Přednáška 2

Algoritmy pro kreslení dvourozměrných objektů

Kreslení úsečky

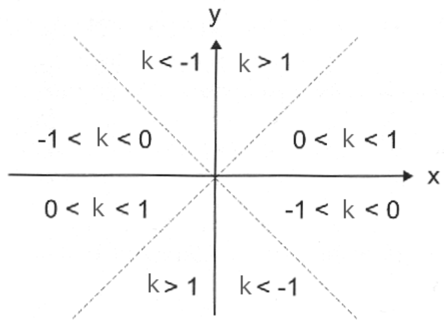
Nejkratší spojnice dvou bodů.

Definice úsečky:
- počátečním a koncovým bodem [x_1, y_1], [x_2, y_2]
- počátečním bodem [x_1, y_1] a vektorem rozdílů (Δx, Δy)=(x_2-x_1, y_2-y_1)

Rovnice přímky $y = kx + c$, kde

$$k = \frac{Y_2 - Y_1}{X_2 - X_1}$$

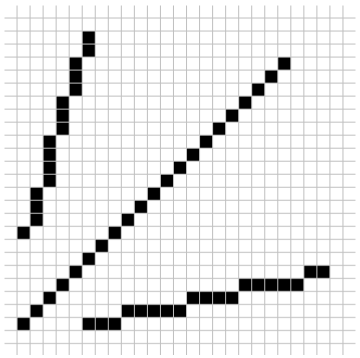
$$c = \frac{X_2Y_1 - X_1Y_2}{X_2 - X_1}$$



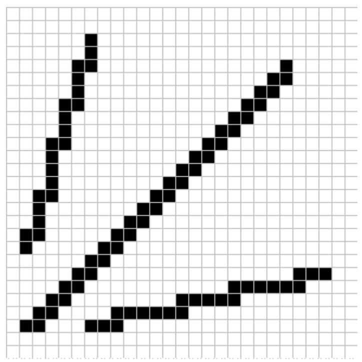
Rasterizace

– převedení objektu na posloupnost bodů

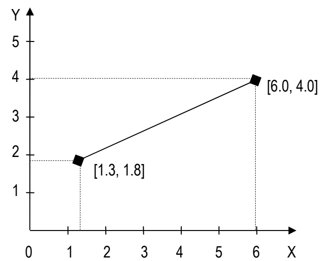
A – osmispojitá (8-connected)

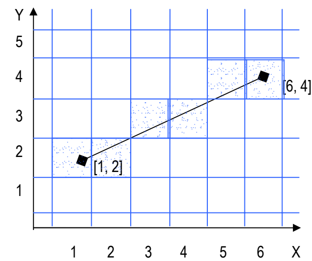


B – čtyřspojitá (4-connected)



Geometrické a rastrové zobrazení úsečky





Problémy: – neceločíselné zadání souřadnic
– používání směrnice (reálná hodnota)
– většina algoritmů předpokládá celočíselné hodnoty

Přijatelné řešení – zaokrouhlení počátečního a koncového bodu

Algoritmy kreslení

Algoritmus založený na výpočtu obou souřadnic na základě rovnice přímky (jednoduchý, ale výpočtově náročný)

Algoritmus DDA (digital differential analyzer) – postupné přičítání konstantních přírůstků k oběma souřadnicím, počínaje prvním bodem úsečky

Bresenhamův algoritmus – nalezení bodů ležících nejblíže skutečné úsečce pouze pomocí celočíselné aritmetiky

Výpočet obou souřadnic

V každém kroku je násobení reálných hodnot, sčítání reálných hodnot a zaokrouhlování.

	A	B	C	D
1	x1=	1,3	x2=	6
2	y1=	1,8	y2=	4
3				
4	k=	0,4681		
5	c=	1,1915		
6				
7	x	y	Round(y)	
8	1,3	1,8000	2	
9	2	2,1277	2	
10	3	2,5957	3	
11	4	3,0638	3	
12	5	3,5319	4	
13	6	4,0000	4	
14				

DDA

- Předpoklad $x_1 < x_2$
- !!! $x_1 = x_2$
- Výpočet směrnice *k*
- Rozlišuje se případ:

- |k|<1, potom řídící osou je osa x,
  $x_{i+1} := x_i + 1$
  $y_{i+1} := y_i + k$

- |k|>1, potom řídící osou je osa y
  $y_{i+1} := y_i + 1$
  $x_{i+1} := x_i + 1/k$

- Začíná se počátečním bodem, který se kreslí $[x_1,y_1]$
- Cyklicky se opakuje pro ostatní body $[x_i,y_i]$
- Neceločíselné souřadnice je třeba zaokrouhlit

