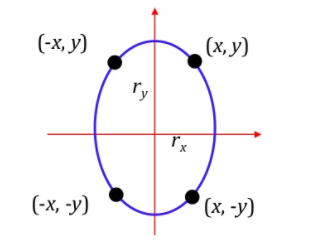
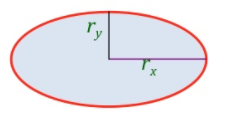
**Question 2. Algorithm description / mathematical derivations**

First of All what is a Ellipse – A modified circle whose radius varies from a max value in one direction, often is the major axis, to a min value in the perpendicular direction. It can be separated into four equal parts – symmetry between quadrants, each quadrants are separate into 2 regions.





Cartesian coordinates: = 1

Polar coordinates:

Therefore the condition parameter is:

if (x,y) is inside the ellipse

if on the ellipse

if outside of the ellipse

Slope = = -1 =>

At slope = -1 separate the each part in to 2 region

= Region 1

Then we use midpoint ellipse Algorithm. Shift x to the right by 1

Shift y downwards by half.

if p < 0. The midpoint is inside the ellipse 🡪 is closer

else the mid point is out side the elliose 🡪 is closer.

Therefore we have to increment:

if

if

for region 2 we use midpoint ellipse Algorithm. Shift x to the right by half

Shift y downwards by 1.

if p < 0. The midpoint is inside the ellipse 🡪 is closer

else the mid point is out side the ellipse 🡪 is closer.

Therefore we have increment:

if

if

**Question 3. Description of the selected algorithm**

Algorithm 1. Floyd-Steinberg algorithm

The first – and arguably most famous – 2D error diffusion formula was published by Robert Floyd and Louis Steinberg in 1976. It diffuses errors in the following pattern:

X 7

3 5 1 1/16

In the notation above, “X” refers to the current pixel. The fraction at the bottom represents the divisor for the error. Said another way, the Floyd-Steinberg formula could be written as:

X/16 7/16

3 /16 5/16 1/16 1/16

Converting a pixel to 0 (black) or 255 (white), if we force the pixel to black, find the X first, We then propagate that error to the surrounding pixels by dividing X by 16, then multiplying it by the appropriate values for all four surrounding points.

Floyd-Steinberg dithering provides reasonably good quality, while only requiring a single forward array (a one-dimensional array the width of the image, which stores the error values pushed to the next row). Additionally, because its divisor is 16, bit-shifting can be used in place of division making it quite fast.

Algorithm 2. Burkes Dithering algorithm

If we change the pattern to :

X 8 4

2 4 8 4 2 1/32

It is an improvement to the Jarvis, Judice, Ninke dithering

For Jarvis,Judice, Ninke dithering algorithm bit-shifting can no longer be used. but only values of 1/48, 3/48, 5/48, and 7/48 are used, so these values can each be calculated but once, then propagated multiple times for a small speed gain. Unfortunately, you may be better off using a single error array the size of the image, rather than erasing the two single-row arrays over and over again.

For Daniel Burkes algorithms Not only did this remove the need for two forward arrays, but it also resulted in a divisor that was once again a multiple of 2. This change meant that all math involved in the error calculation could be accomplished by simple bit-shifting, with only a minor hit to quality.

Conclusions:

There are many new algorithms out there that are better than the old ones.

There are many old algorithms are still in use because they are the best.

From this research, Digital halftoning method is still improving. Burkes Dithering algorithm are bit better than the Floyd-Steinberg algorithm.