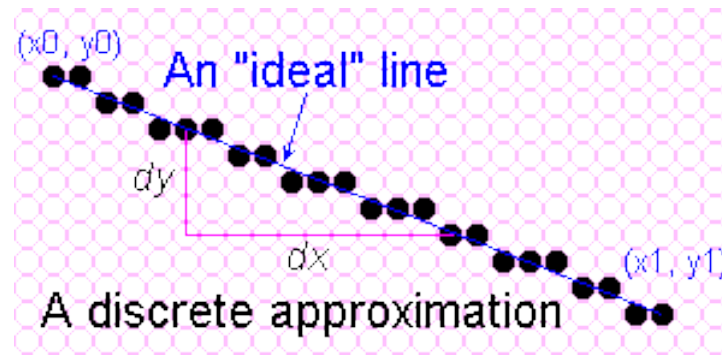


Đồ họa máy tính

Vẽ đường thẳng và đường tròn

Hướng tới một đường thẳng lý tưởng

- Chúng ta chỉ có thể vẽ xấp xỉ đường thẳng một cách rời rạc



- Chiếu sáng các điểm gần nhất với đường thẳng thực tế trong trường hợp chỉ có hai cách thể hiện một điểm:
 - Điểm được thắp sáng hoặc không thắp sáng

Thế nào là một đường thẳng lý tưởng

- Trông phải thẳng và liên tục
 - Trong máy tính chỉ có thể được như vậy với các đường thẳng song song với trục tọa độ hoặc có góc 45° với trục tọa độ
- Phải đi qua hai điểm đầu và cuối
- Phải có mật độ và cường độ sáng đều
 - Đều trên một đường thẳng và đều trên tất cả các đường thẳng
- Thuật toán vẽ phải hiệu quả và có thể thực hiện nhanh

Đường thẳng đơn giản

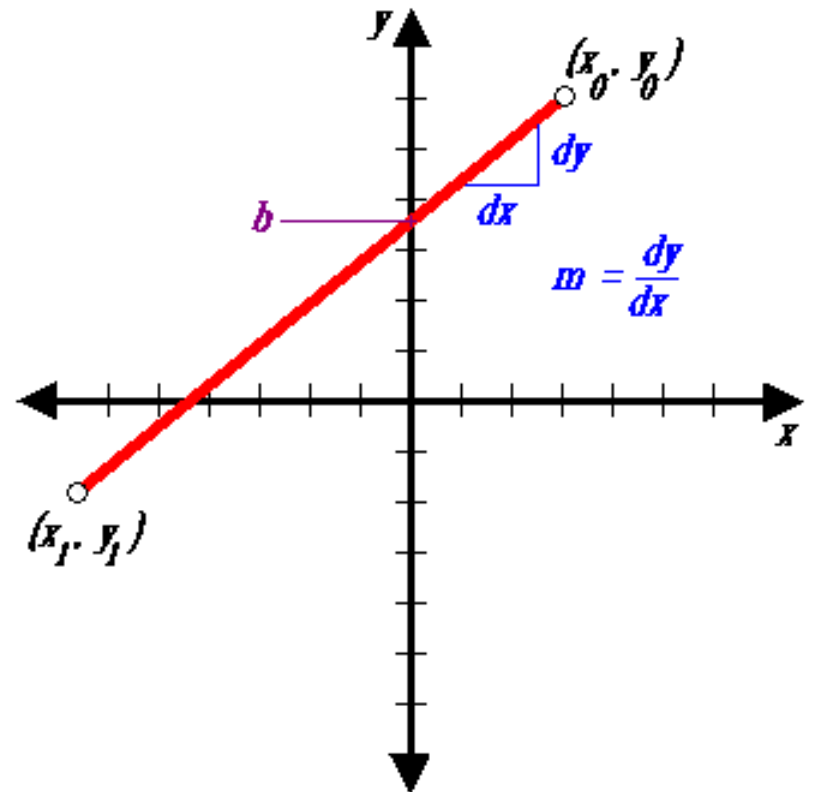
Dựa trên phương trình đường thẳng:

$$y = mx + b$$

Cách tiếp cận đơn giản:

tăng x , rồi tìm ra y

Cần các tính toán số thực

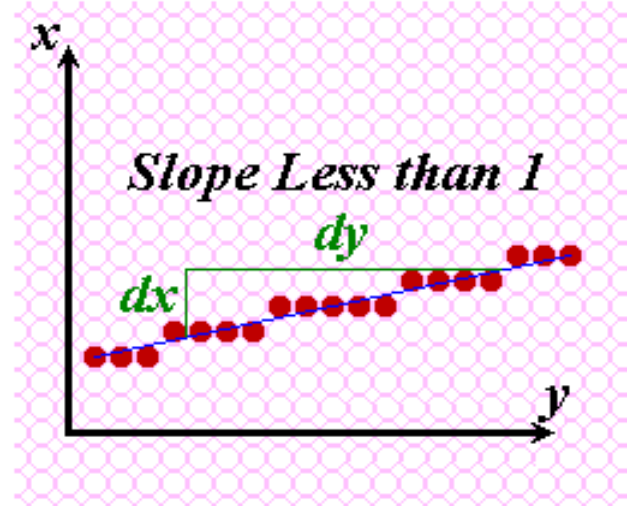
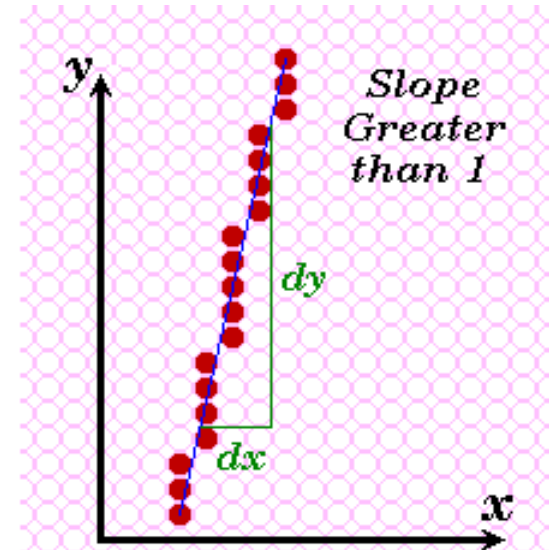


Thuật toán đó có tốt không?

Thuật toán có vẻ ổn với những đường thẳng có hệ số góc nghiêng (slope) bằng 1 hoặc nhỏ hơn,

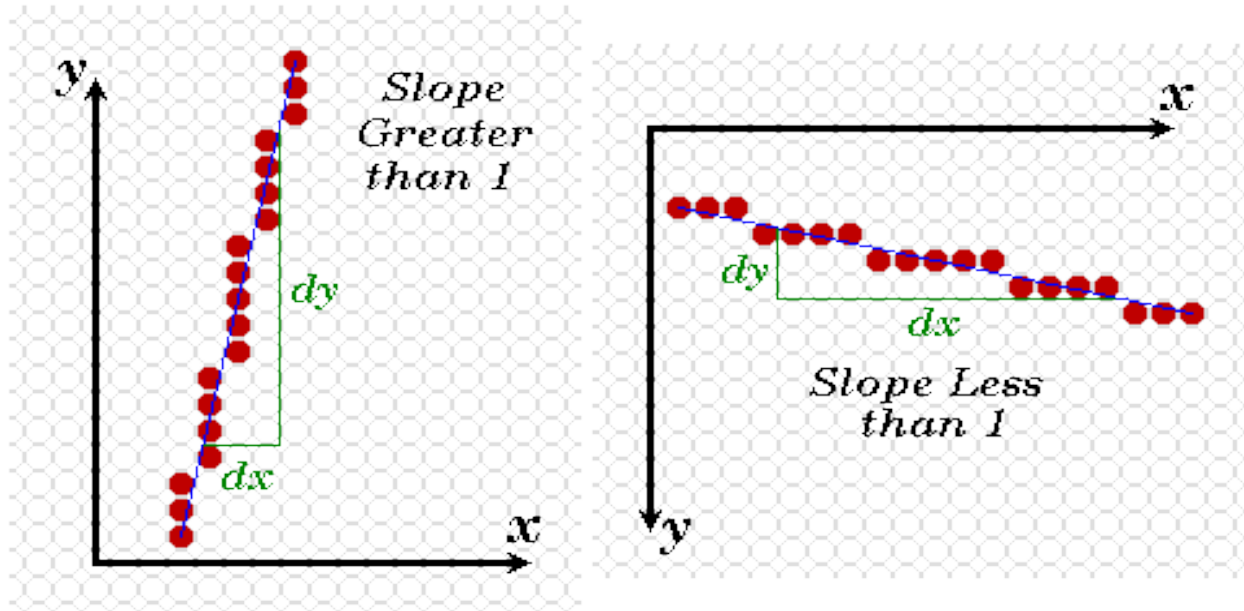
tuy nhiên, nó không tốt cho những đường thẳng với hệ số góc nghiêng lớn hơn 1 – các đường thẳng trông rời rạc – phải thêm các điểm vào các cột thì trông mới ổn.

