

GRAFICKÁ INFORMÁCIA A JEJ SPRACOVANIE

doc. Ing. Branislav Sobota, PhD.

Katedra počítačov a informatiky, FEI TU v Košiciach

P 02

© 2024

VRSTVY VIZUALIZAČNÉHO PROCESU

1. Definovanie/spracovanie modelu (reprezentácia, súradnicové systémy)
2. Transformácie nad objektami
3. Riešenie viditeľnosti
4. Tieňovanie
5. Osvetľovanie
6. Realistické zobrazovanie
7. Kompozícia a Vykresľovanie



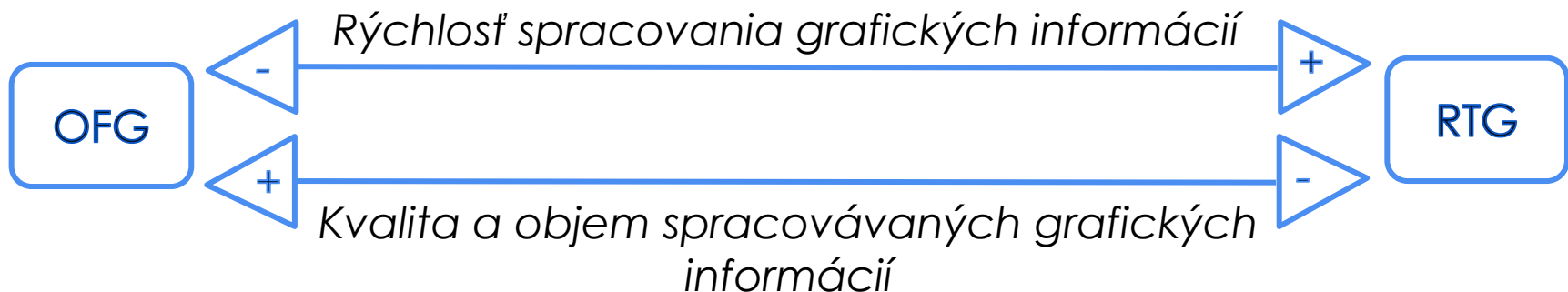
GRAFICKÁ INFORMÁCIA (OBJEKTY)

1. Typy
2. Primitíva
 1. implementácia primitív
 2. filtračné a rozptyľovacie metódy
3. Popis objektu
4. Reprezentácia objektu
5. Priestor objektu



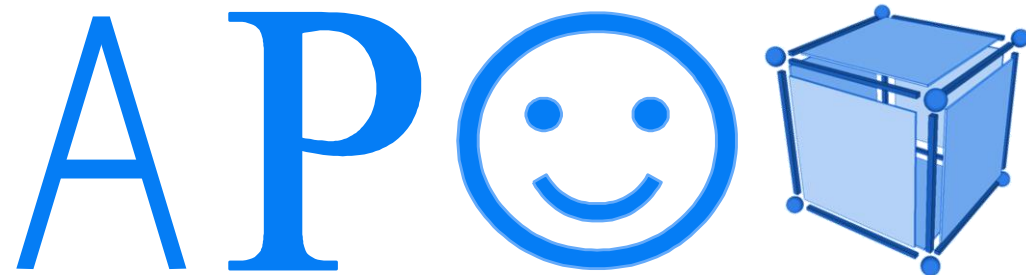
GRAFICKÁ INFORMÁCIA (SPRACOVANIE)

1. Grafická informácia spracovávaná mimo reálneho času (*OFG* (off-line graphics))
2. Grafická informácia spracovávaná v reálnom čase (*RTG* (real-time graphics))

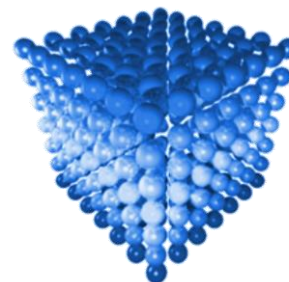
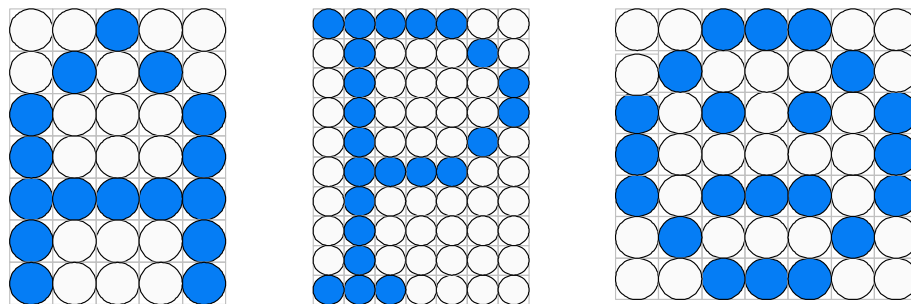


GRAFICKÁ INFORMÁCIA - TYP

1. Vektorová
(spojitý priestor,
rôzne entity)



2. Rastrová
(nespojité priestor,
len body)



GRAFICKÁ INFORMÁCIA

POUŽITIE VEKTOROVEJ GRAFIKY

- Schémy
- Grafy
- Konštruovanie
- Dizajn
- Diagramy
- Reklama
- Mapy
- Piktogramy
- ...



GRAFICKÁ INFORMÁCIA

POUŽITIE RASTROVEJ GRAFIKY

- Obrázky
- Fotografie
- Filmy
- Komunikácia
- Skenovanie
- Televízia
- Biometria
- ...

