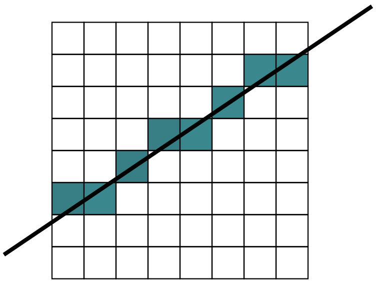
COMPUTAÇÃO GRÁFICA

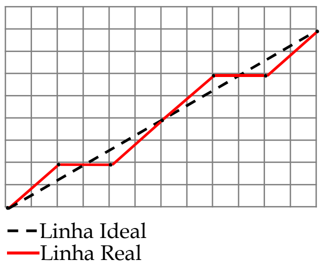
Unidade 3 – Rasterização

Ivan Nunes da Silva

Representação Vetorial x Matricial

- Normalmente, gráficos são definidos por meio de primitivas geométricas como pontos, segmentos de retas, polígonos, etc.
  - Representação vetorial
- Dispositivos gráficos podem ser pensados como matrizes de pixels (*rasters*).
  - Representação matricial
- Rasterização é o processo de conversão entre representações vetorial e matricial.





Considerações Gerais (I)

- Rasterização é um processo de amostragem:
  - Domínio contínuo → discreto
  - Problemas de *aliasing* são esperados
- Cada primitiva pode gerar um grande número de pixels .
  - Rapidez é essencial.
- Em geral, rasterização é feita por hardware.
- Técnicas de *antialiasing* podem ser empregadas, usualmente extraindo um custo em termos de desempenho.

3

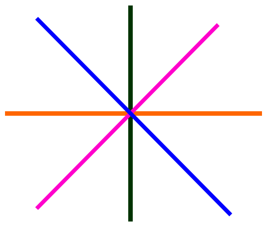
Considerações Gerais (II)

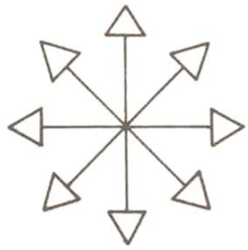
- Considerações Sobre Pontos
  - Um ponto será considerado (constituído) como um pixel.
  - O ponto será desenhado numa determinada posição da tela.
  - Atribui-se ao pixel a posição correspondente ao ponto na tela.
- Considerações Sobre Retas
  - Um segmento de reta é traçado através de pontos discretos entre os pontos extremos do segmento.
  - As coordenadas dos pontos são calculadas através da equação da reta.
  - As coordenadas dos pontos na tela são referenciadas com valores inteiros, então o segmento de reta obtido é uma aproximação do segmento real.
  - O arredondamento dos valores das coordenadas para inteiro implica numa aparência de escada (*aliasing*).

4

Considerações Gerais (III)

- Simetria e Reflexões
  - A representação matricial de segmentos de reta horizontais, verticais e diagonais a 45^0 e a 135^0, não apresentam o efeito escada.
  - Essas direções formam na verdade eixos de simetria no espaço matricial.
  - Qualquer imagem representada no espaço matricial pode ser refletida em relação a essas direções sem apresentar qualquer deformação.





5

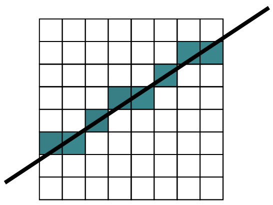
Considerações Gerais (IV)

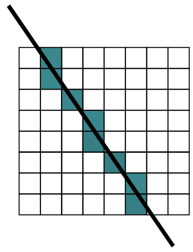
- Características Desejáveis dos Algoritmos Para Traçar Segmentos de Reta:
  - **Linearidade** → Os pixels traçados devem dar a impressão de que estão sobre uma reta.
  - **Precisão** → Os segmentos devem iniciar e terminar nos pontos especificados; caso contrário, pode-se ocorrer pequenos “vazios” entre o final de um segmento e o início de outro.
  - **Espessura (densidade) uniforme** → A densidade da linha é dada pelo número de pixels traçados, dividido pelo comprimento da linha. Para manter a densidade constante, os pixels devem estar igualmente espaçados.
  - **Intensidade independente da inclinação** → Para segmentos de diferentes inclinações.
  - **Continuidade** → A imagem não apresenta interrupções indesejáveis.
  - **Rapidez no traçado dos segmentos** → A velocidade é essencial nos algoritmos de computação gráfica.

