Компютърна графика

Информатика IV курс, II поток 2012/2013 г.

Лекция № 2

лектор: гл. ас. д-р В. Гушев

по курса на $\boxed{\text{доц. д-р Спас Петров Ташев}}$

Растеризиране на примитиви

Примитиви – най-прости криви, които се използват за чертане на фигури върху екрана, напр. *отсечка, окръжност, дъга от окръжност, елипса, символи.*

Да разгледаме правоъгълна координатна система с център долния ляв ъгъл на екрана. В нея точките с координати цели неотрицателни числа ще наричаме пиксели.

Правоъгълникът, в който варират пикселите ще наричаме "растерен дисплей" или само "дисплей" ("екран"). За нас това ще бъде част от екрана на монитора.

Растеризиране на отсечка

Нека е зададена отсечка (сегмент) с крайни точки $A = (a_1, a_2)$ и $B = (b_1, b_2)$.

Избирането на онези пиксели, които са най-близо до точките от отсечката се нарича растеризация на отсечката, а избраните пиксели – растер на отсечката.

Алгоритмите за чертане на отсечка трябва да изпълнявата някои изисквания:

1. Растеризираната отсечка трябва да прилича на отсечка, крайните точки трябва да са избани точно.

2. Осветеността на сегмента трябва да не зависи от мястото и ориентацията.

3. Сегмента трябва да се изчертава бързо.

Алгоритъм на Брезенхам за растеризиране на отсечка

Нека h е големината на страната на пиксела, или разстоянието между пикселите, като в реалния случай $h = 1$.

В зависимост от ъгловия коефициент m на отсечката на всяка стъпка от алгоритъма едната от координатите x или y се променя (увеличава или намалява) с h, а другата координата или се променя с h или остава непроменена, в зависимост от разстоянието между пресечната точка на отсечката и правата, успоредна на координатната ос на другата променлива и намираща се на разстояние hот последния избран пиксел в посока на изменение на първата променлива и най-близкия пиксел от тази права.

Ние ще се интересуваме от това разстояние и ще го наричаме грешка.

Алгоритъмът проверява само знака на грешката, т.е. проверява дали тази пресечна точка се намира под или над (наляво или надясно от) полунивото между пикселите.

