Conversão Matricial - Circunferências e Elipses

Profa. Ana Luísa D. Martins Lemos

March 28, 2018

- Circunferência com centro na origem e raio R
  - Forma explícita

$$x^2 + y^2 = R^2$$

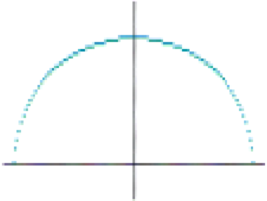
  - Forma paramétrica

$$x = R\cos\theta$$

$$y = R\sin\theta$$

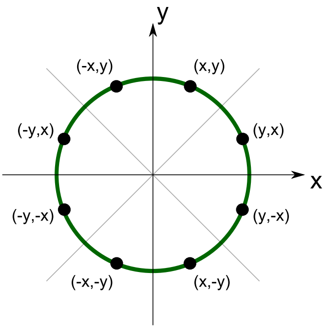
- Por que não usar a equação explícita para traçar $1/4$ da circunferência?
- Por que não usar a forma paramétrica?

- Incrementos unitários em x

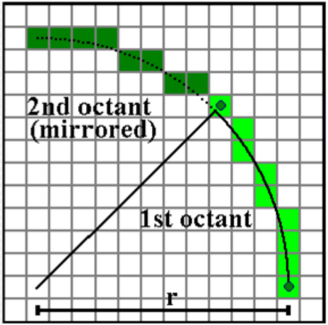


Algoritmo de Bresenham (Circunferências)

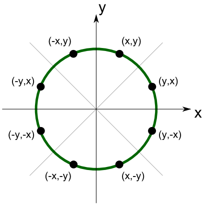
- Traçado de arco de 45^0 no segundo octante, de $x = 0$ a $x = y = R/\sqrt{2}$
- O restante da curva pode ser obtido por simetria
  - Se o ponto (x, y) pertence à circunferência, outros 7 pontos podem ser obtidos de maneira trivial.



- Simetria



- Simetria de ordem 8




```
void CirclePoints (int x, int y, int value){
  write_pixel( x, y,value);
  write_pixel( x,-y,value);
  write_pixel(-x, y,value);
  write_pixel(-x,-y,value);
  write_pixel( y, x,value);
  write_pixel( y,-x,value);
  write_pixel(-y, x,value);
  write_pixel(-y,-x,value);
}
```

Algoritmo de Bresenham (Circunferências)

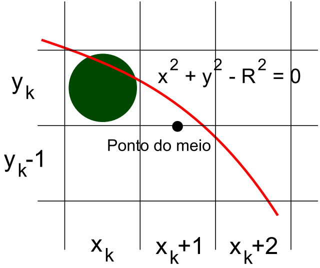
- Define um parâmetro de decisão para definir o pixel mais próximo da circunferência
- Como a equação da circunferência é não linear, raízes quadradas serão necessárias para se calcular distâncias dos pixels
    - Bresenham evita isso comparando o quadrado das distâncias
- Com base na equação da circunferência, define-se qual o pixel mais próximo da mesma
    - Isso é feito em um único octante, o resto é obtido por simetria

Algoritmo de Bresenham (Circunferências)

- $F_{\text{circ}}(x, y) = x^2 + y^2 - R^2$
  - $F_{\text{circ}}(x, y) < 0$ se (x, y) está dentro da circunferência
  - $F_{\text{circ}}(x, y) = 0$ se (x, y) está na circunferência
  - $F_{\text{circ}}(x, y) > 0$ se (x, y) está fora da circunferência
- Incrementa x e testa se pixel está mais perto da circunferência
  - $F_{\text{circ}}(x, y)$ é o parâmetro de decisão e cálculos incrementais podem ser feitos

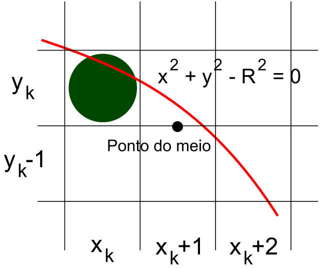
Algoritmo de Bresenham (Circunferências)

- Partindo de (x_k, y_k) as opções são
  - $(x_k + 1, y_k)$
  - $(x_k + 1, y_k - 1)$



