Discrete Analysis for Antialiased Lines

V. Boyer and J.J. Bourdin

Groupe de Recherche en Infographie et Synthèse d'images
Laboratoire d'Intelligence Artificielle
Université Paris 8
2, rue de la Liberté
93256 Saint-Denis cedex 2
FRANCE
boyer, jj@ai.univ-paris8.fr
Tel : 33.1.49.40.64.04
Fax : 33.1.49.40.64.10

Abstract

This paper presents a new fast and smooth antialiasing technique. Drawing straight line is the graphic tool's main primitive. Previous antialiasing techniques improve smoothness but slow down computation.

1. Introduction

As mentioned in Foley[1], one of the drawback of raster systems arises from the discrete nature of the pixel representation. For example, the drawing of straight line presents discontinuities because of pixels approximation.
This effect is a manifestation of a sampling error called aliasing in signal-processing theory[2]. Both theory and practice in modern computer graphics are concerned with antialiasing techniques. These techniques specify gradations in intensity of neighboring pixels, rather than setting pixels to maximum or zero intensity only.
There are two main approaches to solve the problem: to apply a filter on an existing image (postprocessing)[3,4] and to use a modified (antialiased) scan-conversion algorithm (preprocessing)[1]. Preprocessing techniques produce directly antialiased curves. The main idea is to create a "thick" curve. Each pixel intersected by the thick curve is displayed with a non zero intensity. The surface of the intersection between a given pixel and the thick curve will be denoted "covered area".
Unweighted and weighted area sampling are the main preprocessing techniques. Unweighted area sampling set intensity proportional to the amount of covered area. Weighted area sampling adds a function of distance between the center of the pixel and the curve.
Gupta-Sproull scan conversion algorithm for straight lines[5] precomputes a table of intensities.
In this paper, we describe the continuous, discrete and thick lines. Based on these descriptions, we present a new technique using the Gupta-Sproull and unweighted area sampling methods. The deduced algorithm is twice as quick as the previous ones.

2. Line description

In this part, we describe all the necessary elements to understand the new algorithm. Definitions and notations of the continuous, discrete and thick lines are presented. Finally we present new properties in order to compute quickly the continuous thick line.
In the following, the notation $_r$ will denote a real value or an object in the continuous plane.
A point of a plane is defined by two coordinates. Let $P(x_p, y_p)$ and $Q(x_q, y_q)$ be two points of the discrete plane $\mathbb{N} \times \mathbb{N}$ representing the raster device.
In a continuous plane $\mathbb{R} \times \mathbb{R}$ the continuous line between P and Q will refer to the line segment from P to Q.
In a discrete plane, the discrete line from P to Q corresponds to a "linear" path from P to Q. The discrete line is a set of discrete plane's points closest to the continuous line.
Let u and v be respectively the difference of abscissae and ordinates between P and Q.
As noted by Bresenham[6] the discrete line from P to Q is an exact transposition of the line $(0,0)(u,v)$. The slope of the line is given by the pair (u,v). For a given slope (u,v),

$L(u,v)$ will denote the discrete line and $L_r(u,v)$ the continuous line.
Moreover each continuous and discrete line can be computed in the first octant and, by symmetries, drawn[6]. Therefore $u > v \geq 0$. Each continuous line $L_r(u,v)$ and discrete line $L(u,v)$ in the first octant satisfy:

$$\forall\, x_r \in [0,\, u],\ \exists!\, y_r \in [0,\, v] \,/\, (x_r,\, y_r) \in L_r(u,\, v)$$

$$\forall\, x \in [0,\, u],\ \exists!\, y \in [0,\, v] \,/\, (x,\, y) \in L(u,\, v)$$

In the first octant :

$$y_r = \frac{v x_r}{u}$$

and

$$y = \left[\frac{vx}{u}\right]$$

$[z]$ means the best integer approximation of the real value z.
A **thick line** $T_r(L_r,\, t)$ is a set of points $(x_r,\, y_r)$ of the continuous plane that respect:

$$d\left((x_r,\, y_r),\, L_r\right) < \frac{t}{2}$$

where $d\left((x_r,\, y_r),\, L_r\right)$ is the distance between the point $(x_r,\, y_r)$ and the continuous line L_r. t is therefore the thickness of the thick line.
Let γ be the maximum ordinate's difference between two points of $T_r(L_r,\, t)$. If θ is the angle between the line (u, v) and the horizontal line (see figure 1); γ is defined by:

$$\gamma = \frac{t}{\cos\theta} = \frac{t\sqrt{u^2+v^2}}{u}$$

In a discrete plane $\mathbb{N}\times\mathbb{N}$ a thick discrete line $T(L_r,t)$ is the set of all points $(x,y) \in \mathbb{N}\times\mathbb{N}$ that are at least partially covered by $T_r(L_r,\, t)$. For each point $(x,\, y) \in \mathbb{N}^2$, let $A(x,\, y)$ be the surface of a point $(x,\, y)$ covered by $T_r(L_r,\, t)$. If the $A(x,y) \neq 0$ then $(x,y) \in T(L_r,t)$. Remark:

$$\forall x \in [0,u],\ \sum_{y\in\mathbb{N}} A(x,y) = \gamma$$

The sum of the *covered surfaces* of points belonging $T(L_r,t)$ with the same ordinate is equal to γ.

