

Bài 3
GIẢI THUẬT SINH
CÁC THỰC THỂ CƠ SỞ



NỘI DUNG

1. Biểu diễn đoạn thẳng
 - a) Thuật toán DDA
 - b) Giải thuật Bresenham
 - c) Giải thuật trung điểm
2. Biểu diễn đường tròn
3. Biểu diễn đường ê-líp
4. Giải thuật sinh ký tự

Rời rạc hóa điểm ảnh

- Là tiến trình sinh các đối tượng hình học cơ sở bằng phương pháp xấp xỉ dựa trên lưới phân giải của màn hình
- Tính chất các đối tượng cần đảm bảo :
 - mượt - smooth
 - liên tục - continuous
 - đi qua các điểm xác định
 - độ sáng đồng nhất
 - hiệu quả - efficient



1

BIỂU DIỄN ĐOẠN THẲNG

•Biểu diễn tường minh

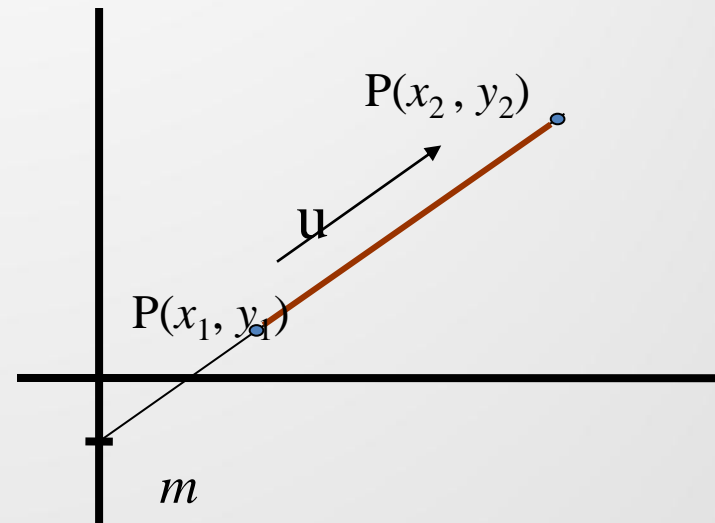
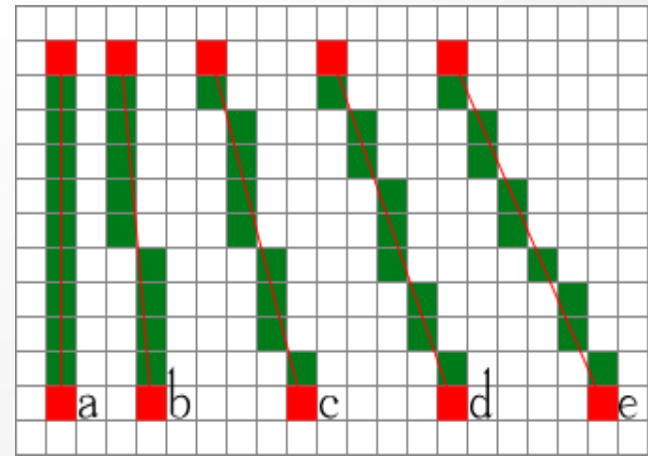
$$(y-y_1)/(x-x_1) = (y_2-y_1)/(x_2-x_1)$$

$$y = kx + m$$

$$- k = (y_2-y_1)/(x_2-x_1)$$

$$- m = y_1 - kx_1$$

$$- \Delta y = k \Delta x$$



Biểu diễn đoạn thẳng

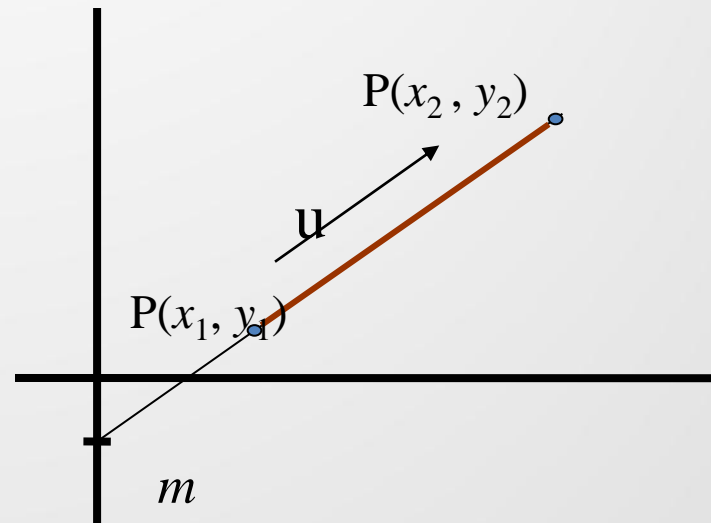
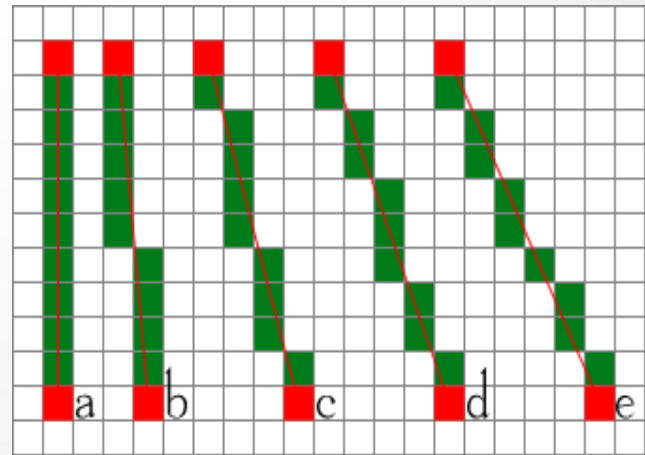
•Biểu diễn không tường minh

