CÁC GIẢI THUẬT SINH THỰC THỂ CƠ SỞ

BUỔI GIẢNG THỨ MÔN HỌC KỸ THUẬT ĐỒ HỌA

SỐ TÍN CHỈ: 03 LỚP: 59CNTT THỜI LƯỢNG: 45 PHÚT

TRÌNH BÀY: ĐOÀN VŨ THỊNH

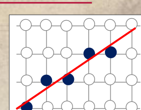
BỘ MÔN: KỸ THUẬT PHẦN MỀM

2.1. GIỚI THIỆU

- Bất kì một ảnh mô tả thế giới thực nào bao giờ cũng được cấu trúc từ tập các đối tượng đơn giản hơn. Ví dụ một ảnh thể hiện bài trí của một căn phòng sẽ được cấu trúc từ các đối tượng như cây cảnh, tủ kính, bàn ghế, tường, ánh sáng đèn
- Với các ảnh đồ họa phát sinh bằng máy tính, hình dạng và màu sắc của mỗi đối tượng có thể được mô tả riêng biệt bằng hai cách:
  - Hoặc là bằng dãy các pixel tương ứng
  - Hoặc là bằng tập các đối tượng hình học cơ sở như đoạn thẳng hay đa giác.
- Sau đó, các ảnh hiển thị bằng cách nạp các pixel vào vùng bộ nhớ màn hình.

2.1. GIỚI THIỆU (tt)

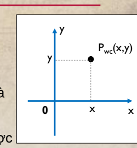
- Với các ảnh được mô tả bằng các đối tượng hình học cơ sở, cần có một quá trình chuyển các đối tượng này về dạng ma trận các pixel trước. Quá trình này còn được gọi là quá trình chuyển đổi bằng dòng quét.



- Đối tượng đồ họa cơ sở đơn giản nhất là điểm và đoạn thẳng, ngoài ra còn có đường tròn, và các đường conics, mặt bậc hai, các mặt và đường splines, các vùng tô đa giác, chuỗi kí tự…

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ

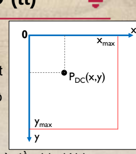
- **2.2.1. Hệ đồ hoạ thế giới thực và hệ đồ hoạ thiết bị**
- Hệ tọa độ thế giới thực (hay hệ tọa độ thực) là hệ tọa độ được dùng mô tả các đối tượng thế giới thực.
- Một trong các hệ tọa độ thực thường được dùng nhất đó là hệ tọa độ Descartes (x, y ∈ R)
- Gốc tọa độ là điểm O có tọa độ (0, 0). Ox, Oy lần lượt được gọi là trục hoành, trục tung;



- x là khoảng cách từ điểm đến trục hoành hay còn được gọi là hoành độ, y là khoảng cách từ điểm đến trục tung hay còn được gọi là tung độ.

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.1. Hệ đồ hoạ thế giới thực và hệ đồ hoạ thiết bị (tt)**
- Các điểm trong hệ tọa độ thiết bị cũng được mô tả bởi một cặp tọa độ (x, y ∈ N. Do đó, các điểm trong các hệ tọa độ thiết bị là rời rạc do tính chất của tập các số tự nhiên.



- Các tọa độ x, y của hệ tọa độ thiết bị không thể lớn tùy ý mà đều bị giới hạn trong một khoảng nào đó. Khoảng giới hạn các tọa độ x, y là khác nhau đối với từng loại thiết bị khác nhau.

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.2. Biểu diễn điểm và đoạn thẳng**
- **Điểm**
  - Điểm là thành phần cơ sở được định nghĩa trong một hệ tọa độ. Đối với hệ tọa độ hai chiều mỗi điểm được xác định bởi cặp tọa độ (x, y). Ngoài thông tin về tọa độ, điểm còn có thuộc tính là màu sắc.

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.2. Biểu diễn điểm và đoạn thẳng (tt)**
- **Đoạn thẳng**

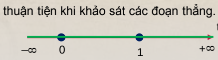
  Một đường thẳng có thể xác định nếu biết hai điểm thuộc nó. Phương trình đường thẳng đi qua hai điểm (x_1, y_1) và (x_2, y_2) có dạng sau: $\frac{x-x_1}{y-y_1} = \frac{x_2-x_1}{y_2-y_1}$
- Hay ở dạng tương đương: $(x - x_1)(y_2 - y_1) = (y - y_1)(x_2 - x_1)$
- Khai triển ta có dạng: $y = mx + b$, trong đó:
  - $m = \frac{D_y}{D_x}$ $Dy=(y_2-y_1)$ $Dx=(x_2-x_1)$ $b=y_1-mx_1$
- Đây còn được gọi là phương trình đoạn thẳng của đường thẳng.