In the following, we concentrate on the case $t = 1$, so in the first octant $\gamma \in [1,\, \sqrt{2}]$.
For x in $[0,\, u]$, there are one, two or three points belonging to $T(L_r,\, t)$. The next paragraph presents these different cases.

Case 1 *(one point per column)*: this case appears only when $v = 0$. Therefore

$$\forall\, (x,\, y) \in T,\, A(x,y) = 1$$

Case 2 *(two points per column)*:

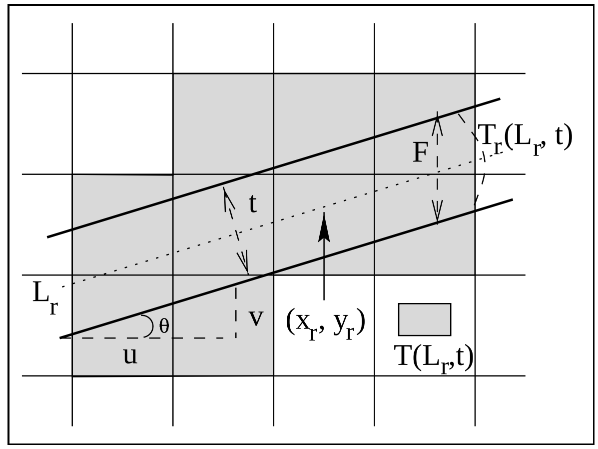

Figure 1: *Visualization of a thick line*

Let $(x,y_\perp)$ and $(x,y_\top) \in T(L_r,\, t)$. As one of them is directly above the other one, let write $y_\top = y_\perp + 1$:

$$A(x,y_\perp) = \frac{\gamma}{2} + \left(\left(y_\perp + \frac{1}{2}\right) - \frac{vx}{u}\right)$$

$$A(x,y_\top) = \frac{\gamma}{2} - \left(\left(y_\top - \frac{1}{2}\right) - \frac{vx}{u}\right)$$

Case 3 *(three points per column)*:
Let $(x,\, y_\perp)$, $(x,\, y_{\mathtt{I}})$ and $(x,\, y_\top)$ be three points of $T(L_r,\, t)$. As mentioned above we can write $y_{\mathtt{I}} = y_\perp + 1$, $y_\top = y_{\mathtt{I}} + 1$.

$$A(x,y_\perp) = \frac{1}{2}\left(y_\perp + \frac{1}{2} - \frac{v\left(x - \frac{1}{2}\right)}{u} - \frac{\gamma}{2}\right) \times \left(\frac{u\left(y_\perp + \frac{1}{2} + \frac{\gamma}{2}\right)}{v} - \left(x - \frac{1}{2}\right)\right)$$

$$A(x,y_\top) = \frac{1}{2}\left(\frac{v\left(x + \frac{1}{2}\right)}{u} + \frac{\gamma}{2} - \left(y_\top - \frac{1}{2}\right)\right) \times \left(\frac{-u\left(y_\top - \frac{1}{2} - \frac{\gamma}{2}\right)}{v} + \left(x + \frac{1}{2}\right)\right)$$

$$A(x,y_{\mathtt{I}}) = \gamma - A(x,y_{\mathtt{I}}) - A(x,y_\top)$$

As the line is composed by an **inner symmetry**[7,8], our purpose is to apply this symmetry to covered areas.

Case 1: the symmetry is obvious
Case 2: Let $(x,y_\top)$ and $(x,y_\perp)$ be points of $T(L_r,t)$ and

$y_\top = y_\bot + 1$ then:

$$A(x,\, y_\bot) = A(u - x,\, v - y_\top)$$

$$A(x,\, y_\top) = A(u - x,\, v - y_\bot)$$

Case 3: Let $(x, y_\top)$, $(x, y_{⌶})$ and $(x, y_\bot)$ be points of $T(L_r, t)$ and $y_{⌶} = y_\bot + 1$, $y_\top = y_{⌶} + 1$ then:

$$A(x,\, y_\top) = A(u - x,\, v - y_\bot)$$

$$A(x,\, y_{⌶}) = A(u - x,\, v - y_{⌶})$$

$$A(x,\, y_\bot) = A(u - x,\, v - y_\top)$$

It is easy to prove that for a value $u - x$ the case 2 (respectively 3) occurs if and only if for x the case 2 (respectively 3) occurs. Therefore the inner symmetry of the thick discrete line is proved.