Giải pháp? - sử dụng phương pháp đối xứng.



Thay đổi thuật toán cho từng góc phần tám (45°) của hệ tọa độ

Rotate and Rename coordinate axes



Có thể thay đổi tên của trục tọa độ, HOẶC, tăng theo trục x nếu $dy < dx$, nếu không thì tăng theo trục y

Thuật toán DDA

- DDA = Digital Differential Analyser
(Phân tích vi phân số hóa)
- Xét đường thẳng theo phương trình tham số theo t :

$$(x_1, y_1)$$

$$(x_2, y_2)$$

$$x(t) = x_1 + t(x_2 - x_1)$$

$$y(t) = y_1 + t(y_2 - y_1)$$

Thuật toán DDA

$$x(t) = x_1 + t(x_2 - x_1)$$
$$y(t) = y_1 + t(y_2 - y_1)$$

- Bắt đầu với $t = 0$
- Tại mỗi bước, tăng t một lượng dt
- Chọn giá trị thích hợp cho dt
- Sao cho không bỏ mất điểm nào:
 - Nghĩa là: $\frac{dx}{dt} < 1$ và $\frac{dy}{dt} < 1$
- Chọn dt là giá trị max của dx và dy

$$x_{moi} = x_{cu} + \frac{dx}{dt}$$
$$y_{moi} = y_{cu} + \frac{dy}{dt}$$

Thuật toán DDA

```
line(int x1, int y1, int x2, int y2)
```

```
{  
float x,y;  
int dx = x2-x1, dy = y2-y1;  
int n = max(abs(dx),abs(dy));  
float dt = n, dxdt = dx/dt, dydt = dy/dt;  
    x = x1;  
    y = y1;  
    while( n-- ) {  
        point(round(x),round(y));  
        x += dxdt;  
        y += dydt;  
    }  
}
```

n - range of t.

Thuật toán DDA

- Vẫn còn sử dụng rất nhiều phép toán số thực.
 - 2 phép làm tròn và hai phép cộng số thực.
- Liệu có cách nào đơn giản hơn không?
- Có cách nào mà chúng ta chỉ cần dùng các phép toán số nguyên?
 - Như vậy sẽ có thể cài đặt dễ dàng trên máy tính hiện thời và có thể chạy rất nhanh.

Thuật toán Bresenham

- Lưu ý trong thuật toán DDA, x hoặc y luôn tăng lên 1
- Giả sử x luôn tăng lên 1, cần tính y cho hiệu quả

Thuật toán Bresenham (...)