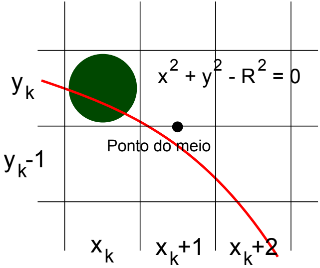
Algoritmo de Bresenham (Circunferências)

- Então a função de decisão é
  - $p_k = F_{\text{circ}}(x_k + 1, y_k - 1/2)$
  - $p_k = (x_k + 1)^2 + (y_k - 1/2)^2 - R^2$
- Se $p_k < 0$ o ponto está dentro da circunferência e y_k está mais próximo da borda
- Caso contrário, $y_k - 1$ está mais próximo



Algoritmo de Bresenham (Circunferências)

- Então a função de decisão é
  - $p_k = F_{circ}(x_k + 1, y_k - 1/2)$
  - $p_k = (x_k + 1)^2 + (y_k - 1/2)^2 - R^2$
- Se $p_k < 0$ o ponto está dentro da circunferência e y_k está mais próximo da borda
- Caso contrário, $y_k - 1$ está mais próximo



Algoritmo de Bresenham (Circunferências)

- A formulação incremental pode ser feita avaliando $x_{k+1}+1 = x_k + 2$

$$p_{k+1} = F_{\text{circ}}(x_{k+1}+1, y_{k+1}-1/2)$$

$$p_{k+1} = [(x_k+1)+1]^2 + (y_{k+1}-1/2)^2 - R^2$$

$$p_{k+1} = p_k + 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1}-y_k) + 1$$

- Se $p_k < 0$, então o próximo ponto é (x_{k+1}, y_k)

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

- Caso contrário será (x_k+1, y_k-1)

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$

com $2x_{k+1} = 2x_k + 1$ e $2y_{k+1} = 2y_k - 2$

Algoritmo de Bresenham (Circunferências)

- O parâmetro de decisão inicial é obtido encontrando o valor de F_{circ} na posição inicial $(x_0, y_0) = (0, R)$

$$p_0 = F_{\text{circ}}(1, R - 1/2)$$

$$p_0 = 5/4 - R$$

- Caso R seja especificado como um inteiro, p_0 pode ser dado por

$$p_0 = 1 - R$$

Algoritmo de Bresenham (Circunferências)

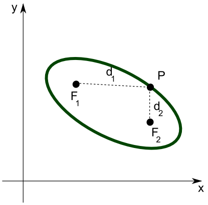
1. Entre com o raio R e o centro da circunferência (x_c, y_c), e atribua as coordenadas para o primeiro ponto
$$(x_0, y_0) = (0, R)$$
2. Calcule o valor inicial do parâmetro de decisão
$$p_0 = 5/4 - R$$
3. Para cada posição x_k, iniciando em $k = 0$, execute o seguinte teste. Se $p_k < 0$, o próximo ponto é $(x_k + 1, y_k)$ e
$$p_{k+1} = p_k + 2x_{k+1} + 1$$
caso contrário, o próximo ponto é $(x_k + 1, y_k - 1)$ e
$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$
onde $2x_{k+1} = 2x_k + 2$ e $2y_{k+1} = 2y_k - 2$
4. Determine os pontos de simetria nos outros 7 octantes
5. Mova cada posição de pixel calculado (x, y) para o caminho da circunferência de centro (x_c, y_c) e plote os valores das coordenadas
$$x = x + x_c, \; y = y + y_c$$
6. Repita os passos de 3 a 5 até que $x \geq y$

- Exercícios:
  1. Considerando $R = 10$, demonstre o algoritmo de Bresenham para traçado de circunferências determinando as posições ao longo do segundo octante de $x = 0$ a $x = y$.
  2. Execute o algoritmo de Bresenham para traçado de uma circunferência de centro em $(5, 6)$ e raio 4.

