



INSTITUTO FEDERAL  
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
Ceará

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DO CEARÁ - CAMPUS FORTALEZA

Paulo Henrique Araujo Nobre

TRABALHO RASTERIZAÇÃO DE RETAS

FORTALEZA - CE

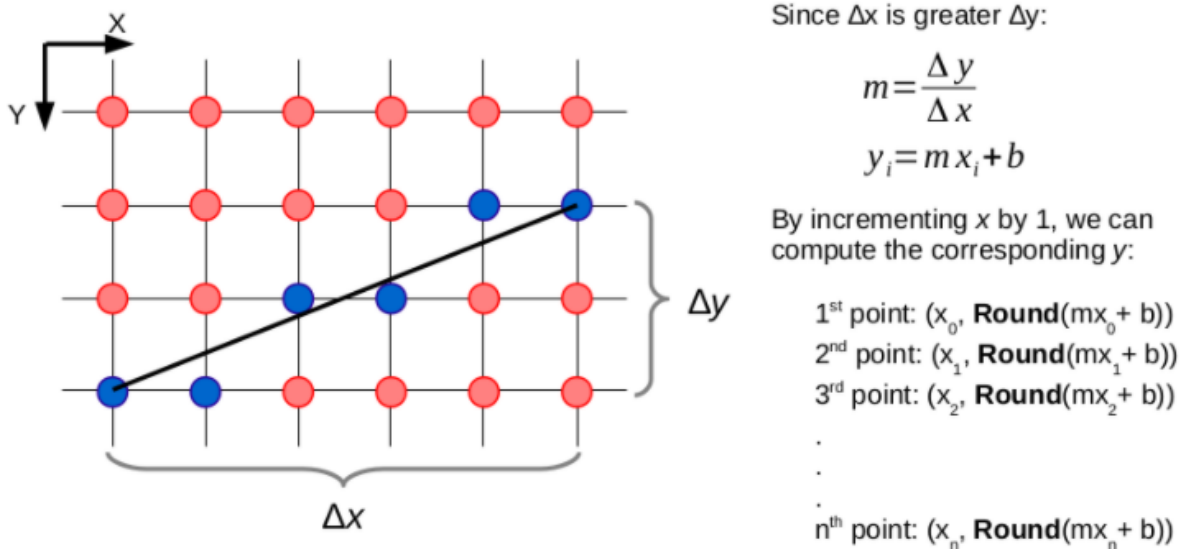
## Sumário:

1 Resumo.....	2
2 Implementação.....	2
2.1 Declaração de variáveis.....	2
2.2 Variáveis de configurações.....	3
2.3 Rasterização.....	3
2.4 Plotting.....	4
3 Resultados.....	4
4 Conclusão.....	5

## 1 Resumo:

A Rasterização é uma aproximação de variáveis contínuas para um espaço discreto. Numa primeira observação é possível o utilizar o coeficiente angular da reta e sua equação, e um arredondamento para encontrar o ponto y a ser pintado, incrementando o ponto x:

**Figura 1** - Exemplo de rasterização de retas



Fonte: [https://johannesca.github.io/cg\\_t1/](https://johannesca.github.io/cg_t1/)

## 2 Implementação:

A implementação da inicial é feita declarando 3(três) retas e em seguida chamando a função `raster_config()` 3(três) vezes e passando as retas para cada uma, no caso, cada chamada é feita para uma resolução distinta.

### 2.1 Declaração de variáveis:

Por padrão foi declarado 3(três) retas e 3(três) resoluções fixas representadas como reta  $x$ :  $[(x_i, x_f), (y_i, y_f)]$ :

- reta 1:  $[(0, 10), (0, 3)]$
- reta 2:  $[(1, 5), (1, 7)]$
- reta 3:  $[(2, 7), (3, 7)]$
- resoluções:  $[(40 \times 30), (800 \times 600), (1600 \times 1200)]$

## 2.2 Variáveis de configurações:

Dentro da função `raster_config()` é feita medições para trabalhar com mais facilidade com os valores.

Nela são feitas as proporções de escala de acordo com a resolução para que os valores das retas sejam proporcionais às resoluções.

No mesmo é criado a matrix usada como estrutura de dados para armazenar os valores da rasterização, com ela é feita uma lógica para criar-lá de comod que comporte de forma sucinta os valores e que não ocorra overflows.

Também é feito os cálculos necessários para o passa seguinte, no caso são os valores de Delta e Coeficiente angular da reta.

Tendo todos os valores e a estrutura de dados, basta chamar a função `raster()` passando a reta, o coeficiente angular, a estrutura de dados, o objeto representante da reta e por final o direcionamento, que no caso é o dado utilizado para saber qual variação é a mais predominante entre  $(\Delta y > \Delta x)$  ou  $(\Delta x > \Delta y)$ .