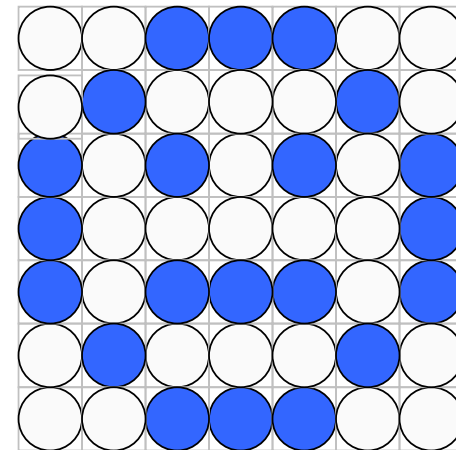


GRAFICKÁ INFORMÁCIA

TRANSFORMÁCIE TYPOV



Vektorový typ



Rastrový typ

GRAFICKÉ OBJEKTY

1. Vektorové

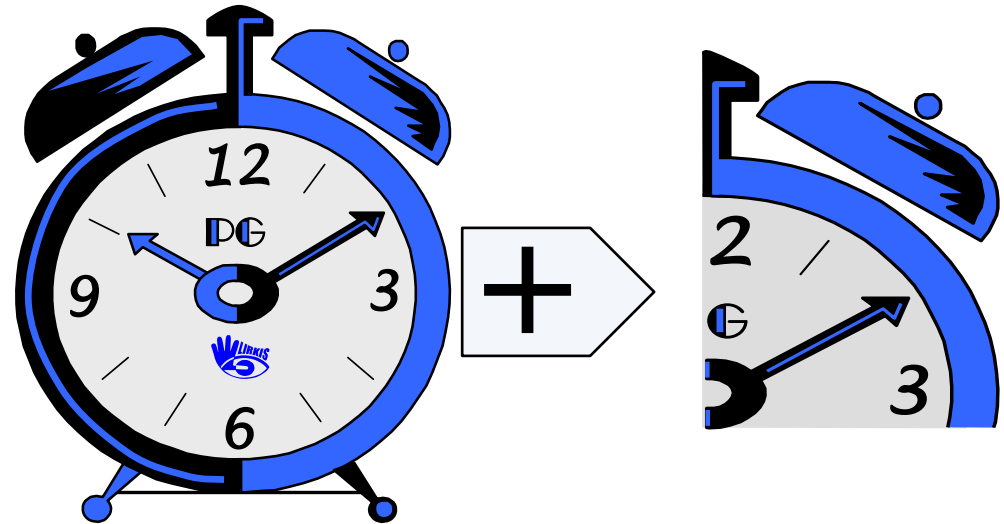


2. Rastrové

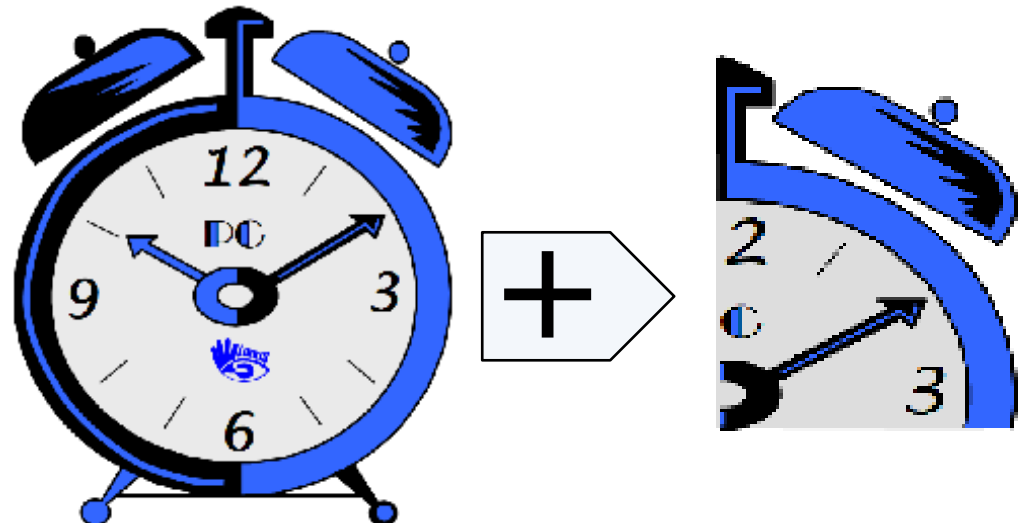


ZVÄČŠENIE OBJEKTOV

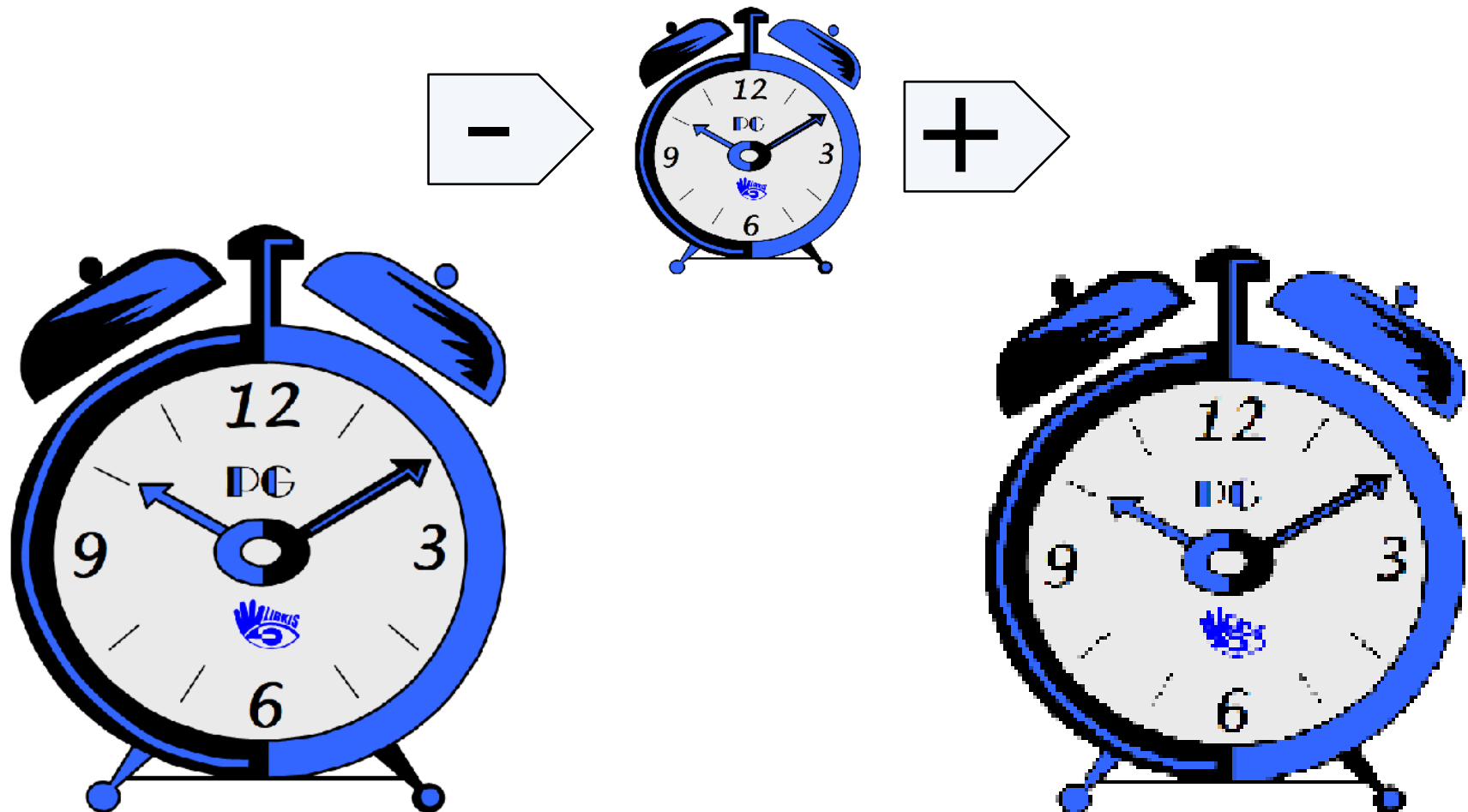
1. Vektorové



2. Rastrové



STRATA INFORMÁCIE PRI RASTRI

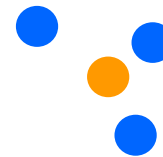


GRAFICKÉ PRIMITÍVA

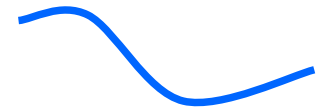
1. Bod
2. Sled bodov
3. Krivka
4. Lomená čiara
5. Grafický text
6. Plocha
7. Vyplnená oblasť
8. Výplňový vzor
9. Všeobecný grafický prvok



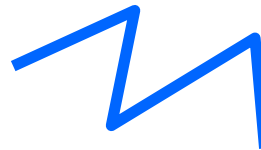
1



2



3



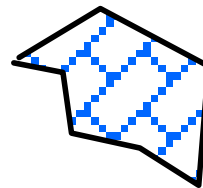
4



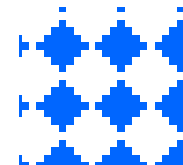
5



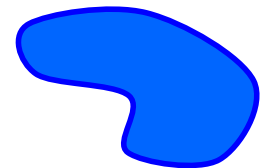
6



7



8



9

GRAFICKÉ PRIMITÍVA - ATRIBÚTY

Konečný tvar primitív je možné riadiť ich atribútmi. Medzi základné atribúty z pohľadu počítačovej grafiky zaradujeme najmä:

1. Farba
2. Typ (napr. čiar, písmo a pod.)
3. Hrúbka (napr. čiar, písmo a pod.)
4. Poloha (napr. písmo)
5. Smer vykreslenia (napr. horizontálny, vertikálny atď.)

GRAFICKÉ PRIMITÍVA - ATRIBÚTY

Atribúty môžu byť jednotlivým elementom priradené:

1. *konvenčne* alebo tiež *individuálne*, *pevne*, čo vedie niekedy k nekompatibilité na rôznych zobrazovačoch.
2. *symbolicky*, najčastejšie formou kódu. Vtedy hovoríme o *viazaných* (bundled) atribútoch. Tieto sú vzhľadom na zobrazovacie zariadenie transparentné.