Algoritmo de Bresenham (Elipse)

- Distância de qualquer ponto aos focos é constante

$$d_1 + d_2 = \text{constante}$$



- Seja um ponto qualquer da elipse $P = (x, y)$ e os focos $F_1 = (x_1, y_1)$ e $F_2 = (x_2, y_2)$

$$\sqrt{(x - x_1)^2 + (y - y_1)^2} + \sqrt{(x - x_2)^2 + (y - y_2)^2} = \text{constante}$$

Algoritmo de Bresenham (Elipse)

$$\sqrt{(x - x_1)^2 + (y - y_1)^2} + \sqrt{(x - x_2)^2 + (y - y_2)^2} = \text{constante}$$

- Essa equação pode ser reescrita como

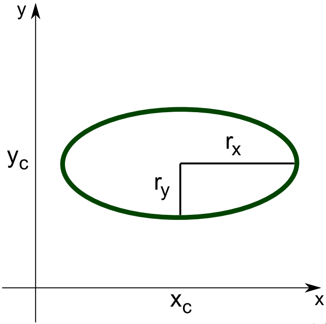
$$Ax^2 + By^2 + Cxy + Dx + Ey + F = 0$$

- A, B, C, D, E e F identificados em função das coordenadas dos focos e das dimensões do eixo maior e do eixo menor. Eixo maior, seguimento de reta que passa pelos focos.

- Cálculo simplificado se os eixos da elipse estão alinhados com os eixos do sistema de coordenadas

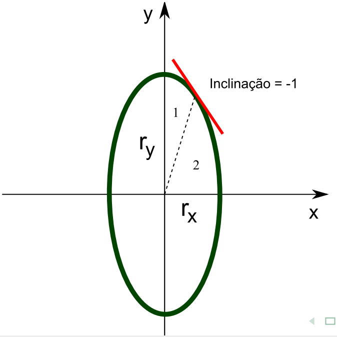
$$\left(\frac{x - x_c}{r_x}\right)^2 + \left(\frac{y - y_c}{r_y}\right)^2 = 1$$

- Simetria em relação aos quadrantes

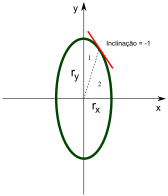


Algoritmo de Bresenham (Elipse)

- Algoritmo semelhante ao de traçado de circunferências
- Aplicado ao primeiro quadrante em duas partes
    - Incrementos unitários em x quando a inclinação é menor que 1
    - Incrementos unitários em y quando a inclinação é maior que 1



- Regiões 1 e 2 podem ser processadas de várias formas
  - Começar em $(0, r_y)$ e seguir em sentido horário com passos unitários em x até que a inclinação se torne menor que -1. A partir daí, seguir com passos unitários em y até o final do primeiro quadrante.
  - Começar em $(r_x, 0)$ e seguir em sentido anti horário com passos unitários em y até que a inclinação se torne maior que -1. A partir daí seguir com passos unitários em x até o final do primeiro quadrante.



Algoritmo de Bresenham (Elipse)

$$\left(\frac{x - x_c}{r_x}\right)^2 + \left(\frac{y - y_c}{r_y}\right)^2 = 1$$

- Considerando $(x_c, y_c) = (0, 0)$

$$F_{\text{elipse}}(x, y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$$

  que possui as seguintes propriedades
  - $F_{\text{elipse}}(x, y) < 0$ se (x, y) está dentro da elipse
  - $F_{\text{elipse}}(x, y) = 0$ se (x, y) está na elipse
  - $F_{\text{elipse}}(x, y) > 0$ se (x, y) está fora da elipse

Algoritmo de Bresenham (Elipse)

- A inclinação da elipse é calculada como

$$\frac{dy}{dx} = -\frac{2r_y^2 x}{2r_x^2 y}$$

e no limite entre as regiões 1 e 2, $dy/dx = -1$, ou seja

$$2r_y^2 x = 2r_x^2 y$$

- Saio da região 1 quando

$$2r_y^2 x \geq 2r_x^2 y$$

Algoritmo de Bresenham (Elipse)

- Exercício
  1. Implementar o algoritmo de Bresenham para o traçado de circunferências, considerando que estas podem estar centralizadas na origem ou fora da origem
  2. Desenvolver e implementar o algoritmo de Bresenham para o traçado de elipses, considerando que estas podem estar centralizadas na origem ou fora da origem