DDA

Jiný postup:

$dx=X_2-X_1$
$dy=Y_2-Y_1$

počet kroku = max (|dx|, |dy|)

px=dx / počet kroků
py=dy / počet kroků

$x_{i+1} := x_i + px$
$y_{i+1} := y_i + py$

Kreslí se bod $[Round(x_{i+1}), Round(y_{i+1})]$

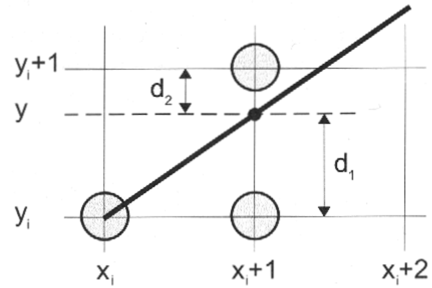
Výpočet DDA

	A	B	C	D
1	x1=	1	x2=	6
2	y1=	2	y2=	4
3				
4	dy=	0,4000		
5	start_y	2,0000		
6				
7	x	y	Round(y)	
8	1,3	2,0000	2	
9	2	2,4000	2	
10	3	2,8000	3	
11	4	3,2000	3	
12	5	3,6000	4	
13	6	4,0000	4	
14				

Bresenhamův algoritmus

Funguje na základě určování chybového členu E_D (Error Differential), který představuje vzdálenosti k dvou nejbližším možným bodům rastru

Rozlišuje se případ k>1 a k<1



Pro x_i+1 nastane jedna ze dvou možností $y_{i+1}= y_i$, nebo $y_{i+1}= y_i+1$, uvažujeme-li splnění podmínek: $0<k<1$

Vhodným vyjádřením E_D stačí testovat pouze znaménko

Besenhamův algoritmus – odvození

- předpokládejme bod $[x_i,y_i]$
- následující může ležet na souřadnicích $[x_i+1,y_i]$ nebo $[x_i+1,y_i+1]$
- d_1 a d_2 představují rozdíly mezi středy uvedených pixelů a skutečnou souřadnicí y na úsečce v bodě x_i+1
- základem je rovnice přímky: $y=kx + c$
- $y=k (x_i+1) + c$
- $d_1= y - y_i \quad = k (x_i+1) + c - y_i$
- $d_2= y_i+1 - y = y_i+1 - k (x_i+1) - c$
- $\Delta d = d_1 - d_2 = 2k (x_i+1) – 2y_i + 2c –1$

- podle Δd dokážeme určit který pixel je blíže skutečné souřadnici (záporná představuje y_i, kladná y_{i+1}),
  stačí znaménko → je vhodný převod do celočíselné aritmetiky, protože $k=\Delta y/\Delta x$, vynásobíme rovnici Δx

- $p_i = \Delta d \Delta x = 2\ \Delta y\ x_i – 2\ y_i\ \Delta x + 2\ \Delta y + \Delta x\ (2c –1)$ ** … konstanta
- p_i je nazývána **predikcí**
- $p_i = 2\ \Delta y\ x_i – 2\ y_i\ \Delta x$ + konst
- $p_{i+1} = 2\ \Delta y\ x_{i+1} – 2\ y_{i+1}\ \Delta x$ + konst
- odečtením rovnic za předpokladu že: $x_{i+1}- x_i=1$ dostaneme

$p_{i+1} = p_i + 2\ \Delta y - 2\ \Delta x\ (y_{i+1} - y_i)$

- z toho dostaneme vztahy pro výpočet p_{i+1} pomocí p_i

pokud (p_i <= 0) ….. $y_{i+1}=y_i$;	$p_{i+1} = p_i + 2\ \Delta y$
pokud (p_i > 0) ….. $y_{i+1}=y_i+1$;	$p_{i+1} = p_i + 2\ \Delta y - 2\ \Delta x$

první p_1 dostaneme **, kde dosadíme $[x_1,y_1]$ a zaokrouhlíme

Výpočet úsečky

	x	p	y
1	x	p	y
2	0	-14	5
3	1	-8	5
4	2	-2	5
5	3	4	5
6	4	-30	6
7	5	-24	6
8	6	-18	6
9	7	-12	6
10	8	-6	6
11	9	0	6
12	10	-34	7
13	11	-28	7
14	12	-22	7
15	13	-16	7
16	14	-10	7
17	15	-4	7
18	16	2	7
19	17	-32	8
20	18	-26	8
21	19	-20	8
22	20	-14	8