$$(y_2 - y_1)x - (x_2 - x_1)y + x_2y_1 - x_1y_2 = 0$$

hay $rx + sy + t = 0$

– $s = -(x_2 - x_1)$

– $r = (y_2 - y_1)$ và $t = x_2y_1 - x_1y_2$



Biểu diễn đoạn thẳng

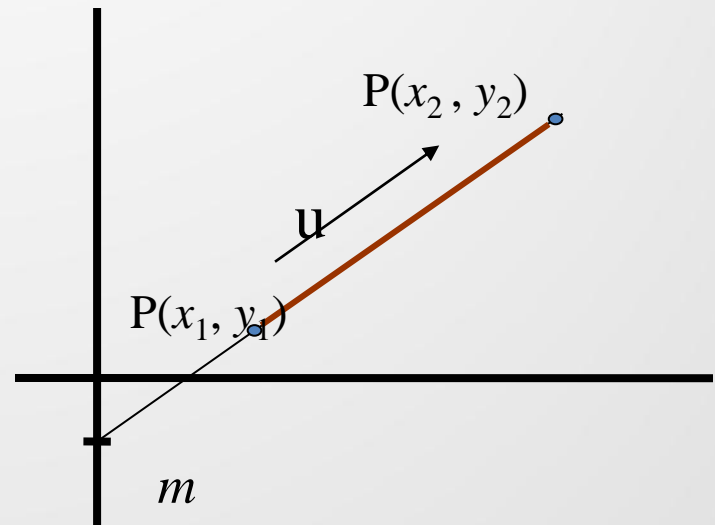
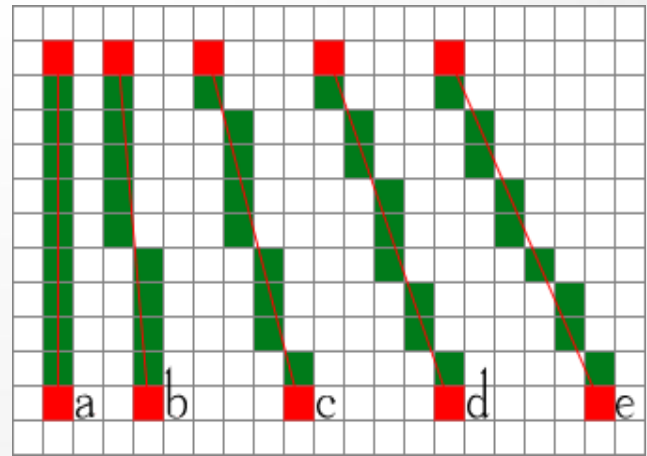
•Biểu diễn tham biến

$$P(u) = P1 + u(P2 - P1)$$

$$u \in [0,1]$$

$$X = x1 + u(x2 - x1)$$

$$Y = y1 + u(y2 - y1)$$



Biểu diễn đoạn thẳng

Giải thuật thông thường

```
DrawLine(int x1,int y1,
int x2,int y2, int color)
{
    float y;
    int x;
    for (x=x1; x<=x2; x++)
    {
        y = y1 + (x-
x1)*(y2-y1)/(x2-x1)
        WritePixel(x,
Round(y), color );
    }
}
```