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.2. Biểu diễn điểm và đoạn thẳng (tt)**
- **Đoạn thẳng (tt)**

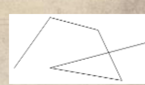
  Nếu khai triển dưới dạng: $(y_2 - y_1)x - (x_2 - x_1)y - x_1y_2 + x_2y_1 = 0$

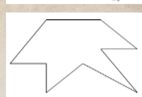
  và đặt $A= y_2 - y_1$, $B = -(x_2 - x_1)$, $C = x_2y_1 - x_1y_2$ thì phương trình đường thẳng sẽ có dạng $Ax + By + C = 0$, dạng này được gọi là phương trình tổng quát của đường thẳng.
- Phương trình tham số của đường thẳng có dạng các tọa độ x, y được mô tả qua một thành phần thứ ba là t. Dạng này rất thuận tiện khi khảo sát các đoạn thẳng.
- $\begin{cases} x = (1-t)x_1 + tx_2 \\ y = (1-t)y + ty_2 \end{cases}$



2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.2. Biểu diễn điểm và đoạn thẳng (tt)**
- **Đường gấp khúc**
- Là tập các đoạn thẳng nối với nhau một cách tuần tự. Điểm giao của hai đoạn thẳng được gọi là đỉnh. Các đường gấp khúc được xác định qua danh sách các đỉnh, mỗi đỉnh được cho bởi các cặp tọa độ (x_i, y_i).
- Một đa giác là một đường gấp khúc có điểm đầu và điểm cuối trùng nhau





2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.2. Biểu diễn điểm và đoạn thẳng (tt)**
- **Các thuộc tính của đoạn thẳng**
  - Màu sắc
  - Độ rộng của nét vẽ
  - Kiểu nét vẽ của đoạn thẳng
- Đối với đường gấp khúc, các đoạn thẳng trong cùng một đường gấp khúc thì có cùng một thuộc tính.

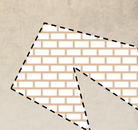
2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.2. Biểu diễn điểm và đoạn thẳng (tt)**
- **Vùng tô**

  Một vùng tô bao gồm đường biên và vùng bên trong. Đường biên là một đường khép kín ví dụ như đa giác.
- **Các thuộc tính của vùng tô bao gồm:**

  Thuộc tính của đường biên: chính là các thuộc tính như thuộc tính của đoạn thẳng.

  Thuộc tính của vùng bên trong: gồm màu tô và mẫu tô.



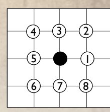
2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật vẽ đoạn thẳng thông thường**

  Giả sử tọa độ các điểm nguyên sau khi xấp xỉ đối tượng thực lần lượt là (x_i, y_i), $i = 0,1,2,....$ Đây là các điểm nguyên sẽ được hiển thị trên màn hình.
- Bài toán đặt ra là nếu biết được (x_i, y_i) là tọa độ nguyên xác định ở bước thứ i, điểm nguyên tiếp theo (x_{i+1}, y_{i+1}) sẽ được xác định như thế nào.

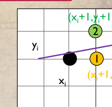
2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật vẽ đoạn thẳng thông thường**
- Nhận xét rằng để đối tượng hiển thị trên lưới nguyên được liền nét, các điểm mà (x_{i+1},y_{i+1}) có thể chọn chỉ là một trong tám điểm được đánh số từ 1 đến 8 (điểm đen chính là (x_i,y_i)). Hay nói cách khác: (x_{i+1},y_{i+1}) = (x_i±1,y_i±1).
- Dáng điệu của đường sẽ cho ta gợi ý khi chọn một trong tám điểm trên. Cách chọn các điểm như thế nào sẽ tùy thuộc vào từng thuật toán trên cơ sở xem xét tới vấn đề tối ưu tốc độ.