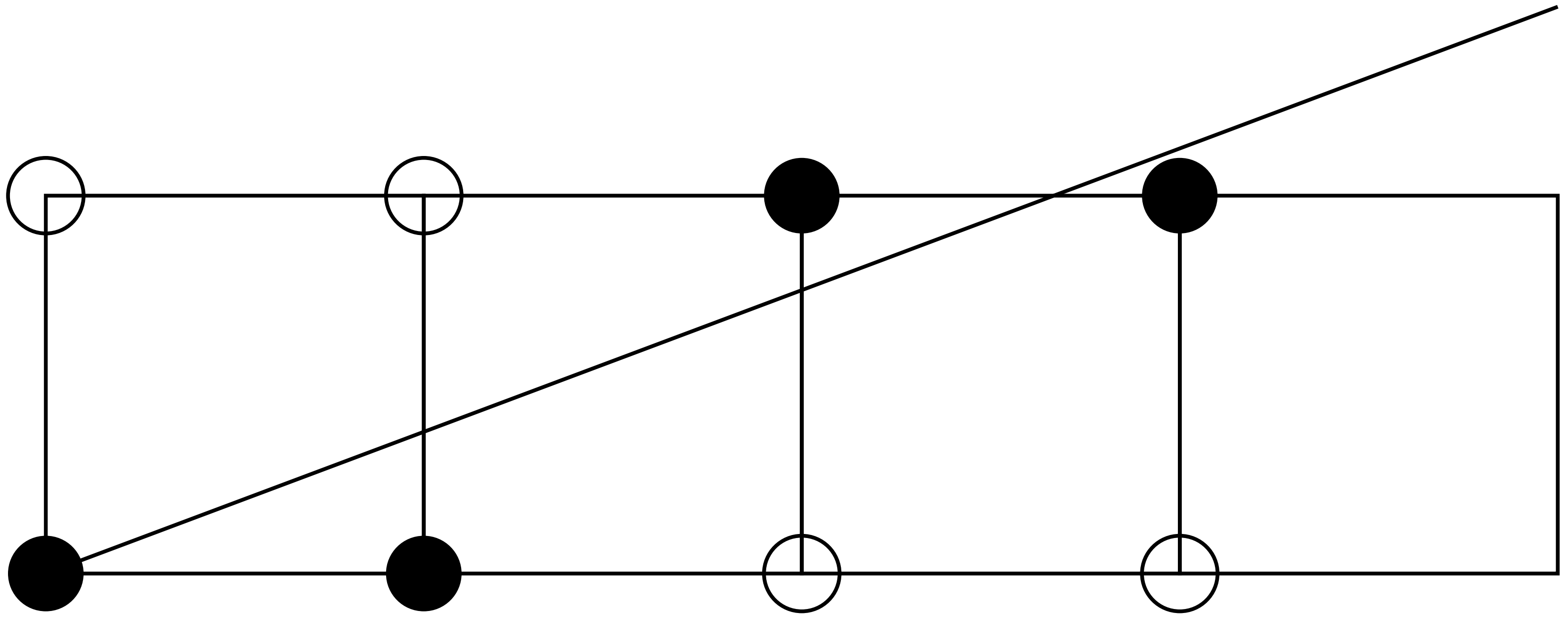

Fig. 2.1

На фигурата е начертана част от грида на пикселите и четири пиксела (черните точки) които приближават наклонената отсечка с наклон $3/8$. При наклон на отсечката m такъв, че $0 \leqq m \leqq 1$, на всяка стъпка x ще се увеличава с h, а y ще се увеличи с h или не, в зависимост от знака на грешката e. Ако отсечката започва от пиксела означен с 1 и с координати (x, y), то този пиксел трябва да бъде избран. Грешката е 0, но тъй като ще се интересуваме само от знака на грешката, то задаваме първоначална стойност на грешката $e = -h/2$. След това x се увеличава с h. Увеличаваме грешката с mh, т.е. $e = e + mh$. За нашия пример грешката става $-1/8h$. Пак е отрицателна и $y-$координата не се променя, т.е. от пикселите от вертикалната права 2 трябва да изберем черния пиксел.

На следващата стъпка $x-$координата нараства пак с h. Грешката нараства с mh и става $2/8h$. Сега вече грешката е положителна и $y-$координата нараства с h, т.е. $y = y + h$. На вертикала 3 това е черният пиксел. Преди да продължим нататък трябва да инициализираме грешката наново. Правим това като извадим h от нея, т.е. $e = e - h$.

Тъй като проверяваме само знака на грешката, за да работим с цели числа можем да умножим с $2\Delta_x$ равенствата

$$
\begin{aligned}
e &= -\frac{h}{2} \\
e &= e + \frac{\Delta_y}{\Delta_x} h, \quad m = \frac{\Delta_y}{\Delta_x} \\
e &= e - h
\end{aligned}
$$

и да положим $\bar{e} = 2e\Delta_x$.

Ще имаме

$$\begin{aligned} 2e\Delta_x &= -\Delta_x h \\ 2e\Delta_x &= 2e\Delta_x + 2\Delta_y h \\ 2e\Delta_x &= 2e\Delta_x - 2\Delta_x h \end{aligned}$$

т.е.

$$\begin{aligned} \bar{e} &= -\Delta_x h \\ \bar{e} &= \bar{e} + 2\Delta_y h \\ \bar{e} &= \bar{e} - 2\Delta_x h \end{aligned}$$

Алгоритъм на Брезенхам за I октант

Нека краищата на сегмента да са (x_1, y_1) и (x_2, y_2).

```
Bresenham.partial(x1,y1,x2,y2)
/* Инициализиране на променливите */
x=x1
y=y1
dx=x2-x1
dy=y2-y1
e=2*dy*h-dx*h
/* Главния цикъл */
while (x<x2)
   setpixel(x,y)
   while (e>0)
     y=y+h
     e=e-2*dx*h
   end while
   x=x+h
   e=e+2*dy*h
end while
```

Ако наклонът на сегмента по абсолютна стойност е по-голям от 1, то трябва $y-$координатата да се променя на всяка стъпка, а за $x-$координатата трябва да се приложи правилото на Брезенхам. От следващата диаграма се вижда лесно и как ще изглежда алгоритъма в общия случай.