Giải thuật DDA

Với $0 < k < 1$

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

$$y_{i+1} = y_i + k$$

với $i=1,2,3,\dots$

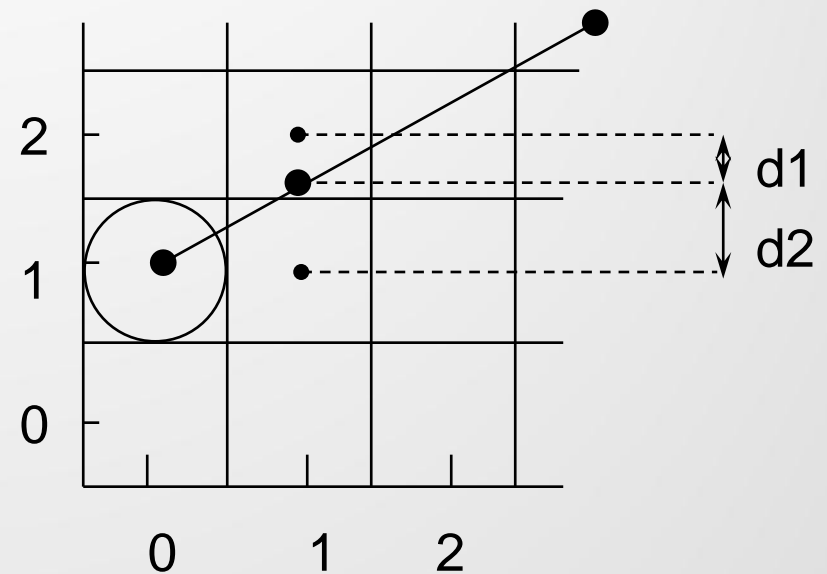
```
Thuật toán ddaline (x1, y1, x2, y2)
x1, y1, x2, y2 : tọa độ 2 điểm đầu cuối
k : hệ số góc
x,y,m : biến

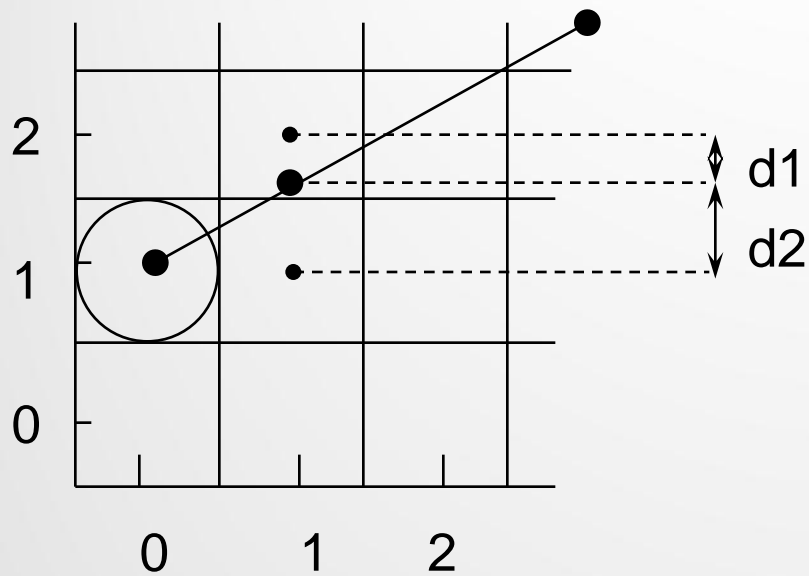
begin
    m =(x2-x1)/(y2-y1);
    x = x1;
    y = y1;
    k = 1/m;
    putpixel(x,y);
while x<x2
    begin
        x = x+1;
        y = y+k;
        putpixel(x,round(y));
    end; end;
```

Thuật toán DDA (Digital Differential Analyzer)

Giải thuật Bresenham

- 1960 Bresenham thuộc IBM
- điểm gần với đường thẳng dựa trên độ phân giai hạn
- loại bỏ được các phép toán chia và phép toán làm tròn như ta đã thấy trong giải thuật DDA
- Xét đoạn thẳng với $0 < k < 1$





$$d_2 = y - y_i = k(x_i + 1) + b - y_i$$

$$d_1 = y_{i+1} - y = y_i + 1 - k(x_i + 1) - b$$

$$\text{if } d_1 \leq d_2 \Rightarrow y_{i+1} = y_i + 1$$

$$\text{else } d_1 > d_2 \Rightarrow y_{i+1} = y_i$$

$$D = d_1 - d_2$$

$$= -2k(x_i + 1) + 2y_i - 2b + 1$$

$$P_i = \Delta x D = \Delta x (d_1 - d_2)$$

Giải thuật Bresenham

$$P_i = -2\Delta y x_i + 2\Delta x y_i + c$$

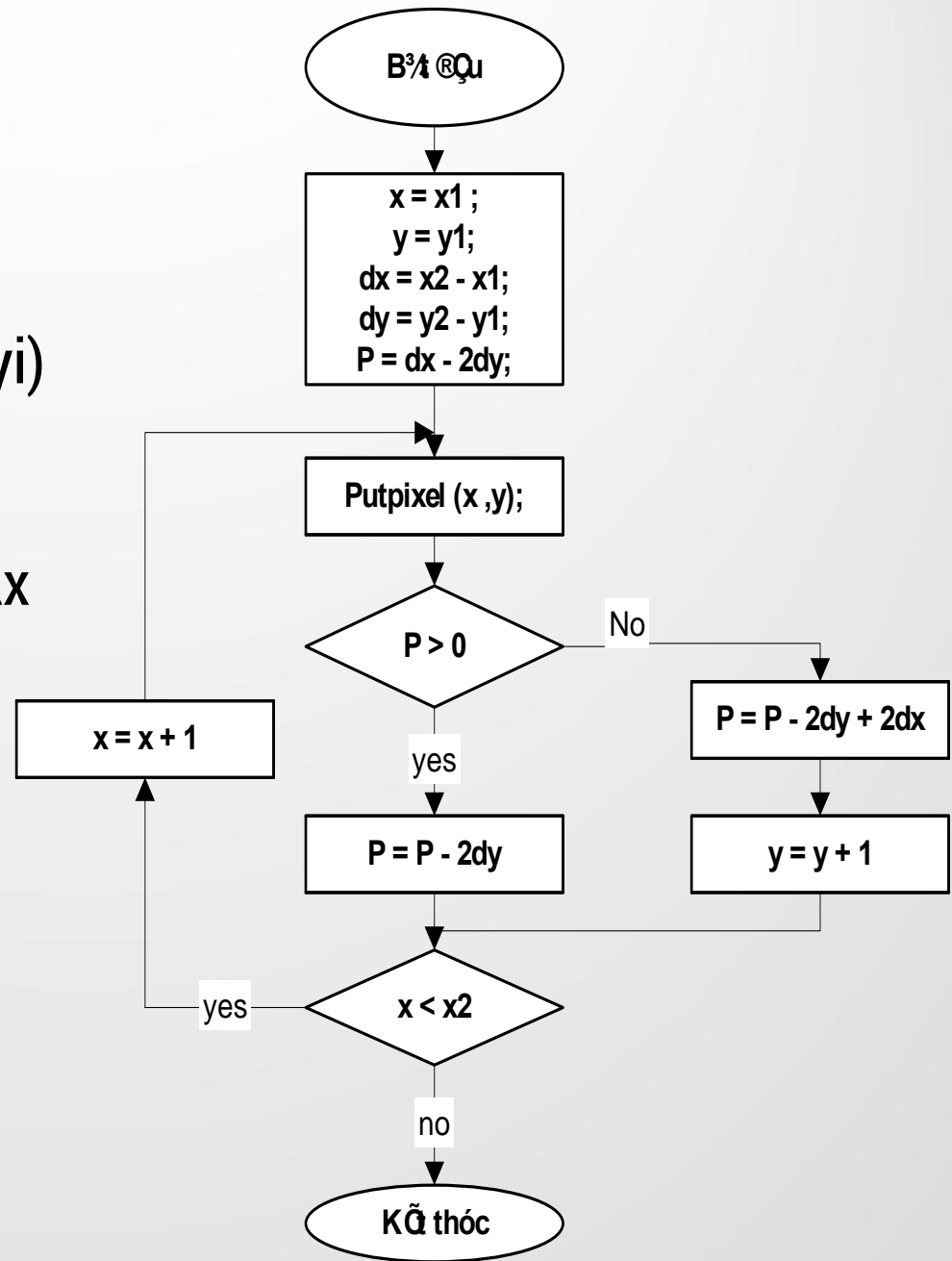
$$P_{i+1} - P_i = -2\Delta y(x_{i+1} - x_i) + 2\Delta x(y_{i+1} - y_i)$$