## 2.3 Rasterização:

Ao chamar a função `raster()` com todos os parâmetros é feito alguns cálculos antes de iniciar a rasterização.

É efetuado o cálculo do valor de  $b$  através da fórmula:

$$b = y - m * x$$

Verifica-se também através de uma condição para ver se a reta é crescente ou decrescente

Enfim é feita a rasterização, que no qual é realizada incrementando a variação predominantes entre  $\Delta y$  e  $\Delta x$  e recalculando o valor oposto utilizando a fórmula:

Para  $y$ :

$$y = m * x + b$$

Para  $x$ :

$$x = (y - b) / m$$

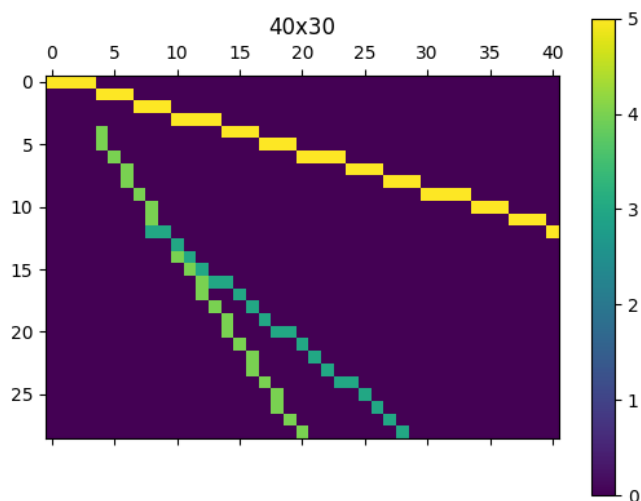
É por final atribuindo o valor representante da cor nos pontos encontrados de  $x$  e  $y$ , sendo elas 5 - amarelo, 4 - verde e 3 - azul.

## 2.4 Plotting:

No final da implementação é feita a chamada de uma figura para realizar a plotagem da matriz, utilizando a biblioteca matplotlib.

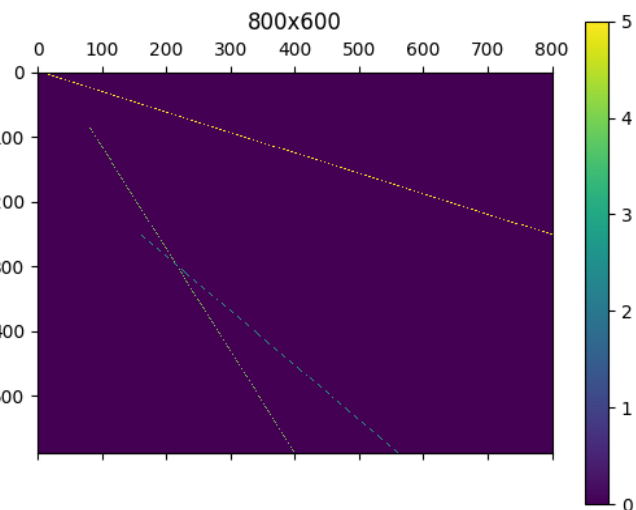
## 3 Resultados:

**Figura 2 - Resultado 1**



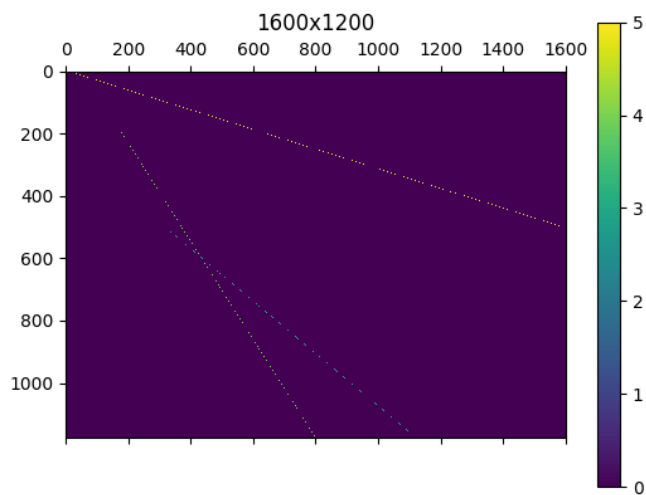
Fonte: Autor

**Figura 3 - Resultado 2**



Fonte: Autor

**Figura 4 - Resultado 3**



Fonte: Autor

## 4 Conclusão:

Tendo em vista os gráficos gerados pela aplicação, vemos a notória diferença da representação das retas de acordo com a distinta resolução de cada um. Vemos que quanto maior a resolução, maior a quantidade de pixels disponíveis na tela, em contrapartida temos gráficos mais pixelados com resoluções menores