2D Graphics Rasterisation

Arden Rasmussen

November 21, 2017

Abstract

2D rasterisation is the process of converting rendering primitives such as polygons and lines, into a matrix of pixels or dots for output on a video display. This paper will look at the methods for rasterising several rendering primitives, such as lines, circles, and arbitrary polygons. This paper will also examine the methods for filling in the raster images generated from the rendering primitives.

1 Introduction

2D rasterisation is used to convert graphics primitives such as polygons and lines into a raster image. The raster image is a matrix of pixel, dot, or point data. These raster images can be used to output to a video display, or printer, or bitmap file formats. The rasterisation methods are necessary for determining what individual pixels of a display format must be set to. Because of this, rasterisation algorithms are found in all sorts of uses in computer science. This paper will examine some of the algorithms that are used for rasterisation of:

1. Lines
2. Circles
3. Polygons

Along with a selection of algorithms that are used for filling polygons.

2 Lines

These algorithms are used to approximate a line segment on discrete graphical media. On the discrete graphical media, a line must be drawn as an approximation of the actual mathematical line segment in nontrivial cases.

2.1 Naive Line-Drawing

Algorithm 1 Naive Line-Drawing

$d_x = x_2 - x_1$
$d_y = y_2 - y_1$
for $x \leftarrow x_1$ to x_2 **do**
    $y = y_1 + \frac{d_y(x - x_1)}{d_x}$
    PLOT(x,y)
end for

This is only a valid algorithm if the points are already ordered such that $x_2 > x_1$. And it only works when $d_x \geq d_y$. To demonstrate the issues that are caused by this algorithm, here are several images depicting results of the algorithm.

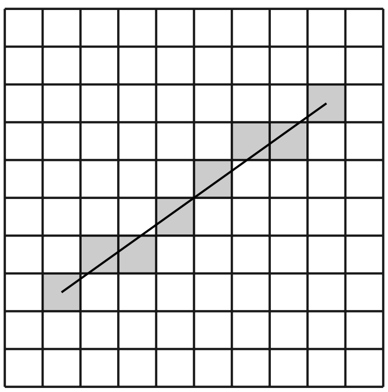

Figure 1: $d_x \geq d_y$

It can be seen that this algorithm can work effectivly for most cases where the slope of the line segment is less than or equal to 1.

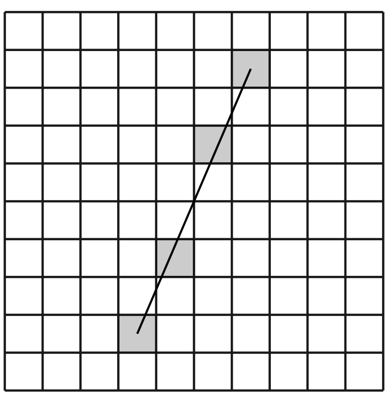

Figure 2: $d_x < d_y$

It can quickly be seen that when the slope of a line exceeds 1, this algorithm quickly fails to draw compleate lines. And leaves large gaps between the pixels that it does draw.

Becaus of these reasons, this algorithm is rarly every used, and the other algorithms demonstrated are prefered.

2.2 Digital Differential Analyzer (DDA)