Príklady rôznych atribútov čiar

BOD

Bod je z pohľadu počítačovej grafiky chápaný ako elementárny (atomárny) objekt.

Základnými atribútmi bodu z pohľadu počítačovej grafiky, sú: jeho *poloha* a *farba* príp. *čas* (v prípade heterogénnej štruktúry dimenzie priestoru, ktorej je bod prvkom)

Niektoré typy bodov:

- **Pixel** – obrazový bod, ktorý je charakterizovaný dvomi súradnicami polohy a svojou farbou. Z pohľadu PG je pixel chápaný ako najmenšia jednotka digitálnej rastrovej grafiky.
- **Voxel** – objemový bod, charakterizovaný tromi polohovými súradnicami a svojou farbou.
- **Texel** – bod textúry, okrem polohy v rámci určitej súradnicovej sústavy má ešte definovanú aj polohu v rámci výplňového vzoru (textúry) a reláciu svojho priradenia vyplňanej oblasti.

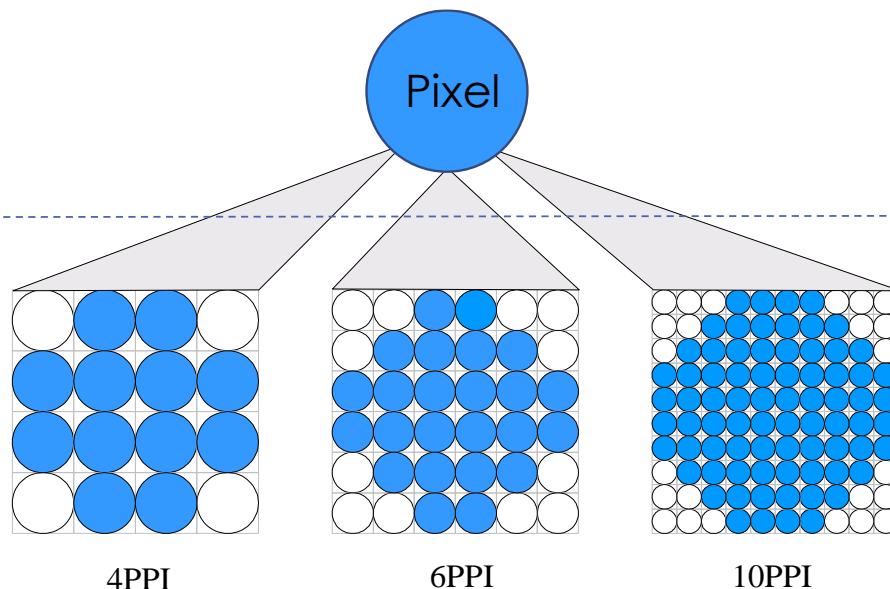
FYZICKÉ VYTVORENIE BODU, ROZLIŠENIE

Na rastrovom zobrazovacom zariadení (napr. na monitore) je pixel vytvorený zobrazením (rozsvietením) istej množiny fyzických bodov výstupnej jednotky zariadenia.

Jemnosť (narastajúca hustota a veľkosť) fyzických bodov ovplyvňuje kvalitu zobrazenia. Udáva sa podľa zariadenia v *PPI* (*Pixel Per Inch*) alebo *DPI* (*Dot Per Inch*)

Pixel na logickej
atomárnej úrovni

Fyzicky reprezentovaný
pixel na výstupnom
rastrovom zariadení



SLED BODOV - POLYMARKER

Je rozširujúcim prvkom a priamo nadväzuje na bod.