6

Rasterização de Segmentos de Reta

- Segmento de reta entre $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$
  - Já foi recortado com relação ao *viewport (janela de visão do traçado).*
- Objetivo é pintar os pixels atravessados pelo segmento de reta.
  - Na verdade, nem todos, apenas os mais próximos.
- Reta de suporte dada por $a\,x + b\,y + c = 0$
- Queremos distinguir os casos:
  - Linhas ~ horizontais → computar y como função de x
  - Linhas ~ verticais → computar x como função de y

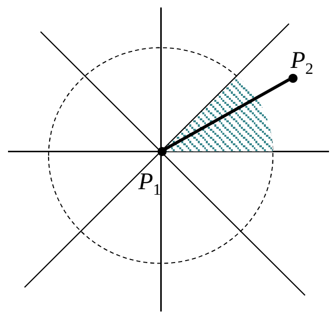




7

Algoritmo Analítico (Conceitos)

- Assumimos segmentos de reta no primeiro octante:
  - Demais casos são resolvidos de forma simétrica.
- Inclinação (entre 0 e 1) dada por: $m = (y_2 - y_1) / (x_2 - x_1)$
- Algoritmo:
  - Para x desde x_1 até x_2 fazer:
    - $y \leftarrow y_1 + (m * (x - x_1))$
    - Pintar pixel (x, y)
- Desvantagens:
  - Aritmética em ponto flutuante.
  - Utilização de multiplicação.



8

Algoritmo Analítico (Implementação)

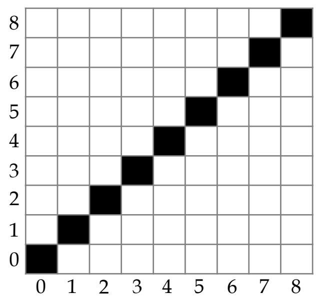
- Seja a equação da reta: $y = m.x + b$
- Se reta vertical, i.e., $x_1 = x_2$
  - Então Para y de y_1 passo 1 até y_2 faça
    - Pintar pixel (x_1 , y)
  - Senão
    - $m \leftarrow (y_2 - y_1) / (x_2 - x_1)$
    - $b \leftarrow y_2 - m*x_2$
    - Para x de x_1 passo 1 até x_2

      $y \leftarrow m*x + b$

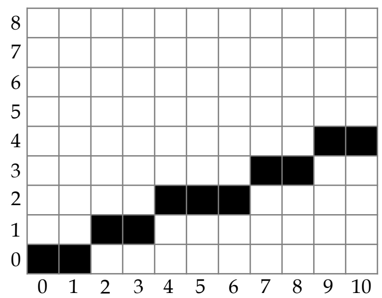
      Pintar pixel (x, y)

Algoritmo Analítico (Deslocamento)

- O Efeito Deslocamento do Algoritmo Analítico:
  - Densidade dos pontos não é constante com a inclinação.
  - Deve-se então determinar em qual dos eixos está o **maior deslocamento** e incrementar esta coordenada.