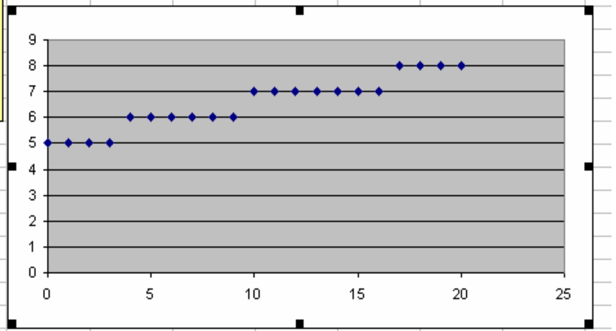
=ay

=2*dy*A2-2*Dx*C2+2*dy+Dx*(2*Posun-1)

=KDYŽ(B2<0;C2;C2+1)

=KDYŽ(B2<0;B2+2*dy;B2+2*dy-2*Dx)

Dx=	20		A[x,y]	0	5
Dy=	3		B[x,y]	20	8
Posun=	5				



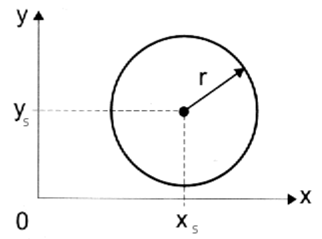
Kreslení kružnice

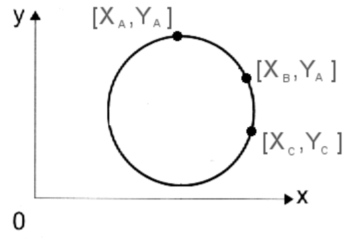
Definice úsečky:
- středem $[x_c, y_c]$ a poloměrem *r*
- pomocí tří bodů ležících na kružnici $[x_A, y_A]$, $[x_B, y_B]$, $[x_C, y_C]$

Rovnice kružnice $x = x_s + r \cos \alpha$

$y = y_s + r \sin \alpha$

$F(x,y)$: $x^2 + y^2 - r^2 = 0$





Algoritmy kreslení

Algoritmus založený na výpočtu obou souřadnic na základě parametrické rovnice kružnice pro α ($0 <= \alpha < 2\pi$)
- správné zvolení přírůstku $\Delta\alpha$ v závislosti na *r*
- nahrazení kružnice lomenou čarou

Pomocí otáčení – na jeden bod kružnice je opakovaně aplikováno otočení o úhel zvětšující se o krok $\Delta\alpha$

$x_2 = x_1 \cos \alpha - y_1 \sin \alpha$

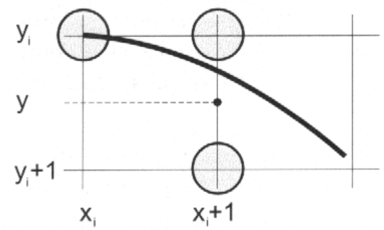
$y_2 = x_1 \sin \alpha + y_1 \cos \alpha$

Pomocí rovnice $y^2 = \pm\sqrt{(r^2 - x^2)}$, kde za x dosazujeme hodnoty z intervalu <-r, r>

Bresenhamův algoritmus – nalezení bodů ležících nejblíže skutečné kružnici pouze pomocí celočíselné aritmetiky

Bresenhamův algoritmus

“midpoint algoritmus”



- řídící osa x, vedlejší y
- změna na x je vždy o 1 pixel
- začátek pro [0, r]
- podmínka konce x>=y

- základní funkce: $F(x,y)$: $x^2 + y^2 - r = 0$
- znaménko určuje polohu bodu vůči kružnici
- předpokládejme bod na kružnici $[x_i, y_i]$
- následující bod: $[x_i+1, y_i]$ nebo $[x_i+1, y_i-1]$
- bod mezi uvedenými možnostmi – midpoint $[x_i+1, y_i - ½]$

Bresenhamův algoritmus

Dosazením midpointu do základní rovnice kružnice dostaneme:

$p_i = F(x_i+1, y_i - ½) = (x_i + 1)^2 + (y_i - ½)^2 - r^2$

Pokud $p_i<0$ → jako další bude nakreslen bod se stejnou souřadnicí y_i, jinak jako další bude nakreslen bod y_i-1

Predikce

$p_{i+1} = p_i + 2x_i + 3 + (y_i - ½)^2 + (y_{i+1} - ½)^2$

pokud ($p_i <= 0$),	...... $y_{i+1}=y_i$;	$p_{i+1} = p_i + 2x_i + 3$
pokud ($p_i > 0$),	...... $y_{i+1}=y_i+1$;	$p_{i+1} = p_i + 2x_i + 5 - 2y_i$