Polymarker definuje logicky zviazanú *množinu* bodov na základe určitej *relácie* medzi prislúchajúcimi atribútmi týchto bodov. Pri akejkolvek operácii (napr. presúvaní) sa táto operácia deje nad všetkými bodmi spolu.

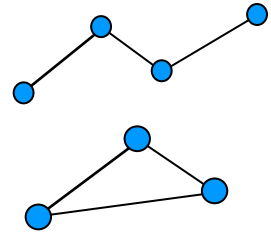
Relácia medzi atribútmi bodov polymarkra môže byť:

- **homogénna** - je relácia definovaná medzi rovnakými atribútmi jednotlivých bodov polymarkra.
Homogénna relácia, napr. medzi atribútmi polohy.
- **heterogénna** - je definovaná medzi rôznymi atribútmi jednotlivých bodov polymarkra, napr. farba niektorého bodu je závislá od polohy druhého.

KRIVKA, LOMENÁ ČIARA

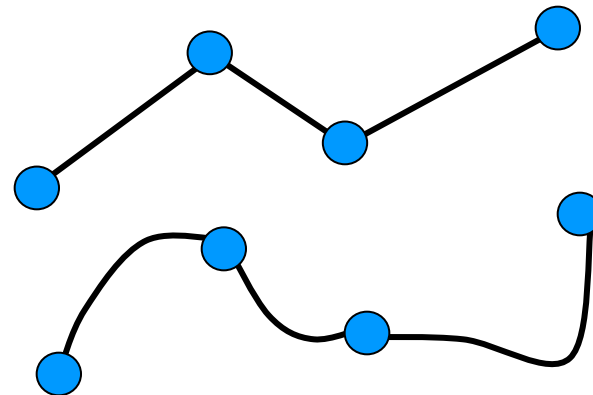
1D útvar definovaný vrcholmi a hranami:

- neuzavretý (acyklický) – krivka, polyline
- uzavretý (cyklický) – plocha, polygón



Podľa typov hrán (segmentov) krivky/čiary:

- lineárny
- nelineárny

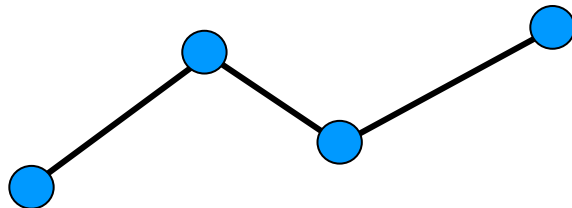


KRIVKA, LOMENÁ ČIARA

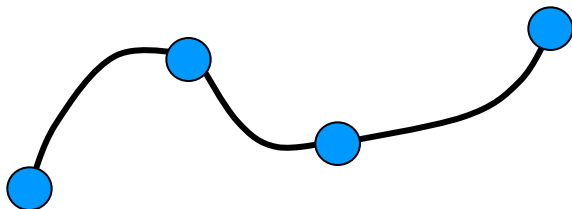
Podľa vplyvu vrcholov na tvar krivky/čiary:

- **Interpolačné** (vrcholy sú súčasťou krivky), typovo: lineárna a nelineárna.
- **Aproximačné** (vrcholy nemusia byť súčasťou krivky, ale vplývajú na jej tvar), typovo: lineárna a nelineárna.

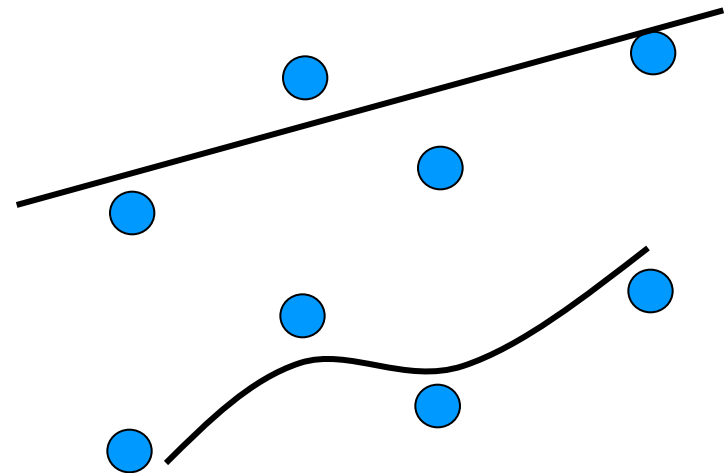
lineárna



nelineárna



interpolácia



aproximácia

ZÁKLADNÉ SPRACOVÁVANÉ GRAFICKÉ PRIMITÍVA

- Úsečka (priamka)
- Kružnica
- Elipsa
- Plocha

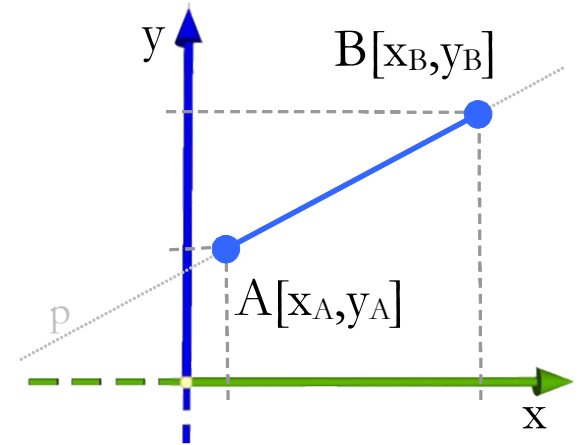


ÚSEČKA

$$y = k \cdot x + c$$

kde:

- y - súradnica bodu na osi y
- x - súradnica bodu na osi x
- k - smernica priamky, na ktorej leží úsečka
- c - posun na osi y



Ak máme dva body $A[x_A, y_A]$ a $B[x_B, y_B]$, ktoré sú koncovými bodmi úsečky, potom základné koeficienty vypočítame nasledovne:

$$k = \frac{y_B - y_A}{x_B - x_A} \quad \text{a} \quad c = \frac{(x_B y_A - x_A y_B)}{(x_B - x_A)}$$

ALGORITMY KRESLENIA ÚSEČKY (RASTROVEJ NÁHRADY)

- Algoritmus založený na výpočte oboch súradníc
- DDA - digital differential analyzer
- Bresenhamov algoritmus



DDA ALGORITMUS

prírastkový algoritmus

(DDA - digital differential analyzer)

založený na postupnom pripočítavaní konštantných prírastkov k obom súradniciam x a y .