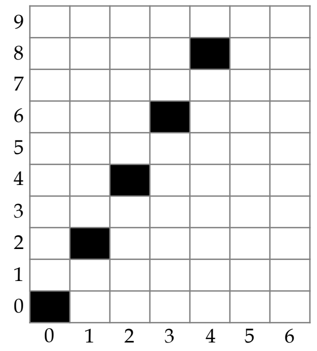


Reta entre (0,0) e (8,8) com m = 1
(INCREMENTO HORIZONTAL)

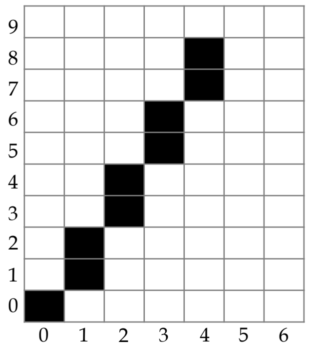


Reta entre (0,0) e (10,4) com m = 0,4
(INCREMENTO HORIZONTAL)

Algoritmo Analítico (Deslocamento)



Reta entre (0,0) e (4,8) com m = 2
(INCREMENTO HORIZONTAL)



Reta entre (0,0) e (4,8) com m = 2
(INCREMENTO VERTICAL)

11

Algoritmo Incremental (Conceitos)

- Algoritmo analítico simples tem vários problemas:
  - Utiliza aritmética de ponto-flutuante.
  - Sujeito a erros de arredondamento.
  - Usa multiplicação.
  - Muito Lento.
- Se observarmos que m é a variação em y para um incremento unitário de x, tem-se:
  - $m = \Delta y / \Delta x$, então tem-se $\Delta x = 1$
  - Logo:
    - Para $\Delta x = 1$ implica em $\Delta y = m$
- Desvantagem:
  - Utilização de aritmética em ponto flutuante.

12

Algoritmo Incremental (Implementação)

- Incrementar os eixos com maior deslocamento
- Se $(x_2 - x_1) > (y_2 - y_1)$
  - Então
    - $m \leftarrow (y_2 - y_1) / (x_2 - x_1)$
    - $y \leftarrow y_1$
    - Para x de x_1 até x_2 (passo 1)

      Pintar pixel (x , y)

      $y \leftarrow y + m$
  - Senão
    - $m \leftarrow (x_2 - x_1) / (y_2 - y_1)$
    - $x \leftarrow x_1$
    - Para y de y_1 até y_2 (passo 1)

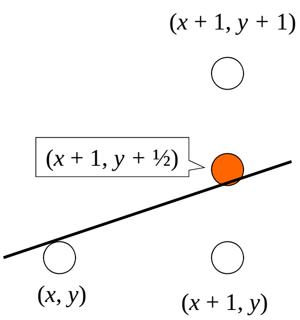
      Pintar pixel (x , y)

      $x \leftarrow x + m$

13

Algoritmo de Bresenham (I)

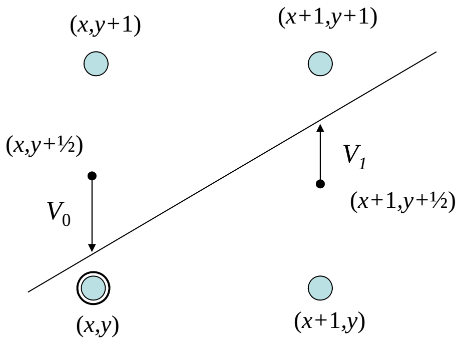
- Algoritmo clássico da computação gráfica.
- Algoritmo incremental que utiliza apenas soma e subtração de inteiros.
- Idéia básica:
  - Em vez de computar o valor do próximo y em ponto flutuante, decidir se o próximo pixel vai ter coordenadas $(x + 1, y)$ ou $(x + 1, y + 1)$
  - Decisão requer que se avalie se a linha passa acima ou abaixo do ponto médio $(x + 1, y + ½)$



14

Algoritmo de Bresenham (II)