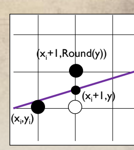
2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật vẽ đoạn thẳng thông thường**
- Xét đoạn thẳng có hệ số góc $0 < m < 1$ và $Dx > 0$
- Với các đoạn thẳng dạng này, nếu (x_i, y_i) là điểm đã xác định được ở bước thứ i (điểm màu đen) thì điểm cần chọn (x_{i+1},y_{i+1}) ở bước thứ (i+1) sẽ là một trong hai trường hợp như hình bên:
- Như vậy: $\begin{cases} x_{i+1} = x_i + 1 \\ y_{i+1} \epsilon \{y_i, y_{i+1}\} \end{cases}$



2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở**
- **Thuật toán DDA (Digital Differential Analizer)**
- Với thuật toán DDA, việc quyết định chọn y_{i+1} là y_i hay y_i+1, dựa vào phương trình của đoạn thẳng *y* = *mx*+*b*.
- Tọa độ của điểm (x_i+1,*y*) thuộc về đoạn thẳng thực.
- y_i+1 sẽ là giá trị sau khi làm tròn giá trị tung độ *y*.



$$\begin{cases} y = m(x_i + 1) + b \\ y_{i+1} = Round(y) \end{cases}$$

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Thuật toán DDA (Digital Differential Analizer) (tt)**
- Nếu tính trực tiếp giá trị thực y ở mỗi bước từ phương trình *y* = *mx* + *b* thì phải cần một phép toán nhân và một phép toán cộng số thực.
- Để cải thiện tốc độ, người ta tính giá trị thực của y ở mỗi bước theo cách sau để khử phép tính nhân trên số thực:
- Nhận xét rằng: $y_{sau} = mx_{i+1} + b = m(x_i + 1) + b$

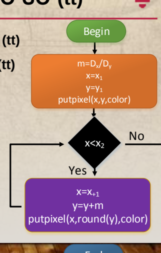
  $y_{trước} = mx_i + b$

  $\Rightarrow y_{sau} = y_{trước} + m$

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Thuật toán DDA (Digital Differential Analizer) (tt)**

```
#define Round(a) int(a+0.5)
int Color = GREEN;
void LineDDA(int x1, int y1, int x2, int y2)
{
  int x = x1;
  float y = y1;
  float m = float(y2-y1)/(x2-x1);
  putpixel(x, Round(y), Color);
  for(int i=x1; i<x2; i++)
  {
    x++;
    y+=m;
    putpixel(x,Round(y),Color);
  }
} // LineDDA
```



2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

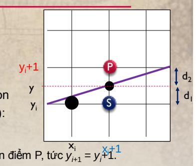
- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật Bresenham**
- Jack Elton Bresenham (11/10/1937, Clovis, New Mexico, Hoa Kỳ) là cựu giáo sư khoa học máy tính. Ông từng làm việc trong phòng thí nghiệm tính toán tại phòng thí nghiệm phát triển San Jose của IBM. Thuật toán được sử dụng cho máy vẽ Calcomp kết nối với IBM 1401 thông qua bảng điều khiển máy của đánh chữ 1407 vào năm 1962. Thuật toán của ông sau đó đã được mở rộng để xây dựng đường tròn (Bresenham's midpoint circle algorithm). Thuật toán Bresenham sử dụng hầu hết trong sản phẩm phần cứng như máy vẽ và trong chip đồ họa của card đồ họa hiện đại và trong nhiều thư viện phần mềm đồ họa.

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật Bresenham (tt)**
- Thuật toán Bresenham đưa ra cách chọn y_{i+1} là y_i hay y_i+1 theo một hướng khác sao cho có thể tối ưu hóa về mặt tốc độ so với thuật toán DDA. Vấn đề mấu chốt ở đây là làm thế nào để hạn chế tối đa các phép toán trên số thực.
- Gọi P (x_i+1, y) là điểm thuộc đoạn thẳng.
- Ta có: $y = m(x_i+1) + b$

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật Bresenham (tt)**
- Đặt $d_1=y-y_i$ và $d_2=(y_i+1)-y$
- Xét tất cả các vị trí tương đối của y so với y_i và y_i+1, việc chọn điểm (x_{i+1}, y_{i+1}) là S hay P phụ thuộc vào việc dấu của (d_1-d_2):



- Nếu (d_1-d_2) < 0, điểm S được chọn, tức là $y_{i+1}=y_i$. Ngược lại, ta chọn điểm P, tức $y_{i+1} = y_i+1$.
- Xét $p_i = D_x(d_1-d_2) = Dx(2y-2y_i - 1) \Rightarrow p_i = Dx\,[2(m\,(x_i+1)+ b) - 2y_i-1]$
- Thay $m = \frac{D_y}{D_x}$ vào phương trình trên ta được: $p_i = 2Dyx_i - 2Dxy_i + c$, với $c = 2Dy + (2b - 1)Dx$