- Giả thiết đường thẳng chúng ta cần vẽ là từ $(0,0)$ đến (a,b) , với a và b là 2 số nguyên, và $0 \leq b \leq a$ (vì (a,b) ở góc phần tám thứ nhất)

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

$$y_i = y_{i-1} + b/a = i*b/a$$

Cần tính y_i và sau đó làm tròn đến số nguyên gần nhất

Thuật toán Bresenham (...)

- Giá trị của tọa độ y bắt đầu từ 0. Tại điểm nào, y sẽ bắt đầu bằng 1?
- Để trả lời câu hỏi này, chúng ta phải tính b/a , $2b/a$, $3b/a$, ..., và xem tại điểm nào các giá trị này bắt đầu lớn hơn $1/2$
- Sau đó, giá trị của y sẽ giữ bằng 1 cho đến khi lớn hơn $3/2$
- Như vậy chúng ta phải so sánh b/a , $2b/a$, $3b/a$... với các số $1/2$, $3/2$, $5/2$, ...

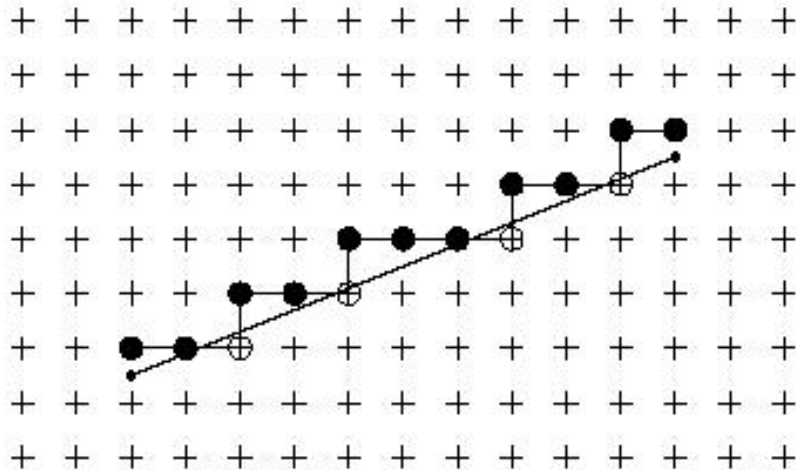
Thuật toán Bresenham (...)

- Tránh làm các phép tính số thực => thay bằng các phép so sánh $2b$, $4b$, $6b$, ... với a , $3a$, $5a$, ..
- Việc so sánh một số với 0 nhanh hơn việc so sánh 2 số với nhau, do đó chúng ta sẽ bắt đầu với việc xét khi nào thì $2b-a$, $4b-a$, ... bắt đầu lớn hơn 0
- Ban đầu, đặt biến quyết định $d = 2b - a$, mỗi lần cần cộng thêm $2b$ vào d
- Đến khi d bắt đầu lớn hơn 0, trừ thêm $2a$ vào d

Thuật toán Bresenham (...)

```
begin
  integer d, x, y;
  d := 2*b - a;
  x := 0;
  y := 0;
  while true do
    begin
      Draw (x,y);
      if x = a then Exit;
      if d ≥ 0 then
        begin
          y := y + 1;
          d := d - 2*a;
        end;
      x := x + 1;
      d := d + 2*b;
    end
  end
end
```

Quan sát các đường thẳng



```
while( n-- )  
{  
  draw(x,y);  
  move right;  
  if( below line )  
    move up;  
}
```


Kiểm tra một điểm nằm ở phía nào của đường thẳng

- Để cài đặt được thuật toán mới cần kiểm tra xem một điểm nằm ở phía nào của đường thẳng.
- Viết phương trình đường thẳng:

$$F(x, y) = ax + by + c = 0$$

- Dễ nhận thấy nếu $F < 0$, điểm đó nằm trên đường thẳng, nếu $F > 0$ điểm đó nằm dưới đường thẳng.