Алгоритъм на Брезенхам в общия случай

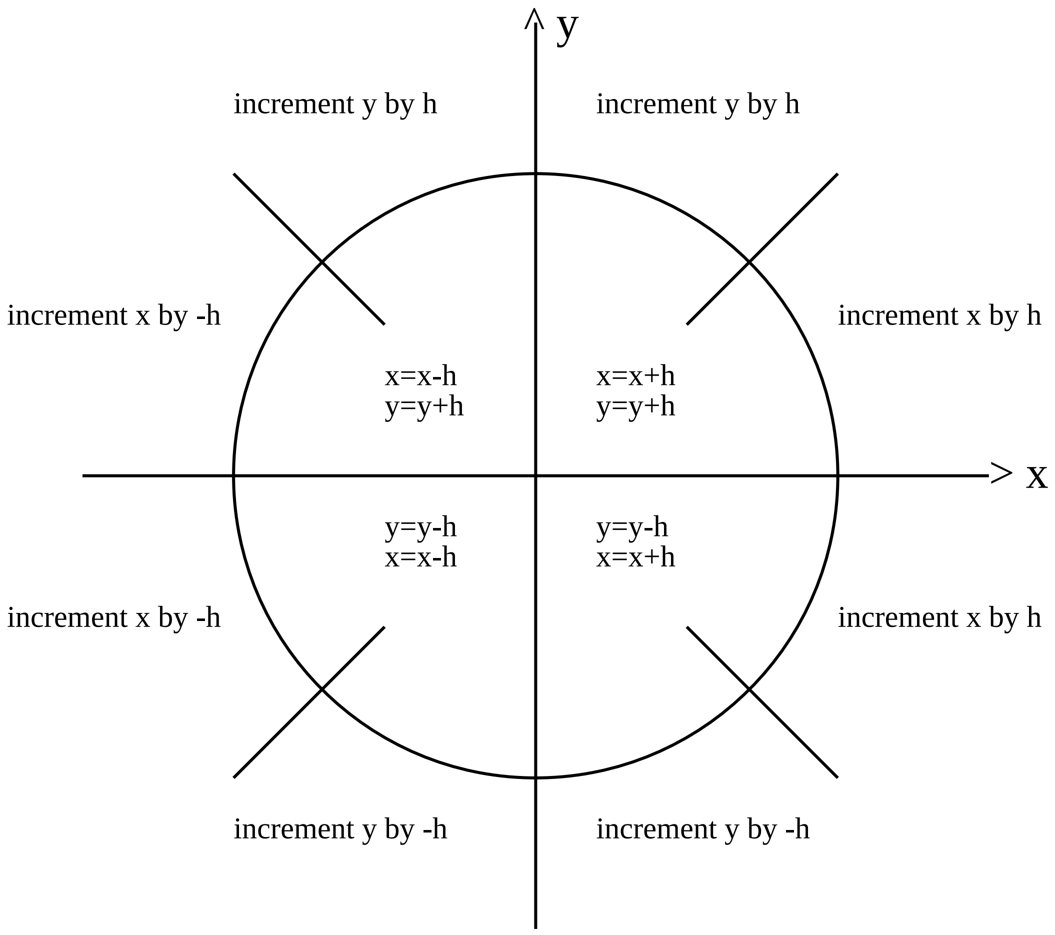

Fig. 2.2

```
Bresenham.general(x1,y1,x2,y2)
/* Инициализиране на променливите */
x=x1
y=y1
dx=abs(x2-x1)
dy=abs(y2-y1)
s1=sign(x2-x1)
s2=sign(y2-y1)
/* размяна на dx и dy в зависимост от наклона */
if dy > dx then
   temp=dx
   dx=dy
   dy=temp
   interchange=1
else
   interchange=0
end if
/* Инициализиране на грешката */
e=2*dy*h-dx*h
```

```
/* Главния цикъл */
for i=0 step h to dx
   setpixel(x,y)
   while (e>0)
     if (interchange = 1) then
       x=x+s1*h
     else
       y=y+s2*h
     end if
     e=e-2*dx*h
   end while
   if (interchange = 1) then
     y=y+s2*h
   else
     x=x+s1*h
   end if
   e=e+2*dy*h
end for
```

