

Geração de Primitivas

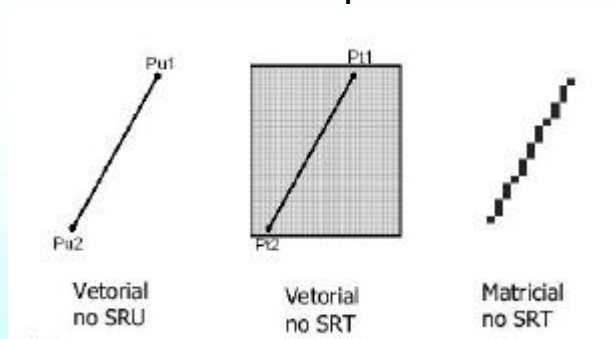
■ Geração de linhas:

- utilização de dispositivos matriciais, com pontos discretos;
- processo: rasterização;
- não há problemas para linhas horizontais, verticais ou com inclinação de 45° ;
- problema básico: qual o pixel ideal para uma linha desejada (?);
- soluções melhores para dispositivos com maior resolução.

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Rasterização

- Converter uma definição geométrica para pixels
- *rasterizar* = escolher pixels!!



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Rasterização

■ Linhas:

- Método Analítico
- Método DDA
- Algoritmo de Bresenham
- Bresenham modificado com antialiasing

■ Circunferência:

- Equação Paramétrica
- Incremental com simetria
- Algoritmo de Bresenham

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Método Analítico

- Dados os pontos extremos de uma linha na tela:
 - $P1(x1,y1)$ $P2(x2,y2)$
- Obtenha a equação da reta que passa pelos pontos, chegando a $ax + by + c = 0$
- calcule o coeficiente de inclinação da reta:
 - $y = mx + b$
 - $m = (y2 - y1) / (x2 - x1)$
 - $b = y1 - m.x1$
- ligue todos os pontos que pertencem à reta, por meio de um **for**:

$\text{for } x = x1 \text{ to } x2$
 $y = m * x + b$
 $\text{ponto}(x,y)$

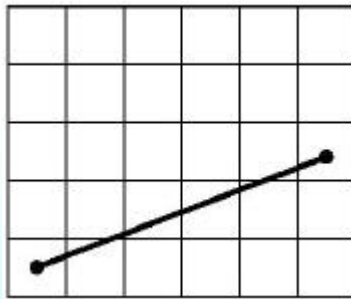
Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Método Analítico

- Suponha a linha definida pelos pontos:

$P1(0,0)$ $P2(5,2)$

Quais pixel são ligados?



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Método Analítico

- Problemas:

- operações com ponto flutuante, no entanto, pixel são inteiros
- muitos cálculos no processo - eficiência (?)
- escolha do pixel não é um fator considerado na elaboração da solução, que pode ser qualquer um das redondezas do número obtido nas contas efetuadas

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: método DDA

- Técnica baseada na solução da equação diferencial da reta:

$$dy / dx = k \text{ (constante) ou}$$

$$\text{Dy} / \Delta x = (y_2 - y_1) / (x_2 - x_1)$$

Assim, temos:

$$y_{i+1} = y_i + \text{Dy}$$

$$y_{i+1} = y_i + (y_2 - y_1) / (x_2 - x_1) \cdot Dx$$

Atenção: escolher a variação mais adequada!!

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: método DDA

$ x_2 - x_1 > y_2 - y_1 $	
Sim	Não
$\text{incremento} = y_2 - y_1 / x_2 - x_1$ $y = y_1$ p/ x de x_1 até x_2 faça dot (x,y,cor) $y = y + \text{incremento}^{(*)}$	$\text{incremento} = x_2 - x_1 / y_2 - y_1$ $x = x_1$ p/ y de y_1 até y_2 faça dot (x,y,cor) $x = x + \text{incremento}^{(*)}$
(*) determina a densidade da linha	

Problemas: truncamentos e arredondamentos

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

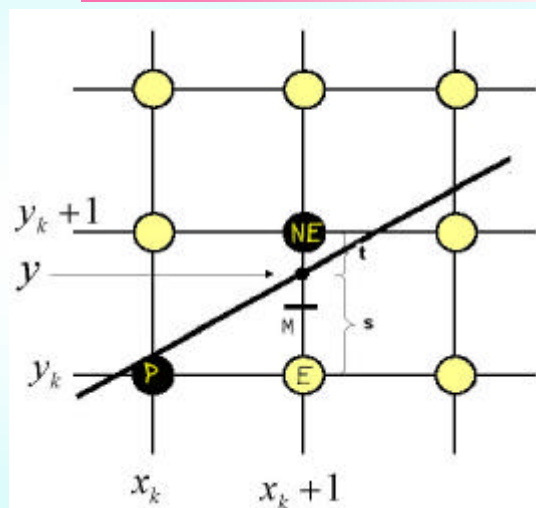
Linhas: Algoritmo de Bresenham

■ Princípios básicos:

- trabalha só com inteiros
- usa um incremento unitário
- aproveita a coerência espacial: similaridade de valores referentes a pixel vizinhos
- escolha entre dois valores de pixel vizinhos
- usa uma variável ***di*** para tomada de decisão entre os pixel vizinhos
- ***di*** é proporcional à uma diferença entre parâmetros *s* e *t* (figura)

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Algoritmo de Bresenham



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Algoritmo de Bresenham

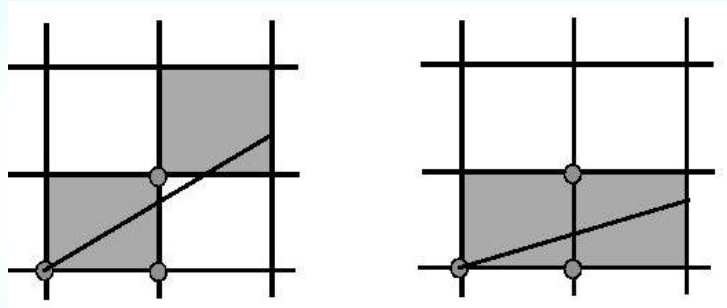
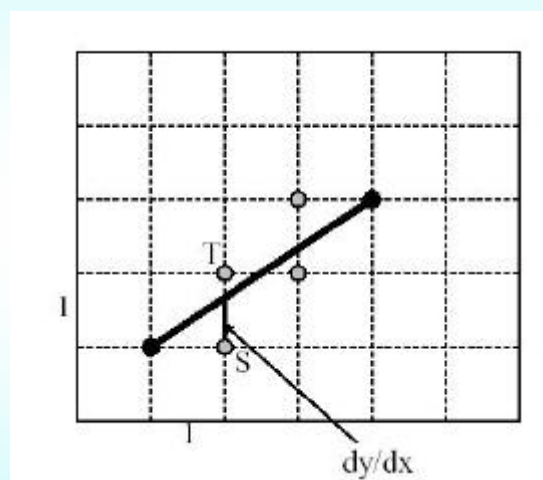


Figura: duas situações distintas de possibilidades para escolha do pixel adequado

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Algoritmo de Bresenham



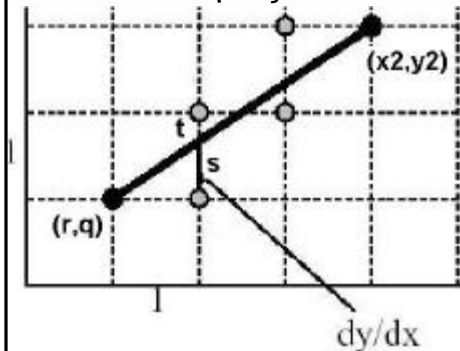
Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Algoritmo de Bresenham

■ Linha: P1(x1,y1) até P2(x2,y2)

$$dx = x2 - x1 \text{ e } dy = y2 - y1$$

equação da reta: $y = dy/dx * x$, nota-se que:



$$\begin{aligned} ys - q &= s, \text{ mas,} \\ ys &= dy/dx.(r+1), \text{ logo} \\ s &= dy/dx.(r+1) - q \end{aligned}$$

$$\begin{aligned} t &= q+1 - yt, \text{ mas,} \\ yt &= dy/dx. (r+1), \text{ logo,} \\ t &= (q+1) - dy/dx.(r+1) \end{aligned}$$

$$\Rightarrow s - t = 2. dy/dx - 2q - 1$$

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Algoritmo de Bresenham

daí, temos, $dx.(s - t) = 2.(r.dy - q.dx) + 2dy - dx$;

chamando $dx(s - t) = d_i$

$$d_i = 2.(r.dy - q.dx) + 2dy - dx; \text{ mas}$$

$$\Rightarrow d_i + 1 - d_i = 2dy - 2dx(y_i - y_{i-1})$$

$$d_{(i+1)} = d_i + 2dy - 2dx.(y_i - y_{i-1})$$

$$\text{lembrando que } d_i = dx.(s - t); \begin{cases} \text{se } s \geq t \Rightarrow d_i \geq 0 - T_i \\ \text{se } s < t \Rightarrow d_i < 0 - S_i \end{cases}$$