Rozlišujeme výpočet pre priamku:

- so smernicou menšou ako 1
- so smernicou väčšou ako 1

DDA ALGORITMUS

$$k = \frac{y_B - y_A}{x_B - x_A} \quad \text{a} \quad c = \frac{(x_B y_A - x_A y_B)}{(x_B - x_A)}$$

$$\begin{aligned} dx &= |x_B - x_A| \\ dy &= |y_B - y_A| \end{aligned} \Rightarrow \text{pocet_krokov} = \max(dx, dy) \Rightarrow \begin{aligned} px &= \frac{dx}{\text{pocet_krokov}} \\ py &= \frac{dy}{\text{pocet_krokov}} \end{aligned}$$



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

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



BRESENHAMOV ALGORITMUS

veľmi efektívny algoritmus generovania bodov na úsečke. Nachádza body ležiace najbližšie danej skutočnej úsečke na základe hodnoty predikčného chybového člena E_D

Rozlišujeme výpočet pre priamku:

- so smernicou menšou ako 1
- so smernicou väčšou ako 1



Dr. Jack Elton Bresenham

BRESENHAMOV ALGORITMUS

$$dx = |x_B - x_A|$$

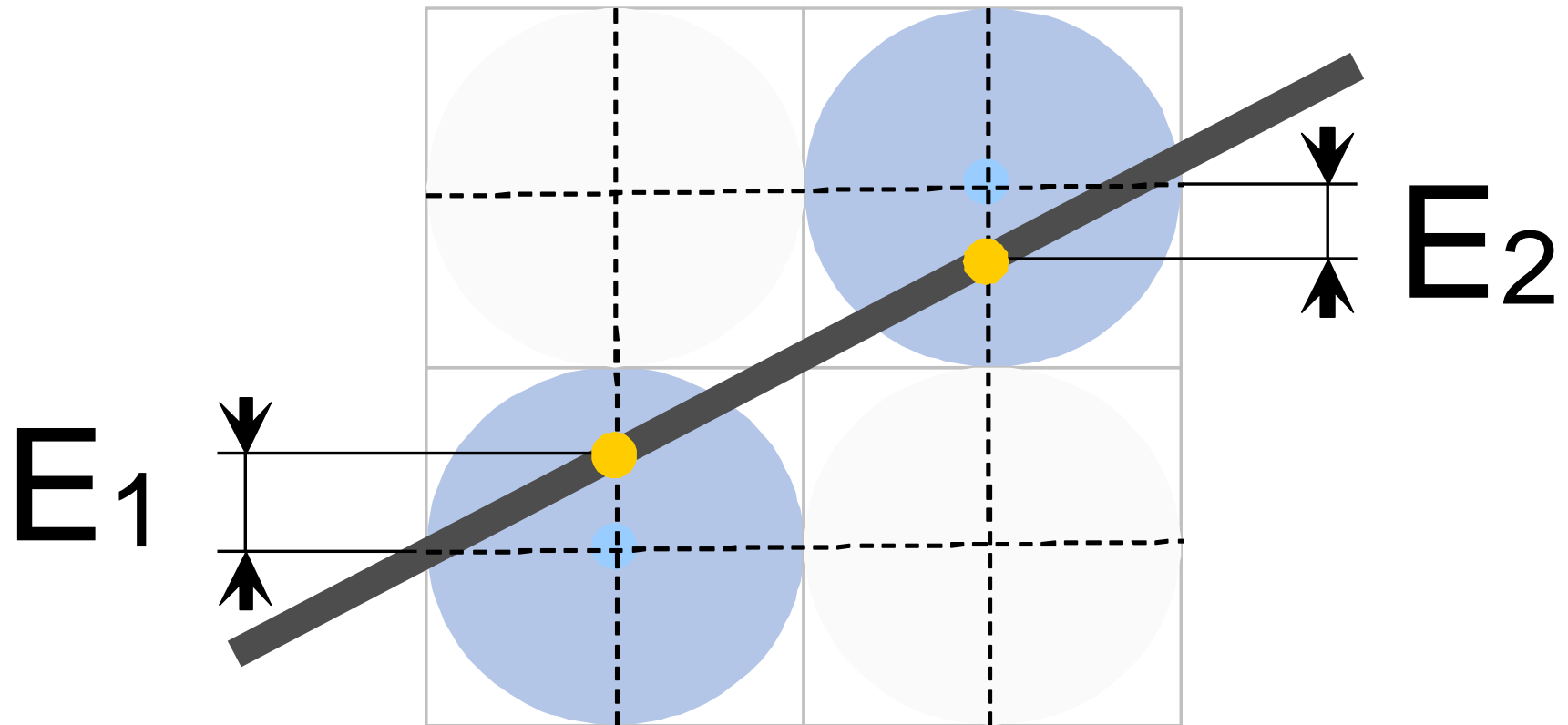
$$dy = |y_B - y_A|$$

$$E_1 = 2 \cdot dx - dy \quad \text{pre prvý bod}$$

$$E_i < 0 \Rightarrow E_{i+1} = E_i + 2 \cdot dy ; y_{i+1} = y_i$$

$$E_i \geq 0 \Rightarrow E_{i+1} = E_i + 2 \cdot dy - 2 \cdot dx ; y_{i+1} = y_i + 1$$

BRESENHAMOV ALGORITMUS

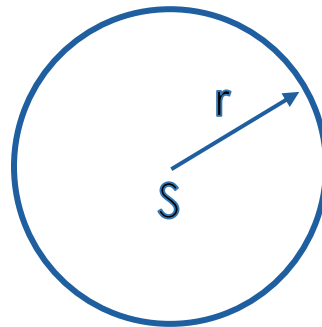


Stanovenie chybového člena E_D pri Bresenhamovom algoritme

KRUŽNICA

množina bodov rovnako vzdialených od
stredového bodu S.

$$y = y_s \pm \sqrt{r^2 - (x - x_s)^2}$$



KRUŽNICA

- Algoritmus kreslenia kružnice na základe parametrického vyjadrenia

$$x = x_s + r \cdot \cos(u)$$

$$y = y_s + r \cdot \sin(u)$$

- Algoritmus kreslenia kružnice podľa predikcie chyby

$$x_1 = 0$$

Poloha:

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

$$y_1 = r$$

y_{i+1} sa stanoví predikciou chyby E_{i+1}

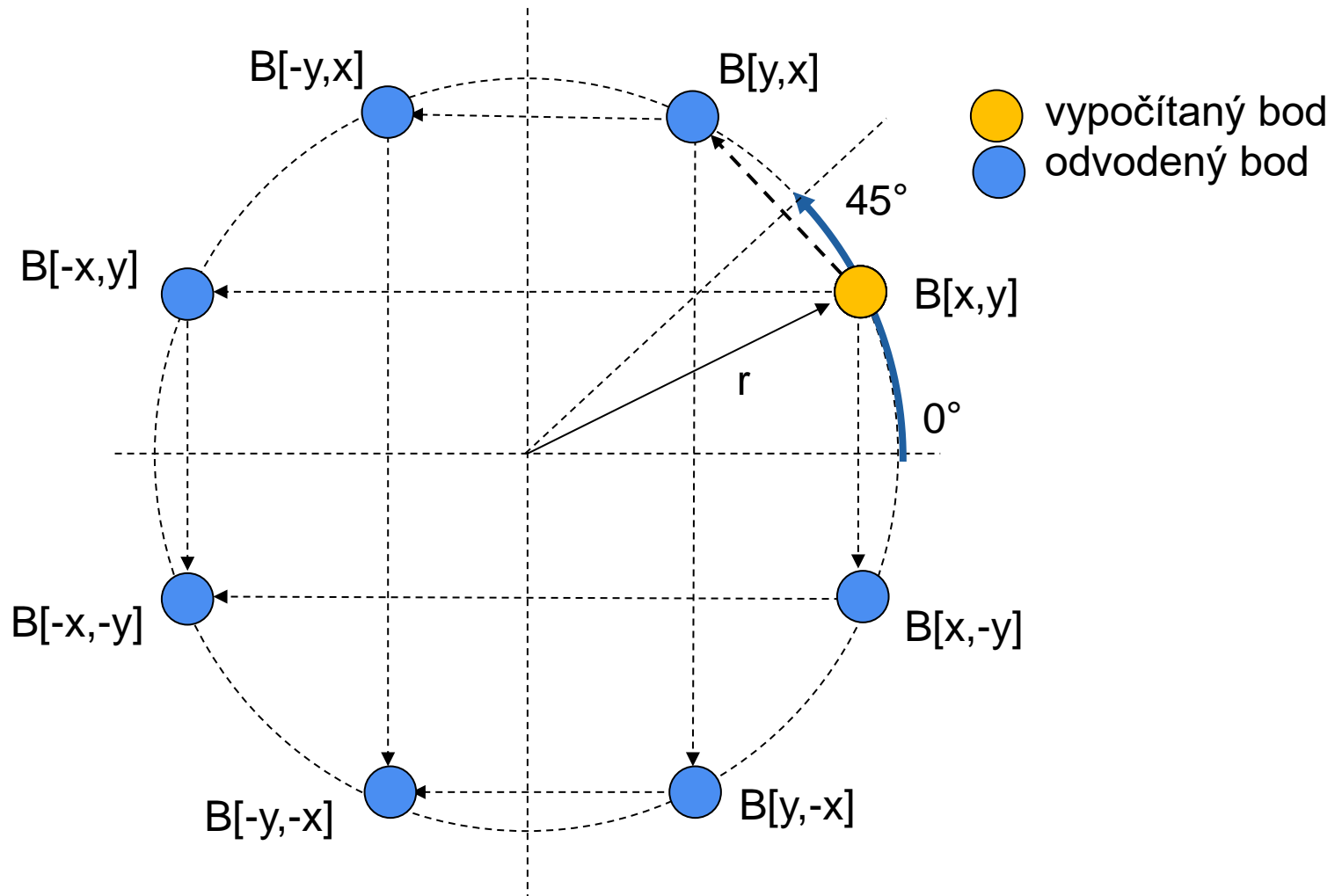
Počiatočná predikcia chyby:

$$E_1 = 1 - r$$

$$E_i < 0 \Rightarrow E_{i+1} = E_i + 2 \cdot x_i + 3$$

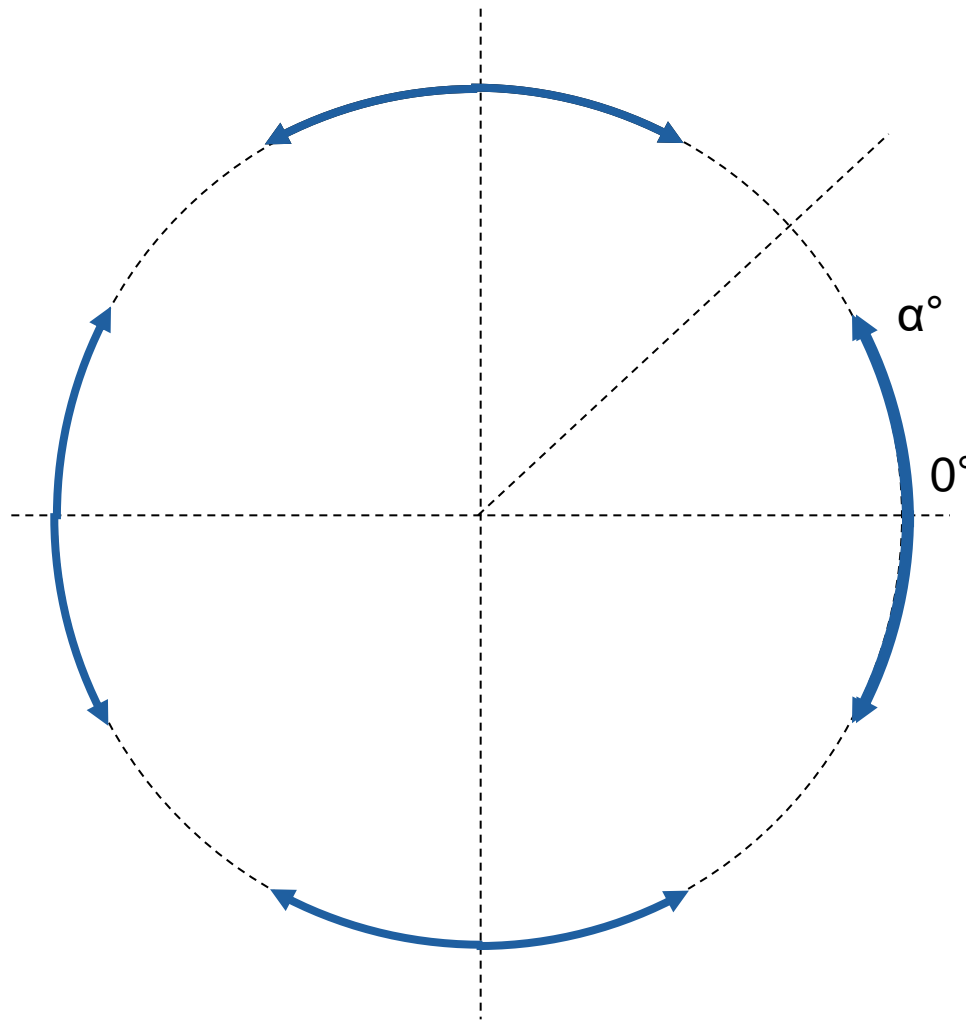
$$E_i \geq 0 \Rightarrow E_{i+1} = E_i + 2 \cdot x_i + 5 - 2 \cdot y_i$$

