for each sample scanned into each pixel. A typical A-buffer coverage mask covers the area of one pixel. Let us define an extended coverage mask as one that covers an area larger than one pixel. For our purposes, we need a mask that extends by one sub-pixel row or column into the area of each neighboring pixel. The extended coverage mask resembles a convolution region of interest and could be used for filtering. See Figure 1.

With such masks, we can conduct line detection between samples in the same pixel. Without the extended coverage information, iteration over neighboring pixels would be necessary to detect lines occurring at the pixel borders. Figure 2 demonstrates line detection between two 16-bit extended coverage masks. Figure 3 gives C code for executing these functions using bitwise operations.

Modifying an A-buffer to produce this information is not particularly difficult. We benefit from associating line samples with coverage masks because some subsystems in the rendering pipeline can process line samples as if they were surface samples. However, a generalized line-shading architecture varies greatly from a surface-shading architecture. The look of a line can rarely be decided by point-sampling. Aesthetic choices are often made based on local area information in world space or image space.[5,6,7] Artists may also want to modify line widths or placement to produce a more clear illustration. Some shading features may even involve repositioning line samples after scan conversion.

Given the development effort, the A-buffer can cope with many of the demands of a line-shading architecture. It can render anti-aliased surfaces and lines in one pass with transparency. Polygons need not be convex or preprocessed. Scan conversion information can easily be provided to subsystems internally for accurate line detection and shading. These features make the technique applicable to a wide range of line-art rendering projects.

### *References*

1. Elber, G. & Cohen, E. (1990). Hidden curve removal for free form surfaces. *ACM SIGGRAPH 90 Conference Proceedings*.
2. Raskar, R. & Cohen, M. (1999, April). Image precision silhouette edges. ACM Symposium on Interactive 3D Graphics.
3. Saito, T. & Tokiichiro, T. (1990). Comprehensible rendering of 3D shapes. *ACM SIGGRAPH 90 Conference Proceedings*.
4. Carpenter, L. (1984). The A-buffer, an antialiased hidden surface method. *ACM SIGGRAPH 84 Conference Proceedings*.
5. Dooley, D. & Cohen, M. (1990, March). Automatic illustration of 3D geometric models: Lines. ACM Symposium on Interactive 3D Graphics.
6. Kamada, T. & Kawai, S. (1987). An enhanced treatment of hidden lines. *ACM Transactions on Graphics, 6 (4)*.
7. Gooch et. al. (1999, April). Interactive technical illustration. ACM Symposium on Interactive 3D Graphics.