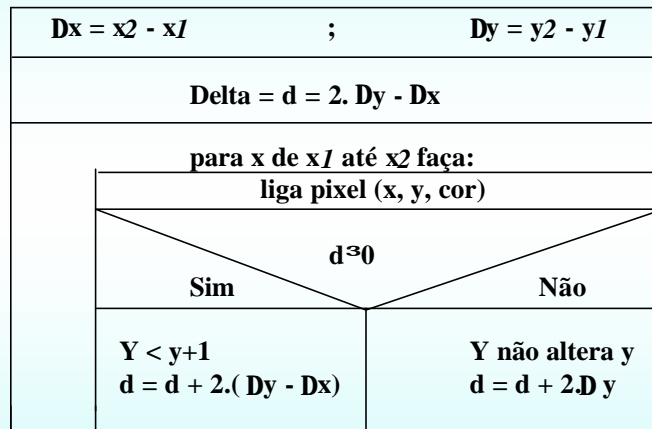
i) Se $d_i \geq 0 \Rightarrow$ tomamos T_i ($y_i = y_{i-1} + 1 = d_i + 2.(dy - dx)$)

ii) Se $d_i < 0 \Rightarrow$ tomamos S_i ($y_i = d_i + 1 = d_i + 2.dy$)

Para ($i = 1$), como $(x_0, y_0) = (0,0) \Rightarrow d_i = 2.dy - dx$, por 1

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Linhas: Algoritmo de Bresenham



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

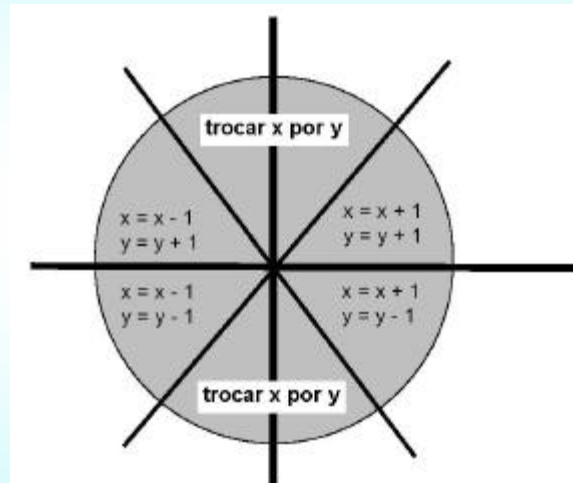
Linhas: Algoritmo de Bresenham

ALGORITMO BRES_INT (x0, y0, x1, y1)

1. $x = x1$; $y = y1$;
2. $dy = y1 - y0$; $dx = x1 - x0$;
3. $m = dy/dx$;
4. $e = 2dy - dx$;
5. FOR $i=1$ TO dx
 6. WritePixel(x, y);
 7. IF ($e > 0$) {
 8. $y = y+1$;
 9. $e = e - 2dx$; }
 10. ELSE $e = e + 2dy$;
 11. $x = x+1$;
12. ENDFOR

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Bresenham - Outros Octantes



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Circunferências: Eq. Paramétrica

- Problemas similares à geração das retas
- primeira solução: eq. Paramétrica

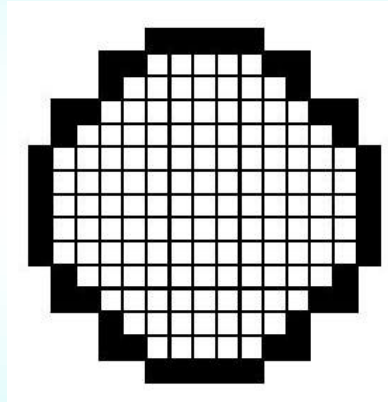
$$y = r \cdot \sin(t)$$

$$x = r \cdot \cos(t)$$

$x = \text{raio}$	$y = 0$
para t de 1 até 360 com passo "t"	
pixel (x , y , cor) $x = r \cdot \cos(\frac{p-t}{180})$ $y = r \cdot \sin(\frac{p-t}{180})$	

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

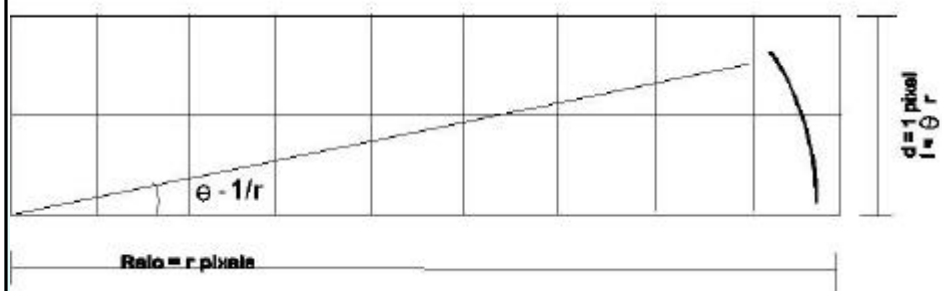
Circunferências: Eq. Paramétrica



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Alg. Incremental com Simetria

