

Relatório: Algoritmos de Desenho de Linhas

Autor: Kaio Guilherme Ferraz de Sousa Silva

1. Introdução

Este relatório apresenta e compara três algoritmos usados para desenhar linhas em sistemas gráficos: o **Algoritmo Analítico**, o **Algoritmo DDA (Digital Differential Analyzer)** e o **Algoritmo de Bresenham**. Cada um desses algoritmos tem suas próprias características, vantagens e desvantagens, e são utilizados dependendo das exigências de precisão e eficiência de diferentes sistemas de renderização de gráficos.

Algoritmos Analisados

1. **Algoritmo Analítico**: Método simples para desenhar linhas utilizando a equação da reta .
$$y = mx + b$$
 2. **Algoritmo DDA (Digital Differential Analyzer)**: Algoritmo baseado em interpolação para calcular e desenhar linhas.
 3. **Algoritmo de Bresenham**: Algoritmo eficiente que evita o uso de ponto flutuante, baseando-se em decisões de erro acumulado.
-

2. Descrição do Funcionamento

2.1 Algoritmo Analítico

O algoritmo **Analítico** baseia-se na equação da reta:

$$y = m * x + b$$

onde:

- **m** é a inclinação da reta, calculada como:
$$m = (y_2 - y_1) / (x_2 - x_1)$$
- **b** é o intercepto no eixo Y, obtido por:

$$b = y_1 - (m * x_1)$$

O algoritmo percorre os valores de **x** entre **x1** e **x2**, calculando os respectivos valores de **y** e desenhando cada pixel correspondente.

Processo:

1. **Calcular a inclinação** **m** da reta.
2. **Determinar o intercepto** **b**.
3. **Percorrer os pontos entre** **x1** **e** **x2**, **calculando** **y** **para cada** **x** **usando a equação da reta.**
4. **Se a linha for vertical (** **x1 == x2** **), variar apenas** **y**, **mantendo** **x** **fixo.**

2.2 Algoritmo DDA (Digital Differential Analyzer)

O **DDA** utiliza a interpolação digital para calcular os pontos da linha. O algoritmo começa a partir de um dos pontos e, para cada incremento de **xx** ou **yy**, calcula o valor da outra coordenada, garantindo que os pontos formem uma linha reta. O DDA é mais eficiente que o método analítico para linhas de inclinação variável, mas pode sofrer de imprecisões devido ao arredondamento.

• Processo:

1. Calcula-se as diferenças **e**.
 $dx = x_2 - x_1$
 $dy = y_2 - y_1$
2. Determina-se qual coordenada (horizontal ou vertical) tem a maior diferença **e**, com isso, decide-se qual coordenada será interpolada.
3. A linha é desenhada ao atualizar a coordenada dominante, e a coordenada não dominante é calculada a partir da interpolação.

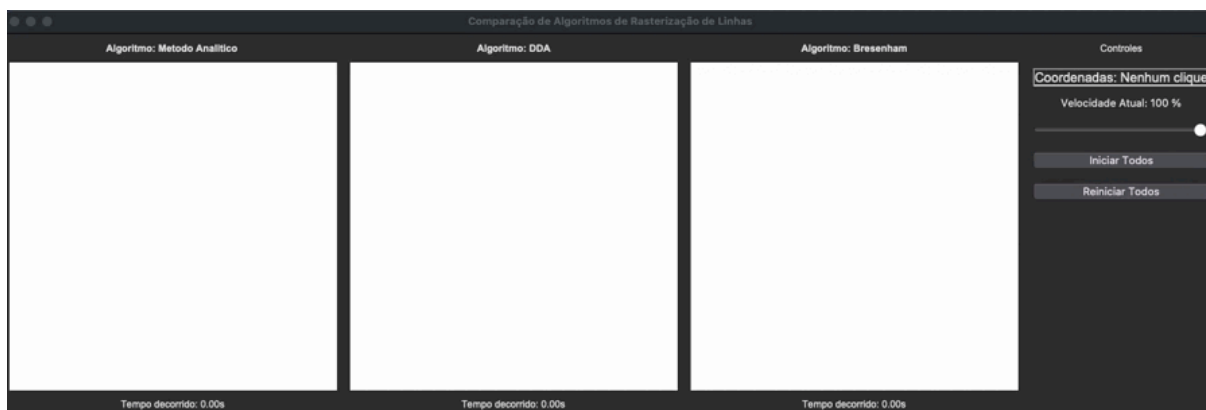
3. Diferenças Entre os Algoritmos

Característica	Algoritmo Analítico	Algoritmo DDA	Algoritmo de Bresenham
Tipo de Algoritmo	Baseado em equações	Interpolação digital	Decisão baseada em erro acumulado

	matemáticas		
Velocidade	Pode ser mais lento em linhas verticais	Eficiente para linhas inclinadas	Muito eficiente, especialmente para sistemas com limitações de hardware
Precisão	Boa, mas com possíveis imprecisões em linhas verticais	Boa, mas imprecisa devido ao arredondamento	Excelente, sem imprecisões de arredondamento

4. Análise de Desempenho

Os algoritmos **Analítico**, **DDA** e **Bresenham** são executados simultaneamente, cada um em uma thread separada para evitar interferências de desempenho entre si. O algoritmo Analítico utiliza a equação da reta para calcular diretamente os pixels, enquanto o DDA realiza a interpolação dos valores com maior precisão, porém com custo computacional adicional. Já o algoritmo de Bresenham, conhecido por sua eficiência, utiliza operações inteiras para determinar os pixels rapidamente. A execução independente de cada algoritmo pode ser observada no GIF abaixo, permitindo uma análise clara de seus desempenhos e características visuais.



Foram usadas matrizes de 400×400 pixel em escala 1:1

5. Conclusão

Cada um dos algoritmos analisados tem suas vantagens e limitações. O **Algoritmo Analítico** é simples de implementar, mas pode ser menos eficiente para linhas verticais. O **DDA** é mais flexível e pode lidar bem com diferentes

inclinações, mas sofre com imprecisões de arredondamento. O **Algoritmo de Bresenham** é o mais eficiente em termos de desempenho e precisão, tornando-se a melhor opção para sistemas gráficos com restrições de hardware ou onde a velocidade de execução é crucial. A escolha do algoritmo depende das necessidades específicas da aplicação, como a eficiência computacional e a precisão exigida para o desenho da linha.