Kiểm tra một điểm nằm ở phía nào của đường thẳng

$$F(x, y) = ax + by + c = 0$$

- Cần phải tìm các hệ số a, b, c.
- Xét dạng khác của phương trình đường thẳng:

$$y = mx + b \text{ do đó } y = \frac{dy}{dx}x + b$$

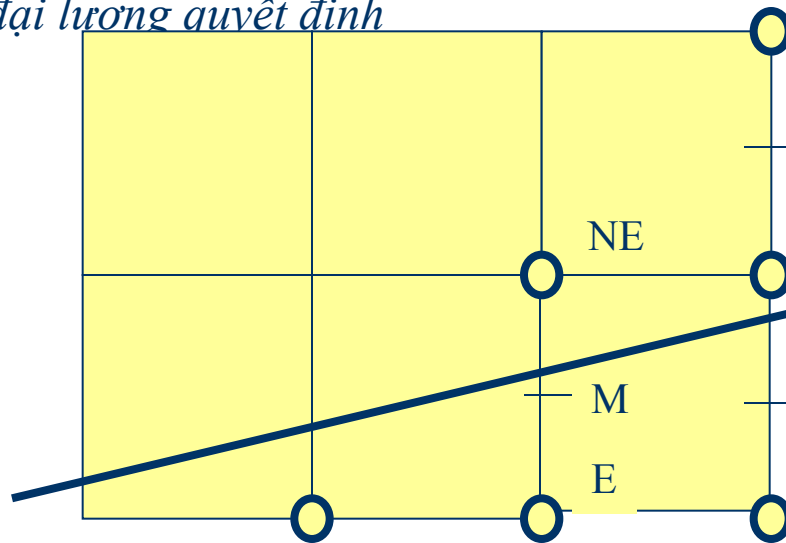
- Do đó

$$F(x, y) = dy \cdot x - dx \cdot y + c = 0$$

Đại lượng quyết định

Tính F tại điểm M

Coi đây là *đại lượng quyết định*



$$d = F(x_p + 1, y_p + \frac{1}{2})$$

Điểm trước
(x_p, y_p)

Các phương án
cho điểm
hiện tại

Các phương án cho điểm tiếp theo

Đại lượng quyết định

Tính d cho điểm tiếp theo, Quyết định xem điểm E và NE sẽ được chọn :
Nếu điểm E được chọn :

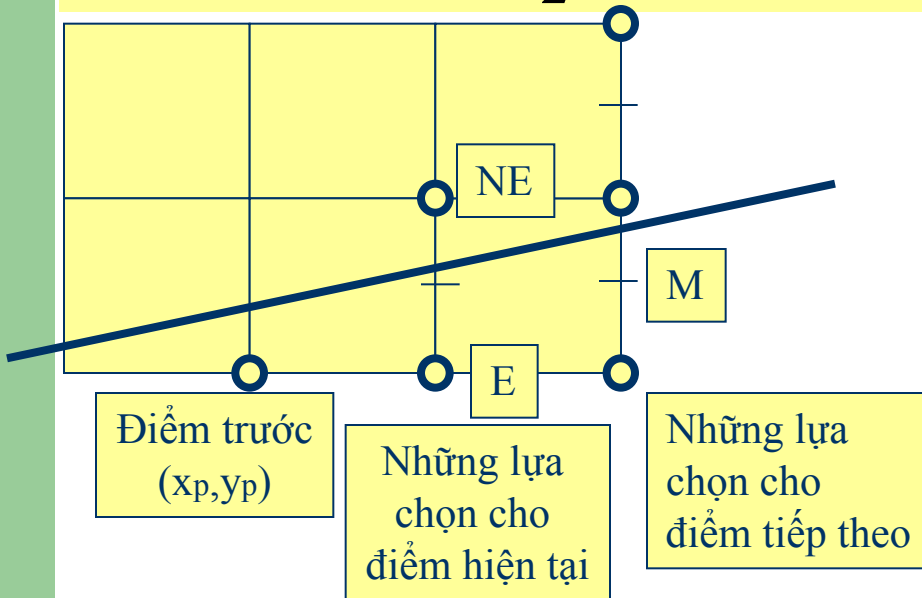
$$d_{moi} = F(x_p + 2, y_p + \frac{1}{2}) = a(x_p + 2) + b(y_p + \frac{1}{2}) + c$$