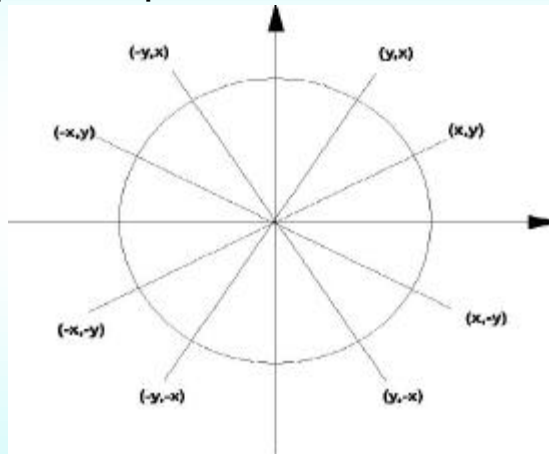
- Fundamento: deslocamento angular com incremento de uma unidade de pixel
 - deslocamento em radianos = $1/r$ (r = raio)



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Alg. Incremental com Simetria

- Propriedade importante: a circunferência é simétrica



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Alg. Incremental com Simetria

- Solução proposta: algoritmo incremental com deslocamento angular constante e pequeno, com a rotação a partir de um ponto inicial

$$\begin{cases} x_{n+1} = x_n \cdot \cos \theta + y_n \cdot \sin \theta \\ y_{n+1} = y_n \cdot \cos \theta - x_n \sin \theta \end{cases}$$

- Problemas:

- erros cumulativos, em função do uso de x_n e y_n nas iterações seguintes
- uso de números reais - necessidade do arredondamento, para cada pixel

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Alg. Incremental com Simetria

```
x = r, y = 0;
% gera pontos sobre o eixo:
pixel (x , y , cor),
pixel (-x , y , cor),
pixel (x , -y , cor),
pixel (-x , -y , cor).
% demais pontos :
x = r cos t,
y = r sen t,
pixel (x , y , cor),
pixel (x , -y , cor),
pixel (-x , y , cor),
pixel (-x , -y , cor),
pixel (y , x , cor),
pixel (y , -x , cor),
pixel (-y , -x , cor).
```

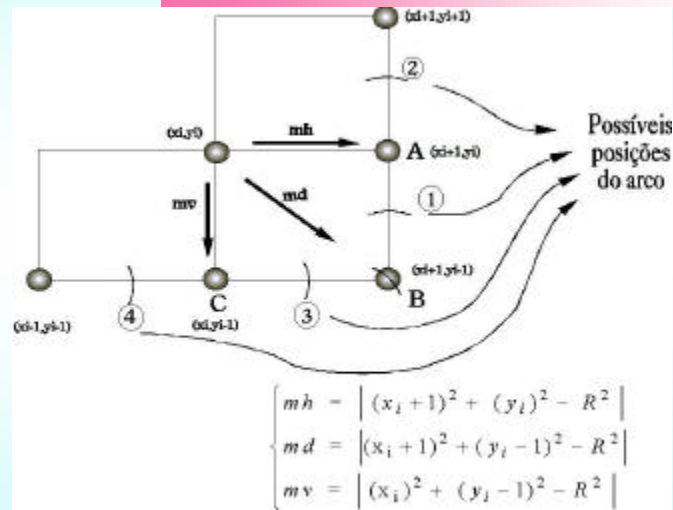
Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Bresenham

- Utiliza a simetria da circunferência
- evita utilizar raízes, potências e funções trigonométricas
- gera o primeiro quadrante e os demais por simetria
- começa do ponto (0,R) e vai diminuindo x exaustivamente, ou começa do ponto (R,0) e aumenta y gradativamente
- escolha recai sobre três pixels, na tentativa de selecionar o que está mais próximo da curva ideal
- o critério de seleção entre tais pontos leva em conta a distância relativa entre os mesmos e a circunferência ideal

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Bresenham



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Bresenham

Para minimizar o quadrado da distância entre um destes pixel e o círculo verdadeiro (mv, md, mh).

Chamando Δ_i = diferença entre o quadrado da distância do pixel ao centro e o raio da circunferência, temos: $\Delta_i = (x_i+1)^2 + (y_i-1)^2 - R^2$ (pixel diagonal)

1º) caso $\Delta_i < 0$ - O ponto B está no interior do círculo e deve-se escolher outro ponto: caso 1 ou 2.

Se $d = mh - md$ e $d = md - mv$ e

- se $d \leq 0 \Rightarrow$ o ponto escolhido é o A;
- se $d > 0 \Rightarrow$ o ponto escolhido é o B;

2º) caso $\Delta_i > 0$ - O ponto B está fora da circunferência (caso 3 ou 4)

- se $d \leq 0 \Rightarrow$ o ponto escolhido é o B;
- se $d > 0 \Rightarrow$ o ponto escolhido é o C;

3º) caso $\Delta_i = 0$ - ponto B

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br