3. New algorithm

The coordinate of the lowest pixel is computed using Bresenham's algorithm. When a diagonal move is computed, three pixels will be drawn. When an axial move occurs, two pixels will be drawn. Before drawing the adequate pixels, the amount of their area covered by the thick continuous line is computed and used as an entry in the color LUT (this is an adaptation of the Gupta-Sproull's algorithm). Using the symmetry mentioned above, the new algorithm computes only the first half of the line. The other half is automatically drawn in the same loop. Previous algorithms iterated u times the loop, and ours iterates it $u/2$ times. The Color Look-Up-Table (Color LUT) is used to produce a perfect visual effect. Two examples are given in figures 2 and 3. Each figure presents two lines of same slope: the upper one is not antialiased while the lower one is drawn using our algorithm.

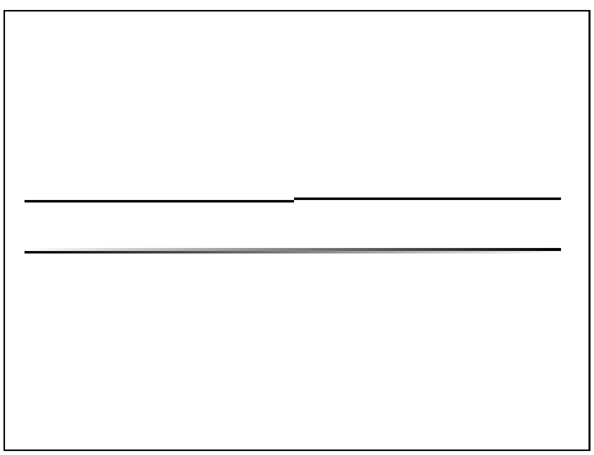

Figure 2: *Results for a line of slope (200, 1)*

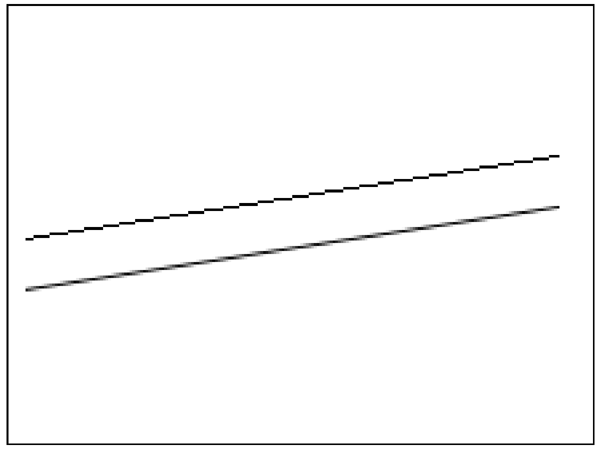

Figure 3: *Results for a line of slope (200, 31)*

4. Conclusion

Based on a Gupta-Sproull's and unweighted area sampling, we have realized an antialiased algorithm for straight lines. Due to the inner symmetry, the new algorithm is proved to be twice as fast as previous ones. In the near future, the new symmetry property will be applied to an efficient algorithm like Boyer et al.[8] and a discrete analysis will be implemented to compute the areas so we can expect a significant speed improvement.

References

1. J.D. Foley, A. Van Dam, S. Feiner, and J. Hughes. *Computer Graphics, Principles and Practice*. Addison Wesley, second edition, 1990.
2. R. Gonzales and P.Wintz. *Digital Image Processing*. Addison Wesley, second edition, 1987.
3. J.F. Blinn. What We Need Around Here is More Aliasing. *IEEE CG&A*, 9(1), January 1989.
4. D.P. Mitchell. Reconstruction Filter in Computer Graphics. In *SIGGRAPH*, pages 221–228, 1988.
5. S. Gupta and R. Sproull. Filtering Edges for Gray-Scale Displays. *Computer Graphics*, 15(3), August 1981.
6. J.E. Bresenham. Algorithm for computer control of a digital plotter. *IBM System Journal*, 4(1):25–30, 1965.
7. J.G. Rokne, B. Wyvill, and X. Wu. Fast line scan-conversion. *ACM TOG*, 9(4):376–388, October 1990.
8. V. Boyer and J.-J. Bourdin. Fast Lines : a Span by Span Method. In *Proceedings of Eurographics'99*, volume 18(3) of *Computer Graphics forum*. Blackwell Publishers, September 1999.