• Nếu $P_i \leq 0 \Rightarrow y_{i+1} = y_i + 1$
 $P_{i+1} = P_i - 2\Delta y + 2\Delta x$

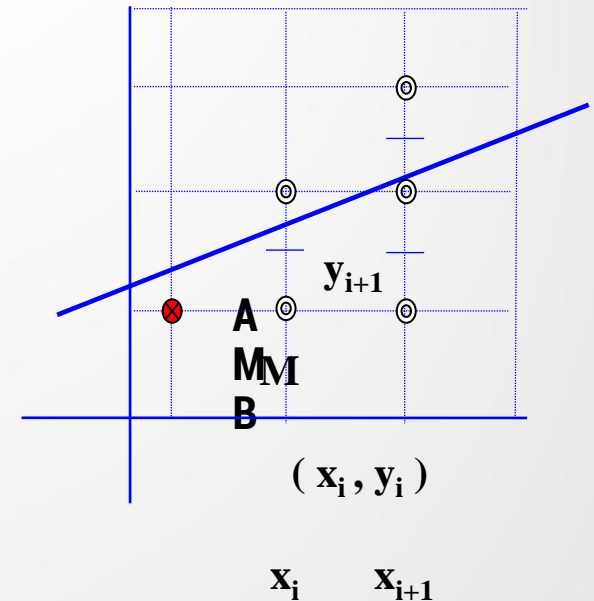
• Nếu $P_i > 0 \Rightarrow y_{i+1} = y_i$
 $P_{i+1} = P_i - 2\Delta y$

$$P_1 = \Delta x(d_1 - d_2)$$

$$P_1 = -2\Delta y + \Delta x$$



- Jack Bresenham 1965 / Pitteway 1967
- VanAken áp dụng cho việc sinh các đường thẳng và đường tròn 1985
- Các công thức đơn giản hơn, tạo được các điểm tương tự như với Bresenham



- $d = F(x_i + 1, y_i + 1/2)$ là trung điểm của đoạn AB
- Việc so sánh, hay kiểm tra M sẽ được thay bằng việc xét giá trị d .
 - Nếu $d > 0$ điểm B được chọn, $y_{i+1} = y_i$
 - nếu $d < 0$ điểm A được chọn. $\Rightarrow y_{i+1} = y_i + 1$
 - Trong trường hợp $d = 0$ chúng ta có thể chọn điểm bất kỳ hoặc A, hoặc B.

Giải thuật trung điểm

Giải thuật trung điểm

- Sử dụng phương pháp biểu diễn không tường minh

$$ax + by + c = 0$$

- Tại mỗi trung điểm của đoạn thẳng giá trị được tính là:

$$ax_i + by_i + c = 0 \Rightarrow (x_i, y_i) \text{ on line}$$

$$ax_i + by_i + c < 0 \Rightarrow (x_i, y_i) \text{ above line}$$

$$ax_i + by_i + c > 0 \Rightarrow (x_i, y_i) \text{ below line}$$

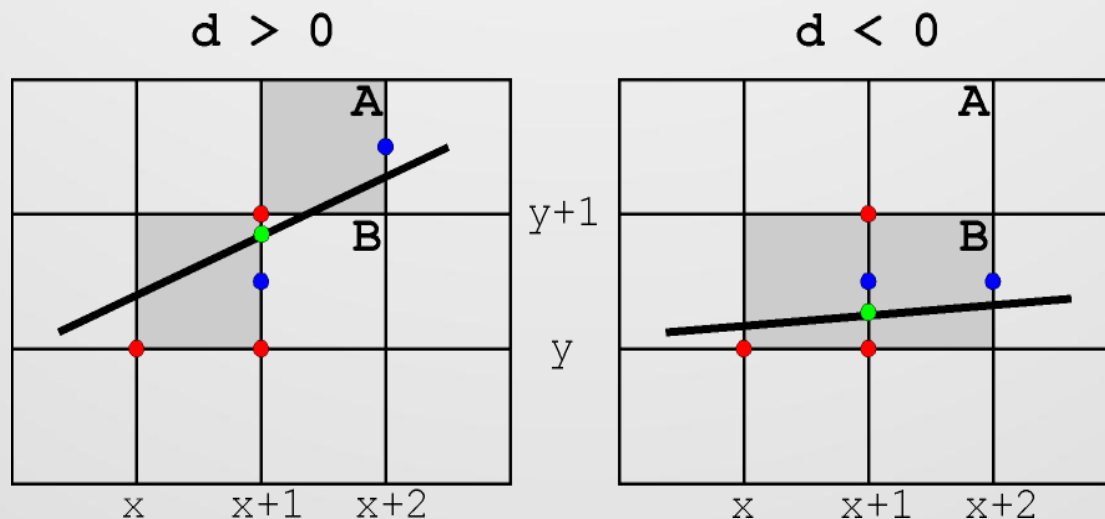
- Chúng ta gọi di là biến quyết định của bước thứ i

$$d_i = a(x_i + 1) + b\left(y_i + \frac{1}{2}\right) + c$$

Giải thuật trung điểm

- If $d_i > 0$ then chọn điểm A \Rightarrow trung điểm tiếp theo sẽ có dạng:

$$\left(x_i + 2, y_i + \frac{3}{2}\right) \Rightarrow d_{i+1} = a(x_i + 2) + b\left(y_i + \frac{3}{2}\right) + c$$
$$= d_i + a + b$$



Giải thuật trung điểm

if $d_i < 0$ then chọn điểm **B** và trung điểm mới là

$$\left(x_i + 2, y_i + \frac{1}{2} \right) \Rightarrow d_{i+1} = a(x_i + 2) + b\left(y_i + \frac{1}{2} \right) + c \\ = d_i + a$$

Ta có:

$$\left. \begin{array}{l} a = \Delta y = y_{end} - y_{start} \\ b = -\Delta x = x_{end} - x_{start} \\ c = C\Delta x \end{array} \right\} \text{where } y = \frac{\Delta y}{\Delta x} x + C = 0 + a + \frac{b}{2}$$

Điểm đầu

$$\left(x_{start} + 1, y_{start} + \frac{1}{2} \right) \Rightarrow d_{start} = a(x_{start} + 1) + b\left(y_{start} + \frac{1}{2} \right) + c \\ = [ax_{start} + by_{start} + c] + a + \frac{b}{2}$$

Giải thuật trung điểm

```
dx = x_end-x_start
dy = y_end-y_start
d = 2*dy-dx
x = x_start
y = y_start
while x < x_end
    if d <= 0 then
        d = d+(2*dy)
        x = x+1
    else
        d = d+2*(dy-dx)
        x = x+1
        y = y+1
    endif
    SetPixel(x,y)
endwhile
```

} initialisation

} choose B

} choose A

Giải thuật trung điểm

- $d = a(x_i + 1) + b(y_i + 1/2) + c$
- Nếu điểm được chọn là B thì M sẽ tăng theo x một đơn vị
 - $d_{i+1} = F(x_i + 2, y_i + 1/2)$
 - $= a(x_i + 2) + b(y_i + 1/2) + c$
 - $d_i = a(x_i + 1) + b(y_i + 1/2) + c$
- Nếu điểm A được chọn thì M tăng theo 2 hướng x và y với cùng một đơn vị.
 - $d_{i+1} = F(x_i + 2, y_i + 3/2)$
 - $= a(x_i + 2) + b(y_i + 3/2) + c$
 - $d_{i+1} = d_i + a + b.$
- Với $a + b = dy - dx.$