- Variável de decisão *V* é dada pela classificação do ponto médio com relação ao semi-espaço definido pela reta.
- Caso 1: Linha passou abaixo do ponto médio:



$$ax+by+c=V$$

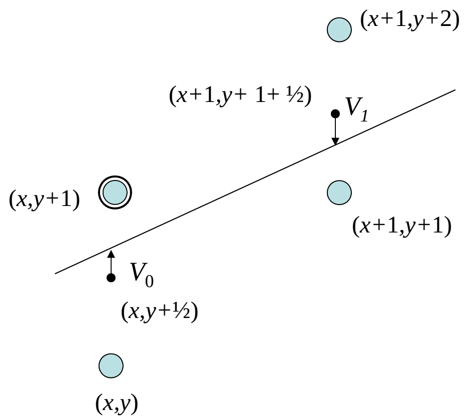
$$\text{onde}\begin{cases} V=0\rightarrow(x,y)\ \text{sobre a reta} \\ V<0\rightarrow(x,y)\ \text{abaixo da reta} \\ V>0\rightarrow(x,y)\ \text{acima da reta}\end{cases}$$

$$V_1=a(x+1)+b(y+\tfrac{1}{2})+c$$
$$V_0=ax+b(y+\tfrac{1}{2})+c$$
$$\therefore V_1=V_0+a$$

15

Algoritmo de Bresenham (III)

- Caso 2: Linha passou acima do ponto médio:



$$V_1=a(x+1)+b(y+1+\tfrac{1}{2})+c$$
$$V_0=ax+b(y+\tfrac{1}{2})+c$$
$$\therefore V_1=V_0+a+b$$

16

Algoritmo de Bresenham (IV)

- Dedução dos Coeficientes da Reta: $a.x + b.y + c = 0$
  - Obtenção dos parâmetros a e b
    - $m = (y_2 - y_1)/(x_2 - x_1) \Leftrightarrow$ inclinação $\Leftrightarrow y = m.x$
    - $(y_2 - y_1).x - (x_2 - x_1).y = 0$
    - $\underbrace{(y_2 - y_1)}_{a}.x + \underbrace{(x_1 - x_2)}_{b}.y = 0$
  - Substituindo a e b na equação da reta suporte:
    - $a.x + b.y + c = 0$
    - $c = -a.x - b.y$
    - $c = -(y_2 - y_1).x - \underbrace{(x_1 - x_2)}_{-1}.y$; mas $x_2 - x_1 = 1$
    - $c = (y_1 - y_2).x + y$; Substituindo $(x, y) = (x_1, y_1) = (x_2 - 1, y_1)$
    - $c = (y_1 - y_2).(x_2 - 1) + y_1 = y_1.x_2 - \cancel{y_1} - y_2.x_2 - y_2 + \cancel{y_1}$
    - $c = y_1.x_2 - y_2.x_2 - y_2 = y_1.x_2 - y_2.(x_2 - 1)$; mas $x_1 = x_2 - 1$
    - $c = y_1.x_2 - y_2.x_1$

17

Algoritmo de Bresenham (V)

- Coeficientes da reta: $a.x + b.y + c = 0$
  - $a = y_2 - y_1$
  - $b = x_1 - x_2$
  - $c = x_2 y_1 - x_1 y_2$
- Para iniciar o algoritmo, precisamos saber o valor de V em $(x_1 + 1, y_1 + ½)$
  - $V = a\,(x_1 + 1) + b\,(y_1 + ½) + c$
    $= \underbrace{a\,x_1 + b\,y_1 + c}_{0} + a + b/2 = a + b/2$
- Podemos evitar a divisão por 2 multiplicando a, b e c por 2 (não altera a equação da reta)

18

Algoritmo de Bresenham (Implementação)

$a \leftarrow y_2 - y_1$
$b \leftarrow x_1 - x_2$
$V \leftarrow 2 * a + b$
$x \leftarrow x_1$
$y \leftarrow y_1$
Enquanto $x \leq x_2$ fazer:

- Pintar pixel (x, y)
- $x \leftarrow x + 1$
- Se $V \leq 0$
  - Então: $V \leftarrow V + 2 * a$; {não altera posição de y}
  - Senão $\begin{cases} V \leftarrow V + 2 * (a + b) \\ y \leftarrow y + 1 \end{cases}$