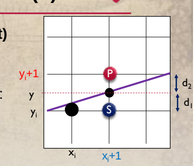
2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật Bresenham (tt)**
- Nhận xét rằng do D_x > 0 nên dấu của biểu thức (d_1-d_2) cũng chính là dấu của p_i. Hay nói cách khác, nếu tại bước thứ i ta xác định được dấu của p_i thì xem như xác định được điểm cần chọn ở bước (i+1). Vấn đề còn lại là làm thế nào để tính được p_i tại mỗi bước thật nhanh.
- Ta có: $p_{i+1} - p_i = (2D_yx_{i+1} - 2D_xy_{i+1} + c) - (2D_yx_i - 2D_xy_i + c)$
  $\Leftrightarrow p_{i+1} - p_i = 2D_y\,(x_{i+1} - x_i) - 2Dx\,(y_{i+1} - y_i)$
  $\Leftrightarrow p_{i+1} - p_i = 2D_y - 2D_x\,(y_{i+1} - y_i)$, do $x_{i+1} = x_i + 1$

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật Bresenham (tt)**
- Từ đây ta có thể suy ra cách tính p_{i+1} từ p_i như sau:

  Nếu p_i < 0 thì $p_{i+1} = p_i + 2D_y$ do ta chọn $y_{i+1} = y_i$.



- Ngược lại, nếu $p_i \geq 0$, thì $p_{i+1} = p_i + 2D_y - 2D_x$, do ta chọn $y_{i+1} = y_i + 1$

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật Bresenham (tt)**
- Giá trị p_0 được tính từ điểm vẽ đầu tiên (x_0, y_0) theo công thức:

  $p_0 = 2D_yx_0 - 2D_xy_0 + c = 2D_yx_0 - 2D_xy_0 + 2D_y-(2b - 1)D_x$

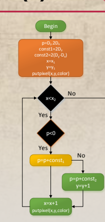
- Do (x_0, y_0) là điểm nguyên thuộc về đoạn thẳng nên ta có: $y_0 = mx_0 + b = \frac{D_y}{D_x}x_0$+b.
- Thế vào phương trình trên ta suy ra: $p_0 = 2D_y - D_x$

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật Bresenham**

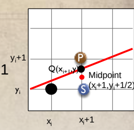
```
void LineBres (int x1, int y1, int x2, int y2)
{
  int Dx, Dy, p, Const1, Const2;
  int x, y;
  Dx = x2 - x1;
  Dy = y2 - y1;
  p = 2*Dy - Dx; // (Dy <<1) - Dx
  Const1 = 2*Dy; // Dy <<1
  Const2 = 2*(Dy-Dx); // (Dy-Dx) <<1
  x = x1;
  y = y1;
  putpixel(x, y, Color);
```

```
for(i=x1; i<x2; i++)
 {
   if (p<0)
   {
    p += Const1;
   }
   else
   {
    p += Const2;
    y++;
   }
   x++;
   putpixel(x, y, Color);
 }
} // LineBres
```



2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở**
- **Giải thuật trung điểm (Midpoint)**
- Thuật toán MidPoint đưa ra cách chọn y_{i+1} là y_i hay y_i+1 bằng cách so sánh điểm thực $Q(x_i+1,y)$ với điểm MidPoint là trung điểm của S và P. Ta có:
- Nếu điểm Q nằm dưới điểm MidPoint, ta chọn S.
- Ngược lại nếu điểm Q nằm trên điểm MidPoint ta chọn P.