2

BIỂU DIỄN ĐƯỜNG TRÒN VÀ ĐƯỜNG Ê-LÍP

AV DOOMG E-EL

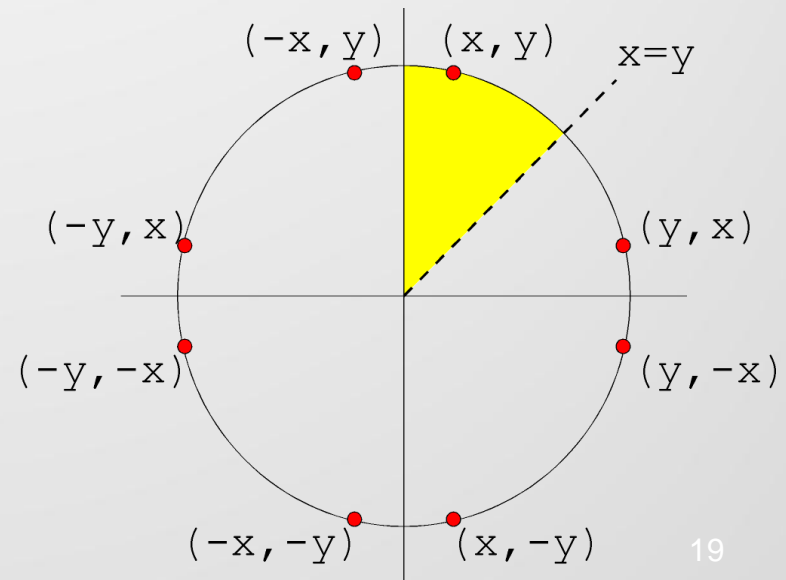
Giải thuật trung điểm cho đường tròn

- Sử dụng phương pháp biểu diễn không tường minh trong giải thuật

$$(x - x_c)^2 + (y - y_c)^2 - r^2 = 0$$

- Thực hiện giải thuật trên 1/8 đường tròn và lấy đối xứng xho các góc còn lại.
- Với d_i là giá trị của đường tròn tại một điểm bất kỳ ta có

$$d_i = \begin{cases} < 0 & \text{if } (x_i, y_i) \text{ is inside circle} \\ = 0 & \text{if } (x_i, y_i) \text{ is on circle} \\ > 0 & \text{if } (x_i, y_i) \text{ is outside circle} \end{cases}$$



Giải thuật trung điểm cho đường tròn

```
d = 1-r  
x = 0  
y = r  
while y < x  
    if d < 0 then  
        d = d+2*x+3  
        x = x+1  
    else  
        d = d+2*(x-y)+5  
        x = x+1  
        y = y-1  
    endif  
    SetPixel(cx+x, cy+y)  
endwhile
```

} initialisation

stop at diagonal \Rightarrow end of octant

} choose **A**

} choose **B**

Translate to the circle center

Giải thuật trung điểm cho đường ê-líp

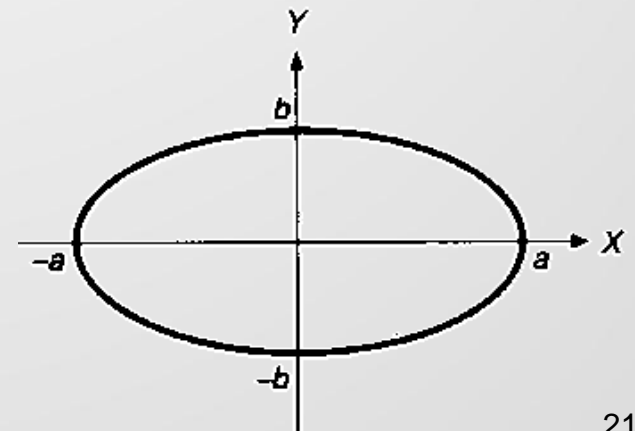
- Tương tự hình tròn
- Áp dụng giải thuật cho $\frac{1}{4}$ đường ê-líp, sau đó lấy đối xứng cho các vị trí còn lại

- Phương trình đường ê-líp

$$F(x, y) = b^2 x^2 + a^2 y^2 - a^2 b^2 = 0$$

– $2a$: đường kính ngang

– $2b$: đường kính dọc



3

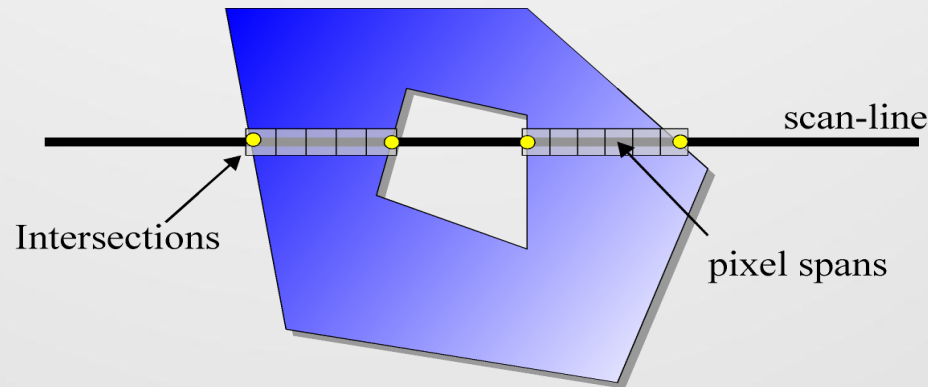
BIỂU DIỄN ĐA GIÁC

Giải thuật đường quét sinh đa giác

- Tồn tại rất nhiều giải thuật sinh đa giác.
- Mỗi giải thuật phục vụ cho 1 loại đa giác nhất định:
- Một số giải thuật chỉ sử dụng được với các tam giác
- Một số giải thuật đòi hỏi đa giác là lồi, không tự cắt chính nó và không có lỗ hổng bên trong