Xem lại :

$$\begin{aligned} d_{cu} &= F(x_p + 1, y_p + \frac{1}{2}) \\ &= a(x_p + 1) + b(y_p + \frac{1}{2}) + c \end{aligned}$$

Do đó :

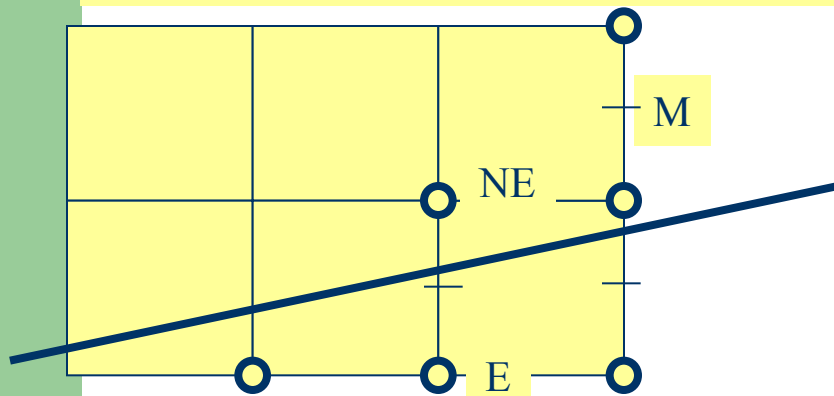
$$\begin{aligned} d_{moi} &= d_{cu} + a \\ &= d_{cu} + dy \end{aligned}$$



Đại lượng quyết định

Nếu điểm NE được chọn :

$$d_{moi} = F\left(x_p + 2, y_p + \frac{3}{2}\right) = a\left(x_p + 2\right) + b\left(y_p + \frac{3}{2}\right) + c$$



Do đó:

$$\begin{aligned}d_{moi} &= d_{cu} + a + b \\ &= d_{moi} + dy - dx\end{aligned}$$

Điểm trước
(x_p, y_p)

Những lựa
chọn cho
điểm hiện tại

Những lựa
chọn cho
điểm tiếp theo

Tóm tắt thuật toán điểm giữa (mid-point algorithm)

- Chọn một trong hai điểm tiếp theo dựa trên dấu của đại lượng quyết định.
- Điểm bắt đầu là (x_1, y_1) .
- Cần phải tính giá trị ban đầu của đại lượng quyết định d .

Giá trị ban đầu của d

Điểm bắt đầu (x_1, y_1)

$$\begin{aligned}d_{batdau} &= F\left(x_1 + 1, y_1 + \frac{1}{2}\right) = a(x_1 + 1) + b\left(y_1 + \frac{1}{2}\right) + c \\ &= ax_1 + by_1 + c + a + \frac{b}{2} \\ &= F(x_1, y_1) + a + \frac{b}{2}\end{aligned}$$

Tuy nhiên (x_1, y_1) là một điểm trên đường thẳng, do đó $F(x_1, y_1) = 0$

$$d_{batdau} = dy - dx / 2$$

Nhân đại lượng này với 2 để loại bỏ mẫu số \Rightarrow không ảnh hưởng đến dấu.

Thuật toán điểm giữa

```
void MidpointLine(int
    x1, y1, x2, y2)
{
    int dx=x2-x1;
    int dy=y2-y1;
    int d=2*dy-dx;
    int increE=2*dy;
    int incrNE=2*(dy-dx);
    x=x1;
    y=y1;
    WritePixel(x, y);
```

