MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIAS E TECNOLOGIAS
DEP. DE CIENCIA DA COMPUTACAO
DCC703 – COMPUTAÇÃO GRÁFICA (2024.2)
Prof. Dr. Luciano Ferreira Silva

EDUARDO HENRIQUE DE ALMEIDA IZIDORIO

RASTERIZAÇÃO DE LINHAS

BOA VISTA, RR

FEVEREIRO 2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIAS E TECNOLOGIAS
DEP. DE CIENCIA DA COMPUTACAO
DCC703 – COMPUTAÇÃO GRÁFICA (2024.2)
Prof. Dr. Luciano Ferreira Silva

EDUARDO HENRIQUE DE ALMEIDA IZIDORIO

RASTERIZAÇÃO DE LINHAS

Relatório de pesquisa apresentado ao Centro de Ciência da Computação do Curso de Ciência da Computação da Universidade Federal de Roraima, como requisito parcial para obtenção de notas da Disciplina COMPUTAÇÃO GRÁFICA – DCC 703, sob orientação do professor Dr. Luciano F. Silva

BOA VISTA, RR

FEVEREIRO 2025

SUMÁRIO

1. Justificativa

Este relatório tem como propósito registrar o desenvolvimento de uma aplicação que ilustra o funcionamento de três distintos algoritmos de rasterização de linhas: Analítico, DDA e Bresenham. Ao longo deste documento, serão apresentadas análises detalhadas, comparações e conclusões sobre esses algoritmos, destacando suas particularidades e aplicabilidades.

2. Introdução

A rasterização de linhas é o processo de transformar uma representação matemática de uma linha em uma sequência de pixels exibidos em um dispositivo de saída. Esse procedimento é fundamental na computação gráfica e desempenha um papel essencial na renderização de elementos gráficos em diversas aplicações, como jogos digitais, gráficos tridimensionais e imagens computacionais.

Principais Algoritmos de Rasterização de Linhas

- **Algoritmo Analítico:** Baseado na equação da reta, este método determina os pontos que devem ser renderizados com alta precisão. No entanto, devido ao seu alto custo computacional, é também o mais lento entre os três.
- **Algoritmo DDA:** Utiliza um método incremental para calcular os pontos da linha, tornando-se mais eficiente que o algoritmo analítico. Apesar de otimizar a continuidade do traçado, pode apresentar pequenas imprecisões devido ao arredondamento dos valores.
- **Algoritmo de Bresenham:** Representa uma versão otimizada do DDA, empregando um processo de aproximação para a determinação dos pixels. É o algoritmo mais rápido, mas pode ser menos preciso que o DDA em determinadas situações.

A rasterização de linhas desempenha um papel essencial na computação gráfica, sendo um processo indispensável para exibir linhas em uma tela. Os três algoritmos mencionados são amplamente utilizados, cada um com suas próprias vantagens e limitações.

A escolha do algoritmo ideal depende diretamente das necessidades específicas da aplicação, equilibrando fatores como **precisão, eficiência e desempenho computacional**.

3. Objetivos

Este trabalho teve como objetivos principais:

- Desenvolver um programa capaz de desenhar retas utilizando os algoritmos **Analítico, DDA e Bresenham**.
- Comparar os resultados obtidos com cada algoritmo, analisando aspectos como **precisão, eficiência e desempenho**.

A implementação desses algoritmos permitiu compreender seu funcionamento na prática, avaliando suas vantagens e limitações. Além disso, a comparação entre os métodos possibilitou uma melhor percepção de sua aplicabilidade em diferentes contextos computacionais.

4. Algoritmo Analítico

Definição:

O **algoritmo analítico de rasterização de linhas** utiliza um método baseado na equação da reta para determinar os pontos que compõem a linha. Ele avalia os extremos $\mathbf{p_1(x_1, y_1)}$ e $\mathbf{p_2(x_2, y_2)}$ por meio da equação:

$$y = m \cdot x + b$$

onde o coeficiente angular (**m**) é calculado como:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

e o termo independente (**b**) é obtido por:

$$b = y_1 - m \cdot x_1$$

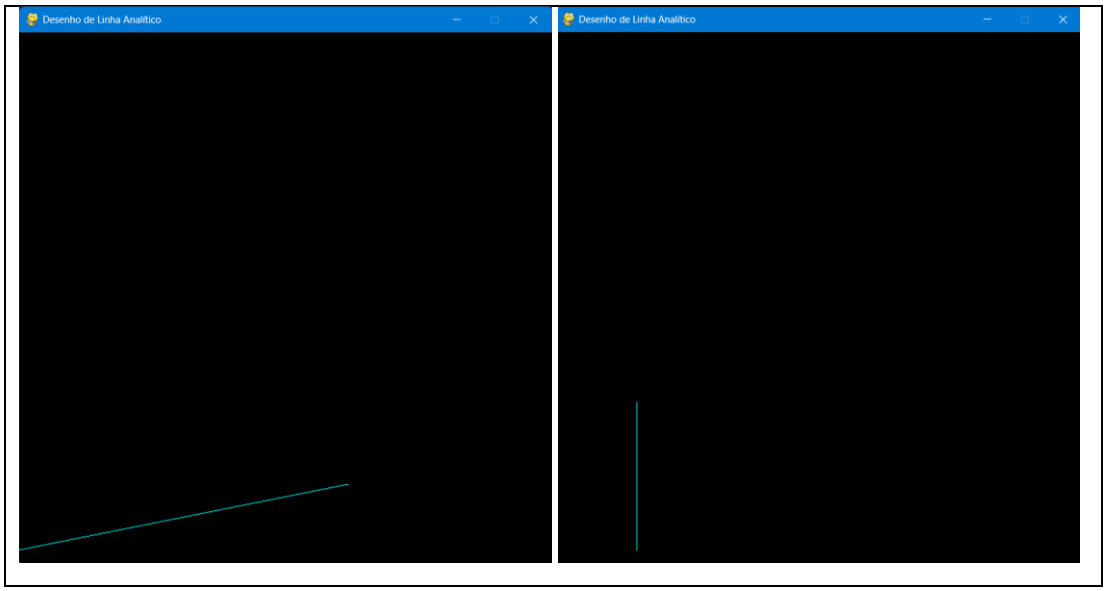
Aplicando essa fórmula aos pontos extremos da linha, conseguimos determinar todos os pontos intermediários necessários para sua rasterização.

Análise:

Embora o método seja matematicamente preciso, sua eficiência é reduzida devido ao uso de operações com ponto flutuante. Ele também pode apresentar erros ao lidar com linhas muito inclinadas, pois o arredondamento de valores pode ocasionar imprecisões.

Execução em código:

Abaixo segue exemplo de utilização do algoritmo, tendo sua visualização em um caso com a angulação extrema, e outro de ponto a ponto.



Conclusão:

O algoritmo analítico é uma abordagem viável para desenhar linhas, mas sua complexidade computacional pode torná-lo ineficiente em aplicações de alto desempenho.

5. Algoritmo DDA

Definição:

O Algoritmo Analisador Diferencial Digital (DDA) é um método utilizado para interpolação de valores ao longo de um intervalo, sendo amplamente aplicado na rasterização de linhas. Sua abordagem consiste em calcular cada novo ponto da linha de forma incremental, utilizando as seguintes equações:

$$x_i = x_{i-1} + 1$$

$$y_i = y_{i-1} + m$$

onde o coeficiente angular (**m**) é definido como:

$$m = \frac{\Delta y}{\Delta x} = \frac{y_{\text{fim}} - y_{\text{início}}}{x_{\text{fim}} - x_{\text{início}}}$$

Esse método permite que a linha seja traçada de forma contínua, adicionando incrementos em **x** e calculando o respectivo valor de **y** de maneira progressiva.

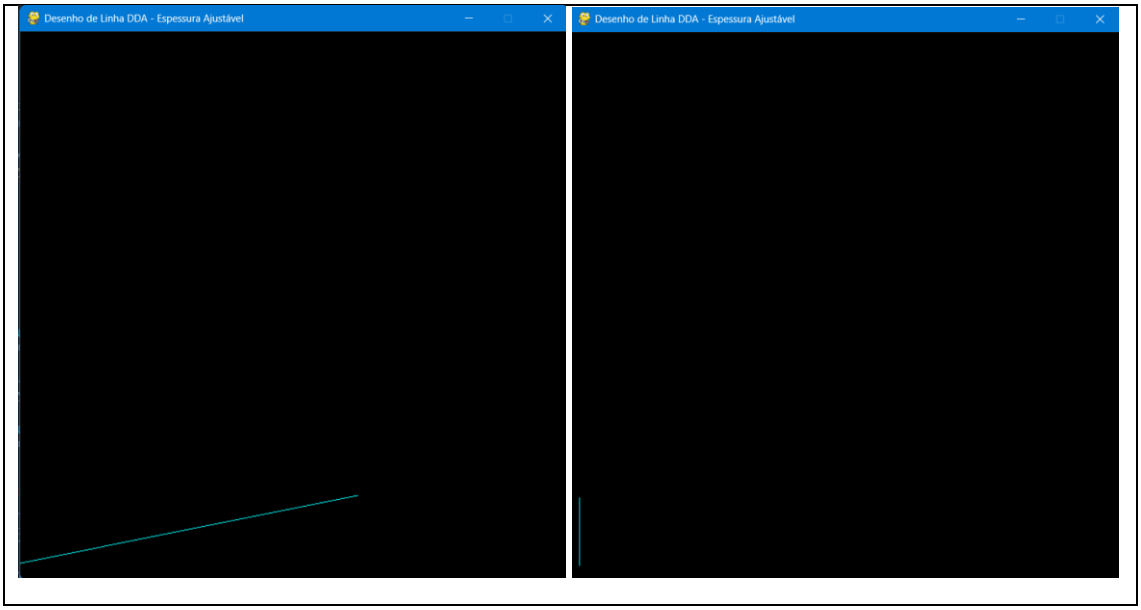
Análise:

Por utilizar números de **ponto flutuante**, o algoritmo DDA executa, a cada iteração, **duas operações simultâneas**: uma soma e um arredondamento do valor interpolado. Isso pode gerar pequenas imprecisões devido ao arredondamento dos cálculos.