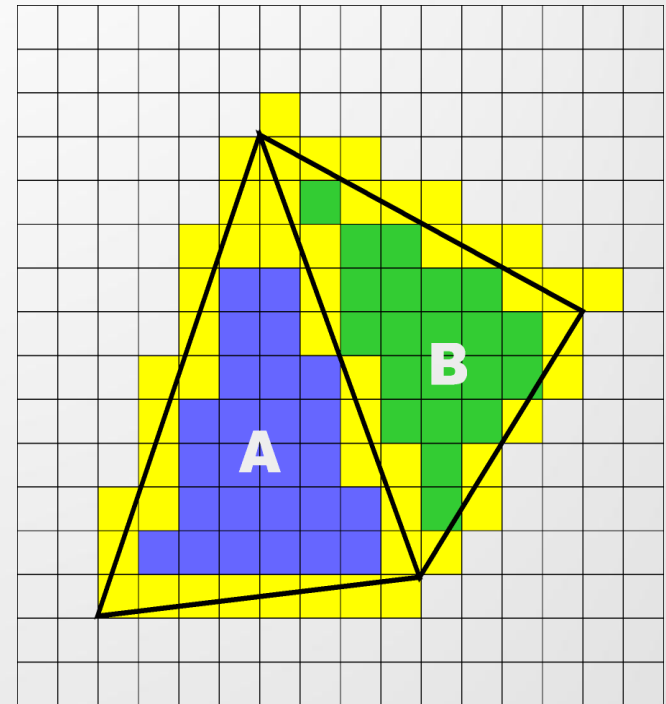
Giải thuật đường quét sinh đa giác




- Polygon scan conversion là giải thuật chung kinh điển cho các loại khác nhau
- Cho mỗi đoạn thẳng quét, chúng ta xác định các cạnh của đa giác cắt đoạn thẳng



Giải thuật đường quét sinh đa giác

- Dùng giải thuật (trung điểm) để xác định các điểm biên cho mỗi đa giác theo thứ tự tăng của x .
- Các điểm phải:
 - Không bị chia sẻ bởi các đa giác lân cận
 - Các đa giác chỉ toàn các điểm cạnh (điểm biên)
- Đảm bảo các đa giác chia sẻ điểm biên mà không chia sẻ các điểm ảnh bên trong của mình.

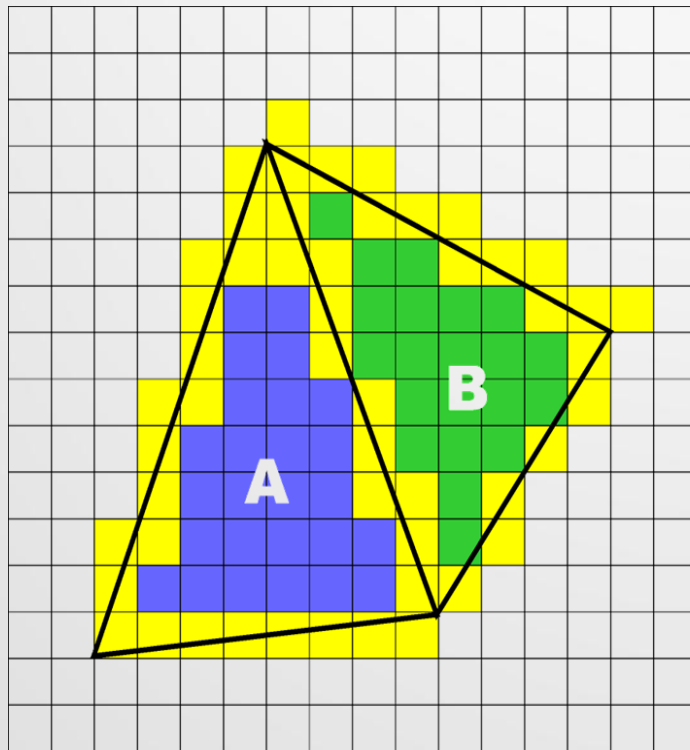


-  polygon A interior
-  polygon B interior
-  edge pixels

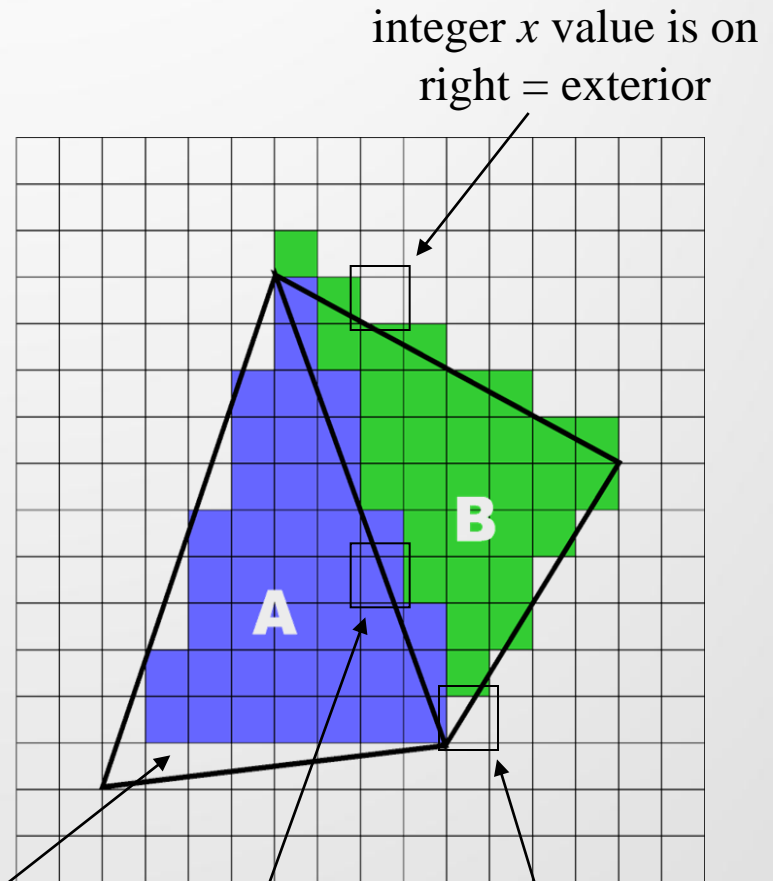
Giải thuật đường quét sinh đa giác

- Thủ tục chung:
 - Xác định giao của đường thẳng quét với cạnh đa giác
 - Sắp xếp các giao điểm theo mức độ tăng dần của x value
 - Điền các điểm ảnh vào giữa cặp các điểm x