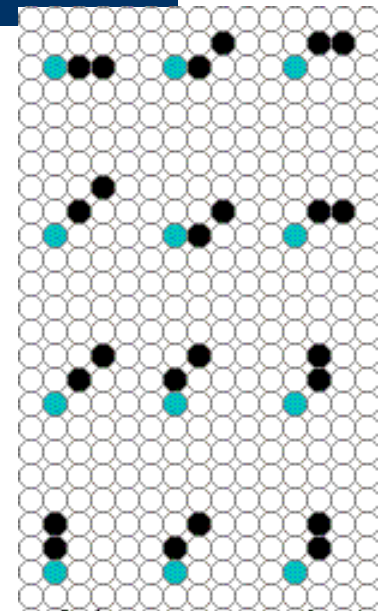
```
    while (x < x2) {
        if (d<= 0) {
            d+=incrE;
            x++;
        } else {
            d+=incrNE;
            x++;
            y++;
        }
        WritePixel(x, y);
    }
}
```


Thuật toán điểm giữa chưa phải là cuối cùng!

Thuật toán 2 bước (2-step algorithm) bởi Xiaolin Wu:

Coi việc vẽ đường thẳng như một au-tô-mát, xét hai điểm tiếp theo của một đường thẳng, dễ dàng thấy chỉ có một lượng hữu hạn các khả năng.

Thuật toán này còn tận dụng sự đối xứng của đường thẳng bằng cách vẽ đường thẳng cùng một lúc từ hai phía đến điểm giữa (giảm gần một nửa sự tính toán)



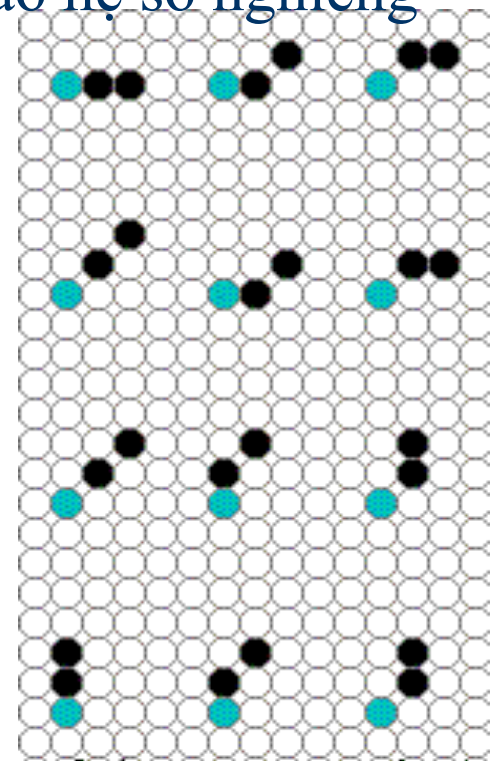
Thuật toán hai bước

Các vị trí tiếp theo của hai điem phụ thuộc vào hệ số nghiêng của đường thẳng: Hệ số nghiêng từ 0 đến $\frac{1}{2}$