Fim_Enquanto

19

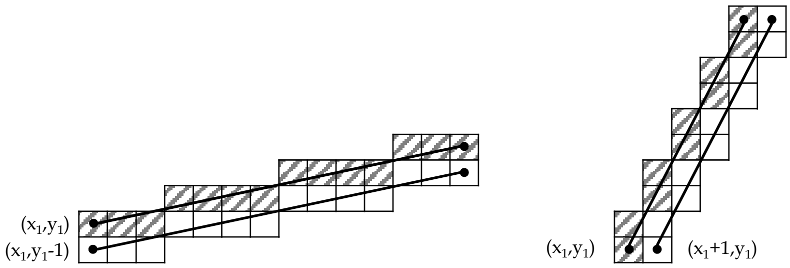
Extensão para demais Octantes

- Se $x_2 < x_1$
  - Trocar P_1 com P_2
- Se $y_2 < y_1$
  - $y_1 \leftarrow - y_1$
  - $y_2 \leftarrow - y_2$
  - Pintar pixel $(x, -y)$
- Se $|y_2 - y_1| > |x_2 - x_1|$
  - Repetir o algoritmo trocando "y" com "x"

20

Linha Com Atributos Particulares

- Linha com Espessura

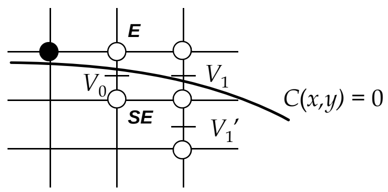


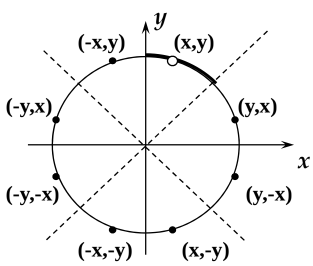
a) $\Delta x > \Delta y$

b) $\Delta y > \Delta x$

Rasterização de Círculos

- Mesma idéia de avaliar incrementalmente uma função que classifica o ponto médio entre um pixel e outro com relação a uma função implícita.
- Apenas um octante precisa ser avaliado, os demais são simétricos (simetria de ordem 8):
  - Para cada pixel computado, oito são pintados
- Derivação um pouco mais difícil que a da reta.
- Outras cônicas podem também ser rasterizadas de forma semelhante.





Aplicação do Algoritmo de Bresenham

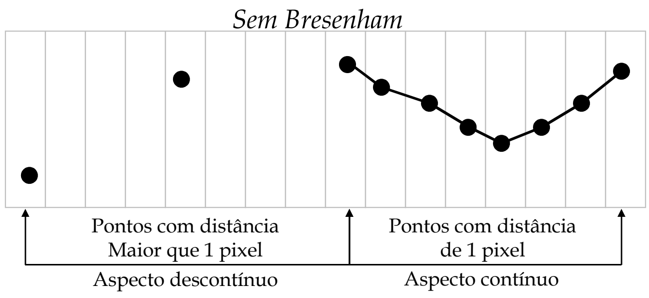
Confecção de Gráficos on-line

- Propósito
  - **Construir gráficos dinamicamente e mostrá-los ao usuário na forma de página de internet.**
    - **D.H. Spatti, P.R. de Aguiar, F.R.L. Dotto (Projeto de Iniciação científica).**
  - **A partir de uma base de dados TXT, os pontos são manipulados e ajustados para ser impressos em um gráfico para acompanhamento de processos de usinagem.**
  - **Para que o sistema não ficasse sobrecarregado, utilizou-se apenas imagens estáticas em formato GIF, contendo os pixels e tabelas HTML.**
  - **Utilizou-se o algoritmo de Bresenham para preencher espaços entre pontos.**

23

Aplicação do Algoritmo de Bresenham

Confecção de Gráficos on-line



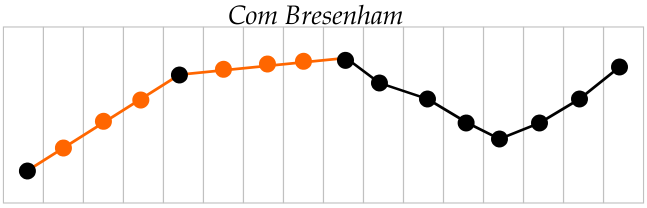
Para corrigir a descontinuidade, o intervalo entre pontos com espaçamento maior que 1 pixel foi preenchido com retas, geradas pela inserção de pontos, através do algoritmo de Bresenham.

