Xiaolin Wu's line algorithm

From Wikipedia, the free encyclopedia

Antialiased line drawn with Xiaolin Wu's algorithm

Xiaolin Wu's line algorithm is an algorithm for line antialiasing, which was presented in the article *An Efficient Antialiasing Technique* in the July 1991 issue of *Computer Graphics*, as well as in the article *Fast Antialiasing* in the June 1992 issue of *Dr. Dobb's Journal*.

Bresenham's algorithm draws lines extremely quickly, but it does not perform anti-aliasing. In addition, it cannot handle any cases where the line endpoints do not lie exactly on integer points of the pixel grid. A naive approach to anti-aliasing the line would take an extremely long time. Wu's algorithm is comparatively fast, but is still slower than Bresenham's algorithm. The algorithm consists of drawing pairs of pixels straddling the line, each coloured according to its distance from the line. Pixels at the line ends are handled separately. Lines less than one pixel long are handled as a special case.

An extension to the algorithm for circle drawing was presented by Xiaolin Wu in the book *Graphics Gems II*. Just like the line drawing algorithm is a replacement for Bresenham's line drawing algorithm, the circle drawing algorithm is a replacement for Bresenham's circle drawing algorithm.

```
function plot(x, y, c) is
    plot the pixel at (x, y) with brightness c (where 0 ≤ c ≤ 1)

// integer part of x
function ipart(x) is
    return int(x)

function round(x) is
    return ipart(x + 0.5)

// fractional part of x
function fpart(x) is
    if x < 0
        return 1 - (x - floor(x))
    return x - floor(x)

function rfpart(x) is
    return 1 - fpart(x)

function drawLine(x0,y0,x1,y1) is
    boolean steep := abs(y1 - y0) > abs(x1 - x0)

    if steep then
        swap(x0, y0)
        swap(x1, y1)
    end if
    if x0 > x1 then
        swap(x0, x1)
        swap(y0, y1)
    end if

    dx := x1 - x0
    dy := y1 - y0
    gradient := dy / dx

    // handle first endpoint
    xend := round(x0)
```

```
    yend := y0 + gradient * (xend - x0)
    xgap := rfpart(x0 + 0.5)
    xpxl1 := xend // this will be used in the main loop
    ypxl1 := ipart(yend)
    if steep then
        plot(ypxl1,   xpxl1, rfpart(yend) * xgap)
        plot(ypxl1+1, xpxl1,  fpart(yend) * xgap)
    else
        plot(xpxl1, ypxl1  , rfpart(yend) * xgap)
        plot(xpxl1, ypxl1+1,  fpart(yend) * xgap)
    end if
    intery := yend + gradient // first y-intersection for the main loop

    // handle second endpoint
    xend := round(x1)
    yend := y1 + gradient * (xend - x1)
    xgap := fpart(x1 + 0.5)
    xpxl2 := xend //this will be used in the main loop
    ypxl2 := ipart(yend)
    if steep then
        plot(ypxl2  , xpxl2, rfpart(yend) * xgap)
        plot(ypxl2+1, xpxl2,  fpart(yend) * xgap)
    else
        plot(xpxl2, ypxl2,  rfpart(yend) * xgap)
        plot(xpxl2, ypxl2+1, fpart(yend) * xgap)
    end if

    // main loop
    for x from xpxl1 + 1 to xpxl2 - 1 do
        begin
            if steep then
                plot(ipart(intery)  , x, rfpart(intery))
                plot(ipart(intery)+1, x,  fpart(intery))
            else
                plot(x, ipart (intery),  rfpart(intery))
                plot(x, ipart (intery)+1, fpart(intery))
            end if
            intery := intery + gradient
        end
end function
```

References [edit]

- Abrash, Michael (June 1992). "Fast Antialiasing (Column)". *Dr. Dobb's Journal* **17** (6): 139(7).
- Wu, Xiaolin (July 1991). "An efficient antialiasing technique". *Computer Graphics* **25** (4): 143–152. doi:10.1145/127719.122734. ISBN 0-89791-436-8.
- Wu, Xiaolin (1991). "Fast Anti-Aliased Circle Generation". In James Arvo (Ed.). *Graphics Gems II*. San Francisco: Morgan Kaufmann. pp. 446–450. ISBN 0-12-064480-0.

External links [edit]

- Xiaolin Wu's homepage
- Matlab algorithm

Categories: Computer graphics algorithms