Hệ số nghiêng từ $\frac{1}{2}$ đến 1

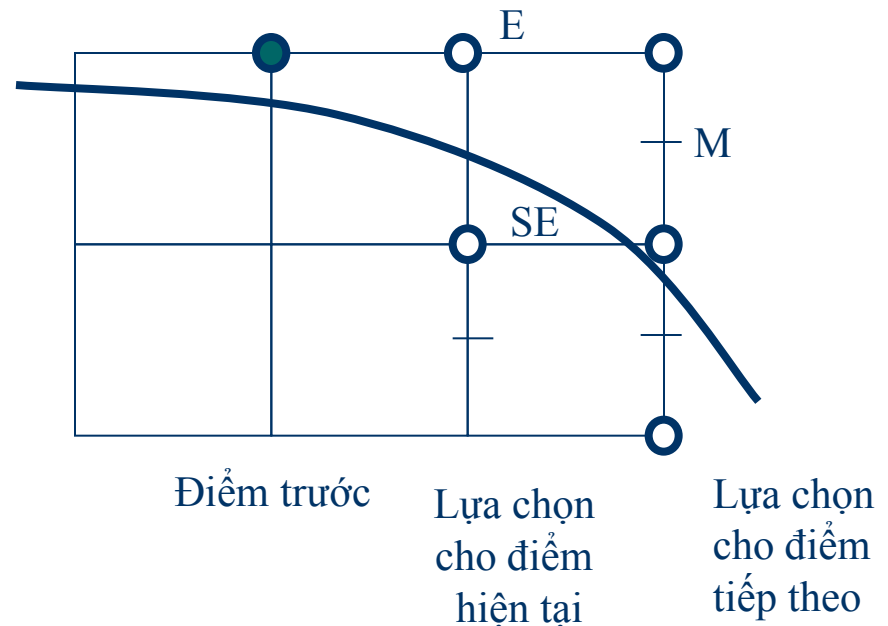
Hệ số nghiêng từ 1 và 2

Hệ số nghiêng lớn hơn 2



Vẽ đường tròn

- Cũng có thể dùng thuật toán điểm giữa để vẽ đường tròn.



Vẽ đường tròn

- Phương trình đường thẳng đường tròn:

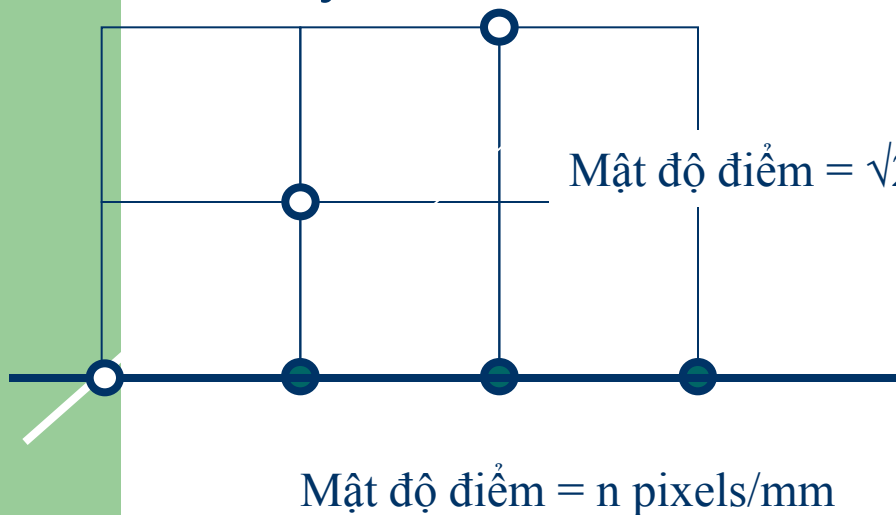
$$f(x, y) = (x - x_c)^2 + (y - y_c)^2 - r^2$$

Neu SE đưoc chon $d_{moi} = d_{cu} + (2x_p - 2y_p + 5)$

Neu E đưoc chon $d_{moi} = d_{cu} + (2x_p + 3)$

Những vấn đề với thuật toán Bresenham và thuật toán điểm giữa

- Các điểm được vẽ trên một đường thẳng đơn \Rightarrow khi thay đổi góc, độ sáng của đường thẳng thay đổi.



Có thể vẽ bằng những màu khác nhau khi thay đổi góc

Tóm tắt về vẽ đường thẳng

- Sử dụng dạng “hiện” (explicit) của đường thẳng
 - Không hiệu quả, khó kiểm soát.
- Sử dụng dạng tham số của đường thẳng.
 - Thể hiện đường thẳng dưới dạng tham số t
 - Tham số DDA
 - Thuật toán Bresenham
- Sử dụng dạng ẩn (implicit) của đường thẳng
 - Chỉ cần kiểm tra điểm nằm ở bên nào của đường thẳng.
 - Thuật toán điểm giữa.
 - Cũng có thể dùng để vẽ đường tròn.

Tóm tắt về vẽ đường thẳng

- Cài đặt các thuật toán vẽ đường thẳng.
 - Thể hiện đường thẳng dưới dạng tham số t
 - Tham số DDA
 - Thuật toán Bresenham
- Cài đặt thuật toán điểm giữa vẽ đường tròn