24

Aplicação do Algoritmo de Bresenham

Confecção de Gráficos on-line

Com Bresenham



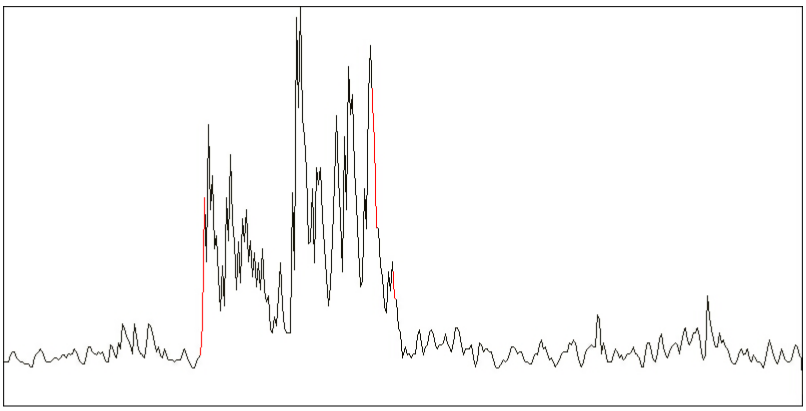
- Pontos Originais
- Pontos inseridos com Bresenham
- Linha com aspecto contínuo original
- Linha com aspecto contínuo, produzida por Bresenham

25

Aplicação do Algoritmo de Bresenham

Confecção de Gráficos on-line

Tela Mostrada ao Usuário



- Linha original
- Linha produzida por Bresenham

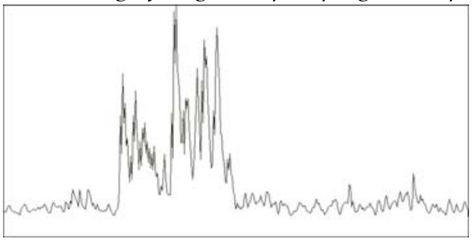
26

Aplicação do Algoritmo de Bresenham

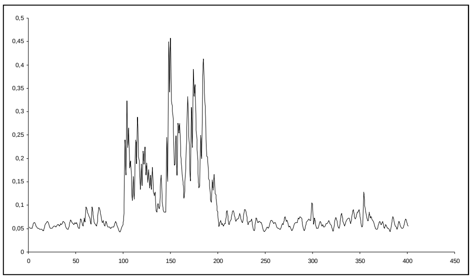
Confecção de Gráficos on-line

Comparação entre o gráfico gerado pelo programa e pelo Excel

Gráfico on-line



Excel



27

Aplicação do Algoritmo de Bresenham

Classificação de Padrões de Café

- Propósito
  - **Classificar grãos de café relacionando-os em 4 grupos distintos {verde, cereja, passa, maduro}.**
    - **I. N. Silva, R. A. Flauzino, J. A. Ulson (Proj. Pesquisa).**
  - **Quantificar automaticamente os volumes dos tipos de café que foram colhidos pela máquina agrícola.**
  - **O Algoritmo de Bresenham foi empregado para se traçar o resultado gráfico da quantificação, no display da máquina agrícola.**
  - **Devido à natureza dos recursos computacionais disponíveis no implemento agrícola, todas as funções gráficas tiveram que ser implementadas manualmente, incluindo traçar gráficos na tela.**

28

Aplicação do Algoritmo de Bresenham

Classificação de Padrões de Café

- Colheitadeira de Café

29

Aplicação do Algoritmo de Bresenham

Classificação de Padrões de Café

- Colheitadeira de Café

30