Алгоритъм на средната точка за растеризиране на отсечка

Нека отсечката да е зададена с наклон в интервала $[0, 1]$ и с краища $A = (x_1, y_1)$ и $B = (x_2, y_2)$. Да запишем уравнението на правата през точките A и B във вида

$$f(x, y) \equiv ax + by + c = 0,$$

където $a = -dy, b = dx, c = x_1y_2 - x_2y_1,\ dx = x_2 - x_1,\ dy = y_2 - y_1$.

Нека да си припомним, че векторът с така определените координати $\overrightarrow{N} = (a, b) = (-dy, dx)$ е перпендикулярен на правата и посоката му е такава, че стойността на f в точка от полуравнината в която той сочи е положителна. Този факт се използва в метода на средната точка.

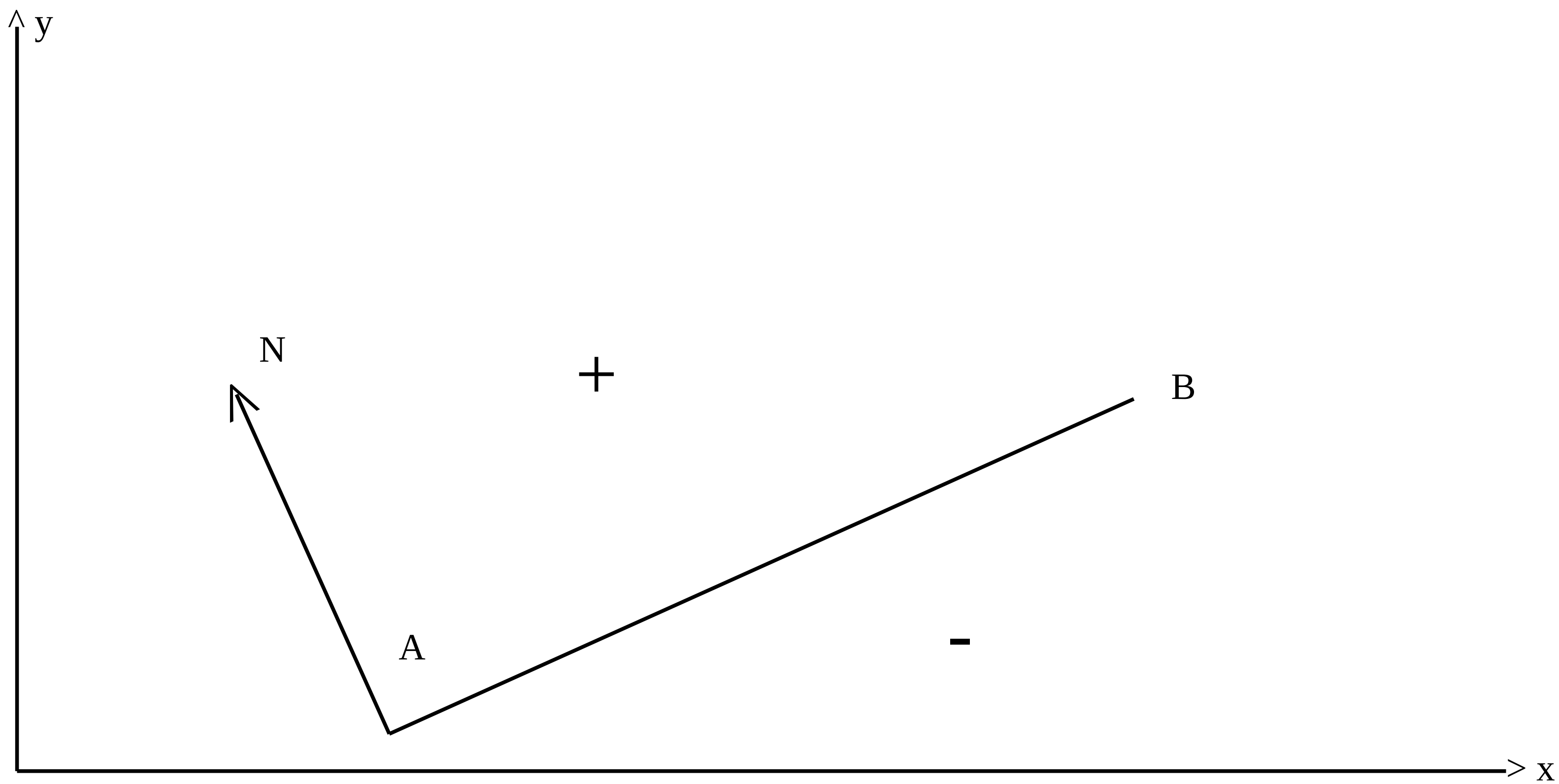

Fig. 2.3

Нека текущата точка да е $P = (x, y)$ и на следващата стъпка $x = x + h$, да трябва да избираме измежду хоризонталната точка $H = (x + h, y)$ и диагоналната точка $D = (x + h, y + h)$.