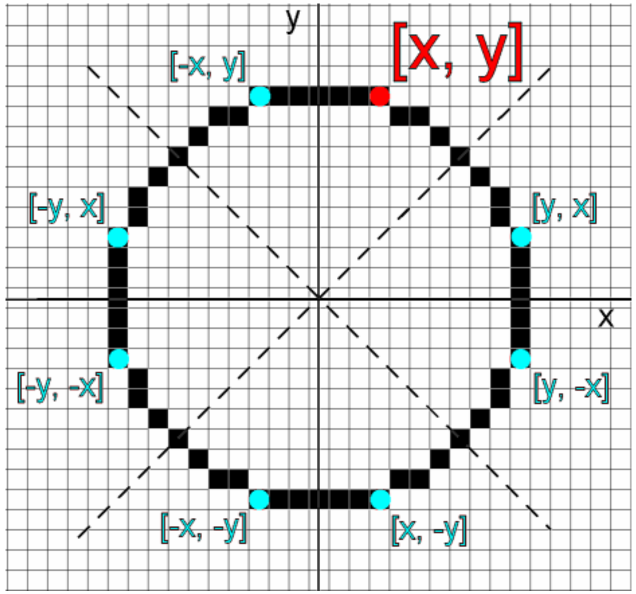
Počáteční nastavení $p_1 = 1 - r$;

Duplikování vypočítaných bodů

Výpočet stačí provést pouze pro jeden oktant, např. pro (0 < x <= y)

Zbývající body se vytvoří pomocí symetrie

Některé body jsou při symetrickém kopírování kresleny několikrát (nutno odstranit při vykreslování v režimu XOR)



Výpočet kružnice

	A	B	C
1	x	p	y
2	0	-19	20
3	1	-16	20
4	2	-11	20
5	3	-4	20
6	4	5	20
7	5	-22	19
8	6	-9	19
9	7	6	19
10	8	-13	18
11	9	6	18
12	10	-7	17
13	11	16	17
14	12	9	16
15	13	6	15
16	14	7	14
17	15		13
18	16		12
19	17		11
20	17		10
21	18		9
22	18		8
23	19		7
24	19		6
25	19		5
26	20		4
27	20		3
28	20		2
29	20		1
30	20		0
31			

=Polomer

Polomer= 20

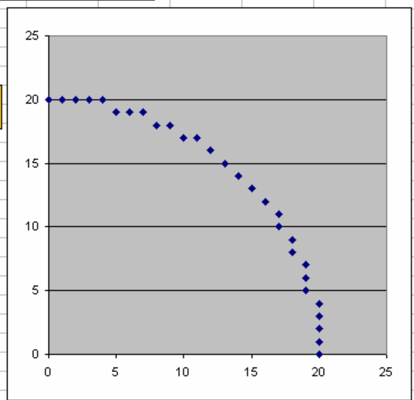
=KDYŽ(B2<0;C2;C2-1)

=1-Polomer

KDYŽ(B2<0;B2+2*A2+3;B2+2*A2+5-2*C2)

KONEC VÝPOČTU
Yi<=Xi

SYMETRIE



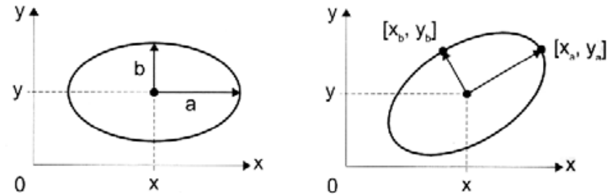
Kreslení elipsy

Definice elipsy:
- středem $[x_c, y_c]$ a délkami poloos *a*, *b*, popřípadě
- středem $[x_c, y_c]$ a délkami body $A=[x_A, y_A]$, $B=[x_B, y_B]$, ležícími na poloosách

Rovnice elipsy:

$x = x_c + a \cos \alpha$

$y = y_c + b \sin \alpha$



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

Bresenhamův algoritmus

Dosazením midpointu do základní rovnice elipsy dostaneme .....