Comparado ao **algoritmo analítico**, o DDA se destaca por resolver o problema de rasterização de linhas **com inclinações superiores a 45º**, que antes apresentavam falhas. Essa melhoria ocorre porque o DDA ajusta qual coordenada será incrementada a cada passo, garantindo uma melhor distribuição dos pontos ao longo da linha.

Execução em código:

Abaixo segue exemplo de utilização do algoritmo, tendo sua visualização em um caso com a angulação extrema, e outro de ponto a ponto.



Conclusão:

O algoritmo DDA equilibra precisão e eficiência, tornando-se uma escolha viável para diversas aplicações em computação gráfica. Embora resolva o problema das retas muito inclinadas, seu uso de operações de ponto flutuante e arredondamentos ainda o torna computacionalmente pesado, dificultando sua escalabilidade.

6. Algoritmo de Bresenham

Definição:

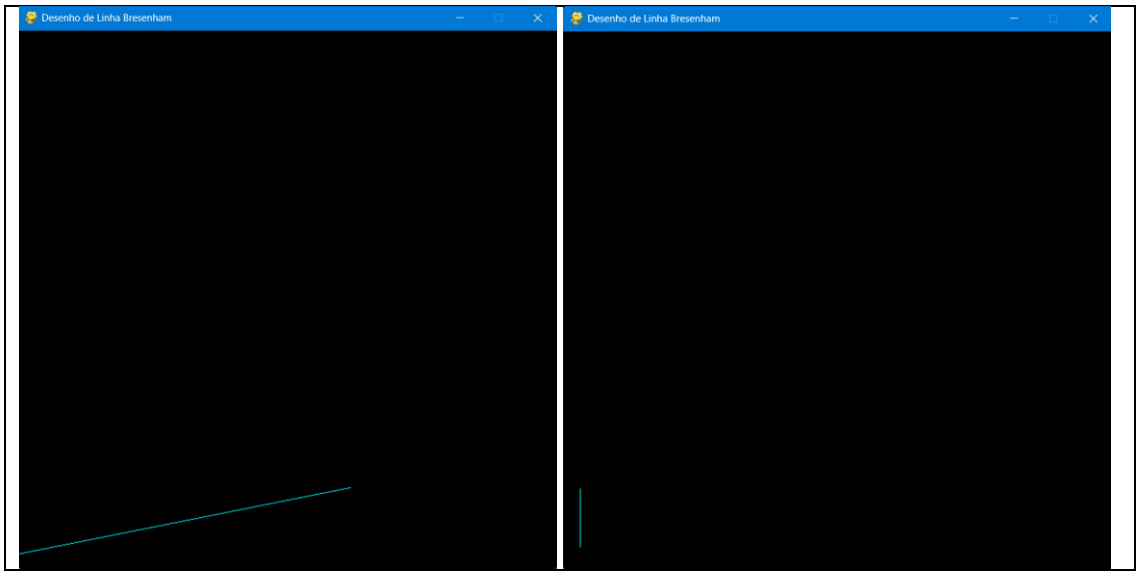
Também conhecido como **algoritmo de ponto médio**, o **Algoritmo de Bresenham** é um método eficiente para a rasterização de linhas. Ele se baseia no princípio de que os **pixels que formam uma linha devem ser contínuos**, garantindo uma transição suave entre os pontos.

Ao invés de utilizar operações com números de **ponto flutuante**, Bresenham trabalha apenas com **operações inteiras**, tornando sua execução mais rápida e eficiente. O algoritmo determina, a cada passo, qual pixel vizinho está mais próximo do caminho ideal da reta dentro do grid, minimizando erros de arredondamento.

Análise:

O algoritmo de Bresenham se destaca como a melhor escolha para a rasterização de linhas devido à sua eficiência computacional. Como utiliza apenas operações inteiras, evita os problemas de arredondamento e cálculos complexos, proporcionando melhor desempenho em comparação com os outros métodos.

Execução em código:



Conclusão:

Devido à sua velocidade, precisão e baixo custo computacional, o algoritmo de Bresenham é amplamente utilizado na computação gráfica moderna, sendo a opção mais eficiente para aplicações que exigem alto desempenho.

7. ANEXOS (Link do Github)

GitHub – Computer Grafics

8. REFERENCIAS

Computer Graphics - Line Generation Algorithm <https://www.tutorialspoint.com/computer_graphics/line_generation_algorithm.htm>