2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

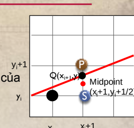
- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật trung điểm (Midpoint) (tt)**
- Ta có dạng tổng quát của phương trình đường thẳng:

  $Ax + By + C = 0$, với $A = y_2 - y_1$, $B = -(x_2 - x_1)$, $C = x_2y_1 - x_1y_2$
- Đặt $f(x,y) = Ax + By + C$, ta có nhận xét:

$$f(x,y)\begin{cases} < 0, \text{nếu } (x,y) \text{ nằm phía trên đường thẳng} \\ = 0, \text{nếu } (x,y) \text{ thuộc đường thẳng} \\ > 0, \text{nếu } (x,y) \text{ nằm phía dưới đường thẳng} \end{cases}$$

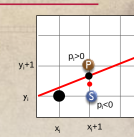
2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật trung điểm (Midpoint) (tt)**
- Lúc này việc chọn các điểm S, P ở trên được đưa về việc xét dấu của
- $p_i = f(midpoint) = f(x_i+1, y_i+\frac{1}{2})$
- Nếu $p_i < 0$, điểm MidPoint nằm phía trên đoạn thẳng. Lúc này điểm thực Q nằm dưới điểm MidPoint nên ta chọn S, tức là $y_{i+1} = y_i$.
- Ngược lại, nếu $p_i \geq 0$, điểm MidPoint nằm phía dưới đoạn thẳng. Lúc này điểm thực Q nằm trên điểm MidPoint nên ta chọn P, tức là $y_{i+1} = y_i+1$.



2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật trung điểm (Midpoint) (tt)**
- Mặt khác: $p_{i+1} - p_i = f(x_{i+1}+1, y_{i+1}+\frac{1}{2}) - f(x_i+1, y_i+\frac{1}{2})$
- $p_{i+1} - p_i = [A(x_{i+1}+1)+B(y_{i+1}+\frac{1}{2})+C] - [A(x_i+1)+B(y_i+\frac{1}{2})+C]$
- $p_{i+1} - p_i = A(x_{i+1} - x_i) + B(y_{i+1} - y_i)$
- Vậy: $p_{i+1} = p_i + D_y$, nếu $p_i < 0$, do ta chọn S: $y_{i+1} = y_i$
- $p_{i+1} = p_i + D_y - D_x$, nếu $p_i \geq 0$ do ta chọn P: $y_{i+1} = y_i+1$



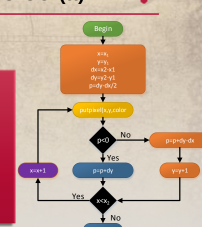
2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật trung điểm (Midpoint) (tt)**
- Ta tính giá trị p_0 ứng với điểm ban đầu (x_0, y_0), với nhận xét rằng:
- (x_0, y_0) là điểm thuộc về đoạn thẳng, tức: $Ax_0 + By_0 + C = 0$
- $p_0 = f(x_0+1, y_0+\frac{1}{2}) = A(x_0+1)+B(y_0+\frac{1}{2}) + C$
- $\Rightarrow p_0 = (Ax_0 + By_0 + C) + A + B/2 = A + B/2 = D_y - Dx/2$

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Giải thuật trung điểm (Midpoint) (tt)**

```
void Mid_line(int x1, int y1, int x2, int y2,
int c)
{
  int x, y, dx, dy,d;
  y = y1;
  dx = x2 - x1;
  dy = y2 - y1;
  p= dy - dx/2;
  for (x=x1; x<=x2; x++)
  {
    putpixel(x, y, color);
  if (p<0)
     {
       p = p + dy;
     }
  else
    {
      y ++;
      d = d + dy - dx;
    }
  }//end for
}//end midpoint line
```



2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- So sánh DDA và Bresenham

	DDA	Bresenham
Số học	Số thực	Số nguyên
Phép toán	Nhân và Chia	Cộng và Trừ
Tốc độ	Chậm hơn	Nhanh hơn
Độ chính xác	Không bằng Bresenham	Chính xác hơn DDA
Hạn chế	DDA cho sai số lớn hơn khi áp dụng để vẽ đường cong, đường tròn	
Bộ nhớ	Tốn bộ nhớ hơn do phải xử lý dấu phẩy động	Ít tốn bộ nhớ hơn do chỉ thực hiện phép toán cộng và trừ

2.2. CÁC ĐỐI TƯỢNG ĐỒ HỌA CƠ SỞ (tt)

- **2.2.3. Các giải thuật xây dựng thực thể cơ sở (tt)**
- **Bài tập áp dụng:**
- **Bài 1.** Sử dụng thuật toán Bresenham trình bày các bước vẽ đường thẳng đi qua 2 điểm A(3,7) và B(9,10).
- **Bài 2.** Sử dụng thuật toán Midpoint trình bày thứ tự các bước vẽ đường thẳng đi qua 2 điểm A(4,8) và B(10,12).

thank you

Xin chân thành cám ơn quý thầy cô đã quan tâm theo dõi