Giải thuật đường quét sinh đa giác



horizontal edge
removed



rounded down for A
rounded up for B

y_{\max} not
included

integer x value is on
right = exterior

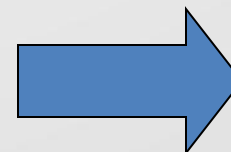
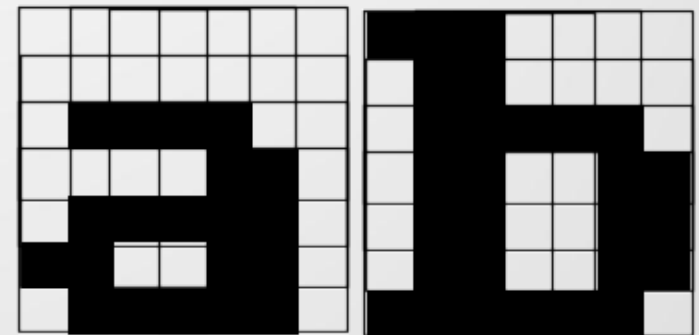


4

BIỂU DIỄN KÝ TỰ

Giải thuật sinh ký tự

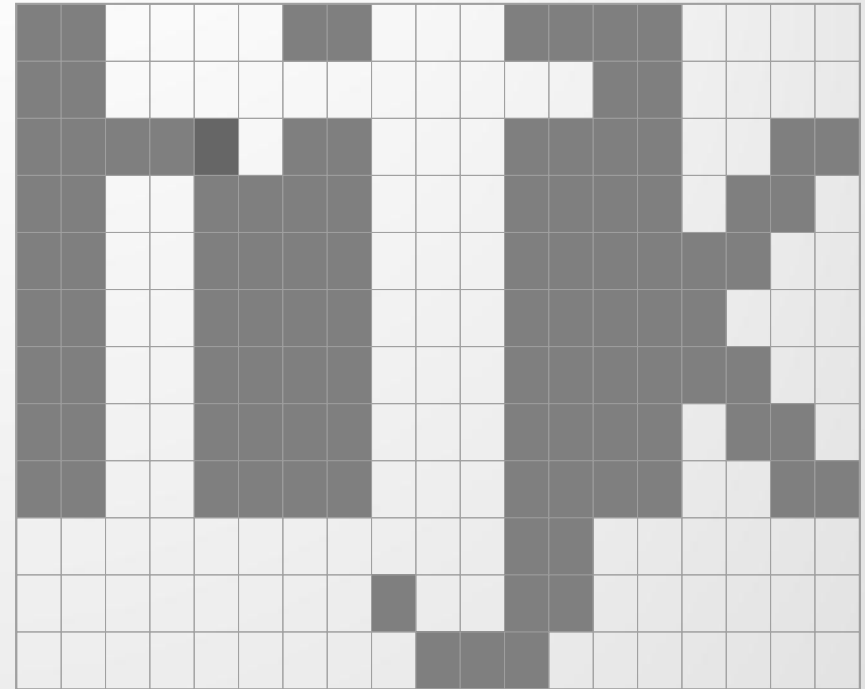
- Trên cơ sở định nghĩa mỗi ký tự với một font chữ cho trước là một bitmap chữ nhật nhỏ
- Font/typeface: set of character shapes
- fontcache
 - các ký tự theo chuỗi liên tiếp nhau trong bộ nhớ
- Dạng cơ bản
 - (thường N, nghiêng I, đậm B, nghiêng đậm B+I)
- Thuộc tính
 - Also colour, size, spacing and orientation



ab

```
Typedef struct
{
int leftx,
int width;
} Char location; //Vị trí
```

```
Typedef struct
{
CacheId;
Heiglit; // Độ rộng chữ
CharSpace; // Khoảng cách
// giữa các ký tự
Charlocation Table [128];
} fontcache
```



Cấu trúc font chữ

Ký tự vector

- Xây dựng theo phương pháp định nghĩa các ký tự bởi đường cong mềm bao ngoài của chúng.
- Tối kém nhất về mặt tính toán
- Chất lượng cao



So sánh

- Đơn giản công việc sinh ký tự (copypixel)
- Lưu trữ lớn
- Các phép biến đổi (I,B, scale) đòi hỏi lưu trữ thêm
- Kích thước không đổi
- Phức tạp (Tính toán phương trình)
- Lưu trữ gọn nhẹ
- Các phép biến đổi dựa vào các công thức biến đổi
- Kích thước phụ thuộc vào môi trường (không có kích thước cố định)