За целта ще гледаме дали грешката $E = f(M)$ е положителна или отрицателна. Ако $E < 0$, то нашата права е пресякла вертикалната права през точката $x + h$ над средната точка на точките H и D и следователно пиксела D е по-близо до правата и него избираме. В другия случай $E \geqq 0$ избираме H. Трябва да намерим новата грешка:

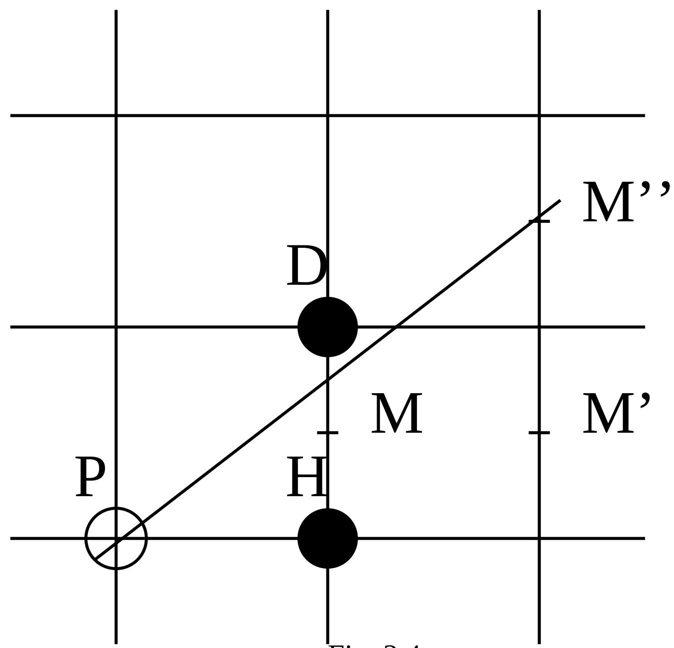

Fig. 2.4

Ако е избрана точка H, то новата грешка ще бъде

$$E_H = f(M') = f(M) + f(M') - f(M) = E_P + [a(x+2h) + b(y+\frac{h}{2}) + c] - [a(x+h) + b(y+\frac{h}{2}) + c] = E_P + ah$$

т.е.

$$E = E - dy.h \quad (1)$$

Ако е избрана точка D то:

$$E_D = f(M'') = f(M) + f(M'') - f(M) = E + [a(x+2h) + b(y+h+\frac{h}{2}) + c] - [a(x+h) + b(y+\frac{h}{2}) + c] = E_P + ah + bh$$

т.е.

$$E = E - dy.h + dx.h \tag{2}$$

В началото инициализираме

$$x = x_1,\ y = y_1,$$

$$E = f(M) = -dy(x_1 + h) + dx(y_1 + \frac{h}{2}) + c =$$

$$= f(x_1, y_1) + -dy\ h + dx\ \frac{h}{2} = -dy\ h + dx\ \frac{h}{2}.$$

За да работим само с цели числа можем да умножим всички пресмятания по 2, т.е. да работим с нова грешка $(\bar{E}) = 2E$.