Výpočet stačí provést pouze pro jeden kvadrant.
Během výpočtu v daném kvadrantu je třeba změnit řídící osy.
Změna řídících os probíhá v bodě, kde je směrnice tečny = -1 (45°)

... parciální derivace se sobě rovnají
$dF/dx=2b^2x$
$dF/dy=2a^2y$

Predikce

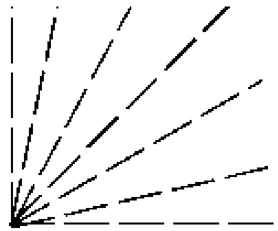
pokud ($p_i <= 0$),	...... $y_{i+1}=y_i$;	$p_{i+1} = p_i + b^2(2x_i + 1)$
pokud ($p_i > 0$),	...... $y_{i+1}=y_i+1$;	$p_{i+1} = p_i + b^2(2x_i + 1) – 2a^2y_i$

Počáteční nastavení $p_1= b^2-ba^2+a^2/4$;

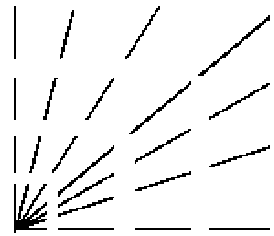
Kreslení přerušovaných čar

- jednodušší algoritmus určující vykreslení/nevykreslení vypočítaného pixelu na základě pomocného vzorku (pole) podle požadovaného vzhledu čáry (pracuje v metrice rastru)
- složitější algoritmus určuje nejdříve hranice přerušení (v metrice reálného systému) a takto stanovené úseky vykreslí

Začátek a konec čáry musí být vždy nakreslen.



správně



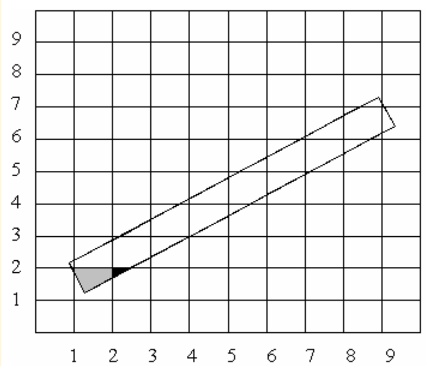
špatně

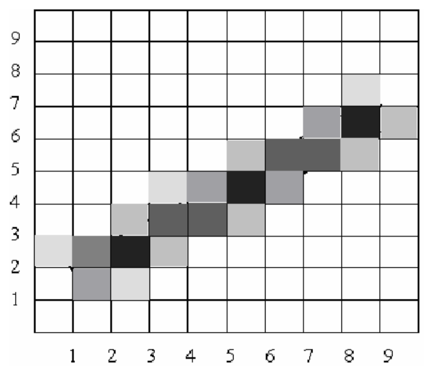
(Anti)aliasing

- metody zahrnující odstranění (potlačení) zubatosti čar
- je vyžadována možnost kreslení v odstínech zvolené barvy

- Všechny objekty mají určitou velikost.
- Bod není nekonečně malý.

Nastavení odstínu (intenzity), úměrné podle plochy pokrytí





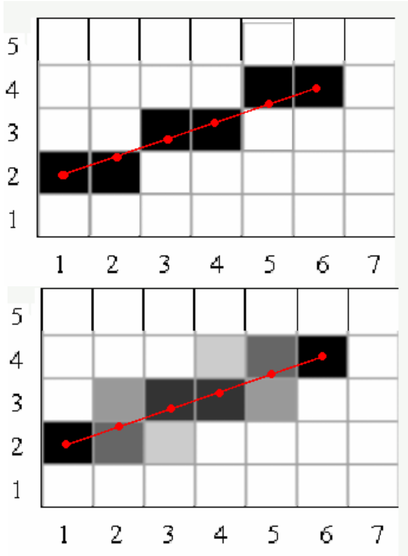
- Kreslí se vždy více (dva) pixelů, kterými úsečka prochází.
- Intenzita je určena podle vzdálenosti mezi úsečkou a středem pixelu
  - o desetinná část v DDA
  - o člen D v Bresenhamově alg.

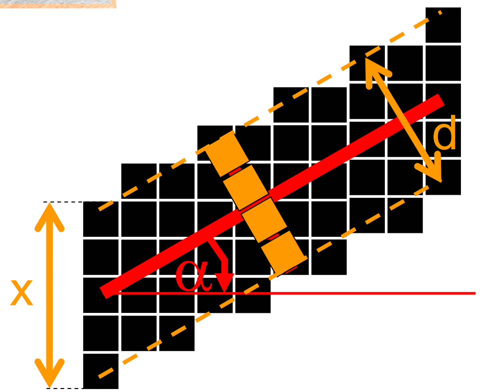
A=[1,2]; B=[6,4]; k=2/5

2. bod = [2, 2.4]
pixel[2,2] → 60 % = jas 102
pixel[2,3] → 40 % = jas 153

3. bod = [3, 2.8]
pixel[3,3] → 30 % = jas 204
pixel[3,4] → 70 % = jas 51

Jas platí pro černou (0) čáru na bílém (255) pozadí





$X = d / \cos(\alpha)$

Pro požadovanou čáru tlustou 4 pixely je x = 4,61 ⇒ kreslí se 5 pixelů nad sebou