KRUŽNICA



Kreslenie kružnice využitím osovej súmernosti

KRUŽNICA

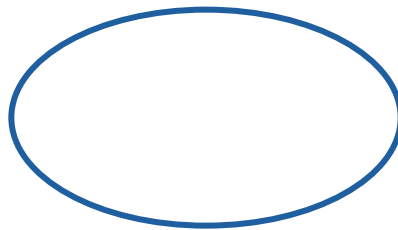


Kreslenie kružnice využitím osovej súmernosti

ELIPSA

množina bodov rovnako vzdialených od dvoch ohnísk

$$\frac{(x - x_s)^2}{a^2} + \frac{(y - y_s)^2}{b^2} = 1$$



ELIPSA

- Algoritmus kreslenia elipsy na základe parametrického vyjadrenia

$$x = x_s + a \cdot \cos(u)$$

$$y = y_s + b \cdot \sin(u)$$

- Algoritmus kreslenia elipsy podľa predikcie chyby

$$x_1 = 0$$

Poloha:

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

$$y_1 = r$$

y_{i+1} sa stanoví predikciou chyby E_{i+1}

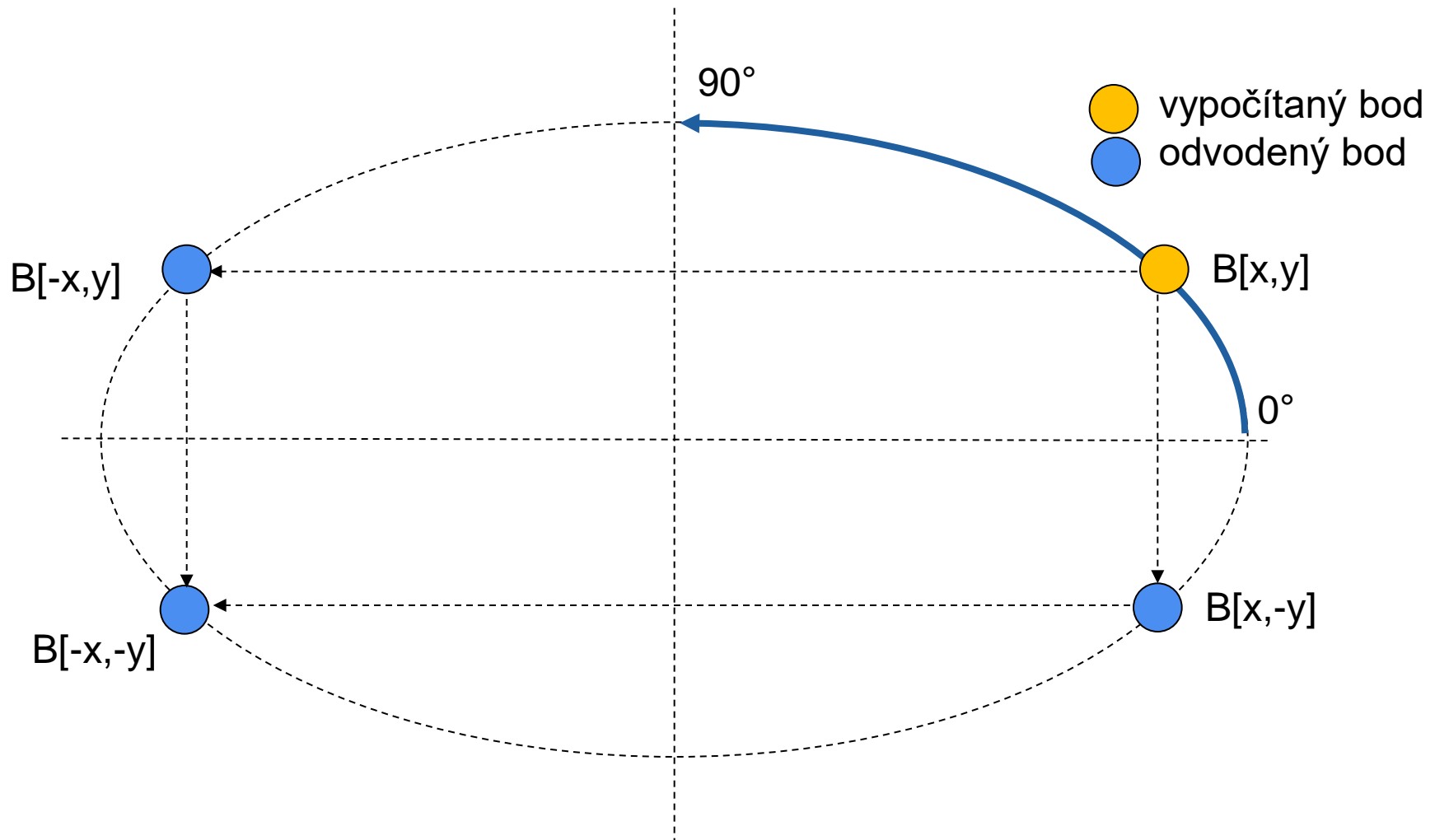
Počiatočná predikcia chyby:

$$E_1 = b^2 - b \cdot a^2 + \frac{a^2}{4}$$

$$E_i < 0 \Rightarrow E_{i+1} = E_i + b^2 \cdot (2 \cdot x_i + 1)$$

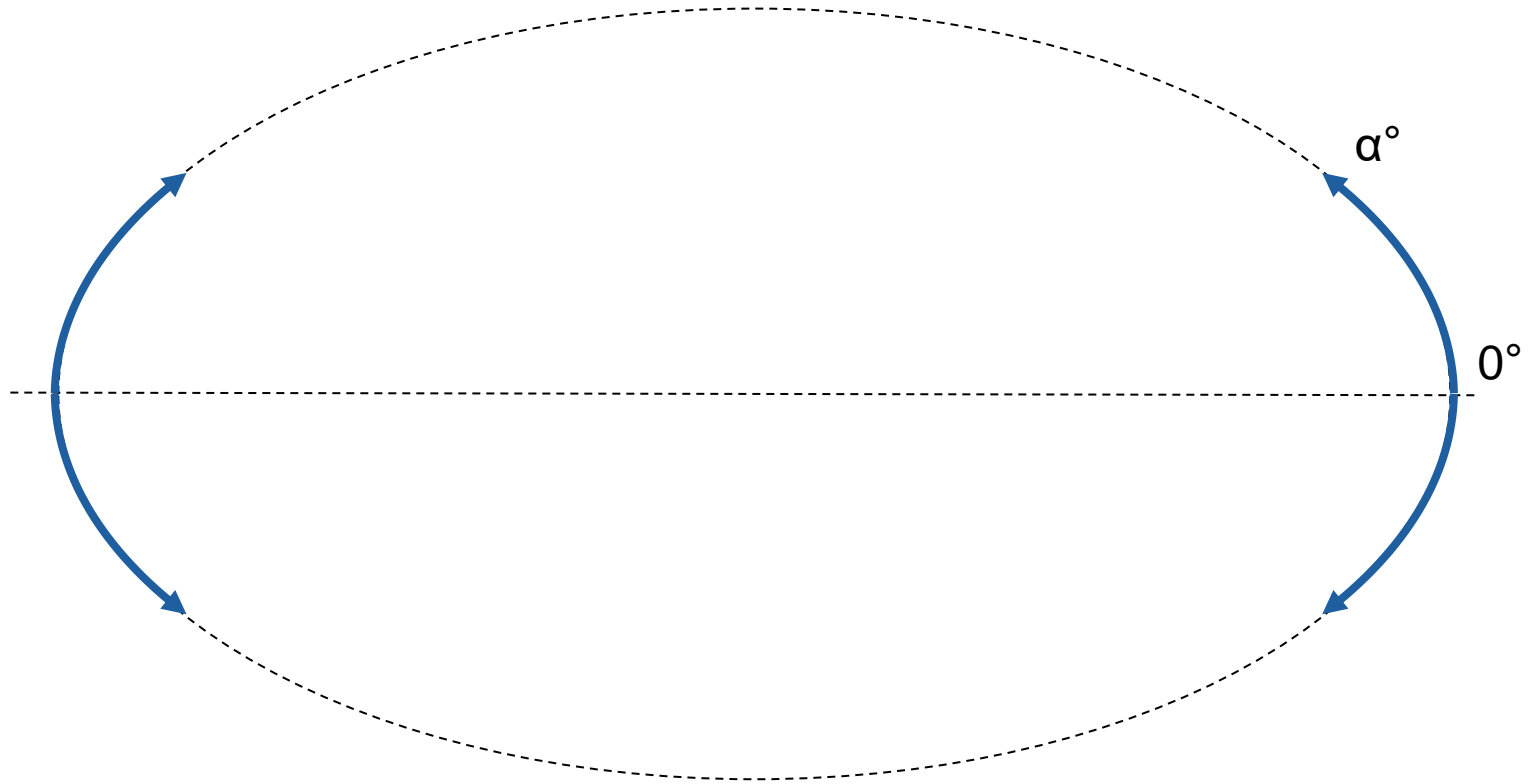
$$E_i \geq 0 \Rightarrow E_{i+1} = E_i + b^2 \cdot (2 \cdot x_i + 1) - 2 \cdot a^2 \cdot y_i$$

ELIPSA



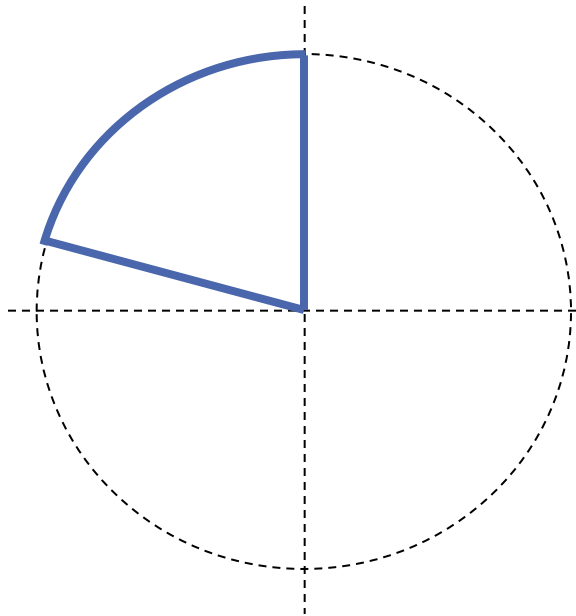
Kreslenie elipsy využitím osovej súmernosti

KRUŽNICA

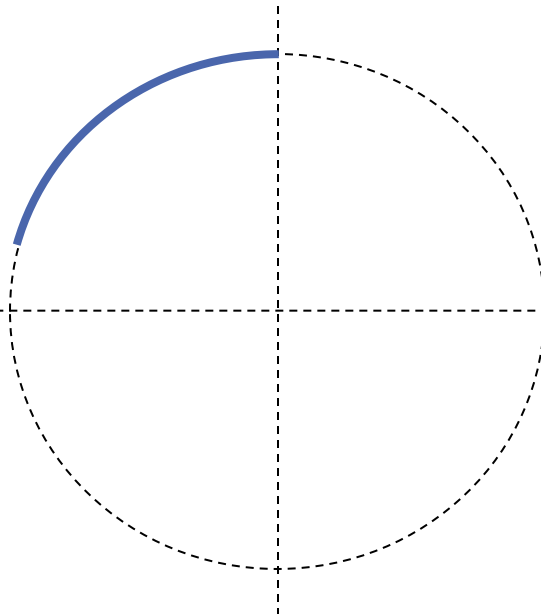


Kreslenie elipsy využitím osovej súmernosti

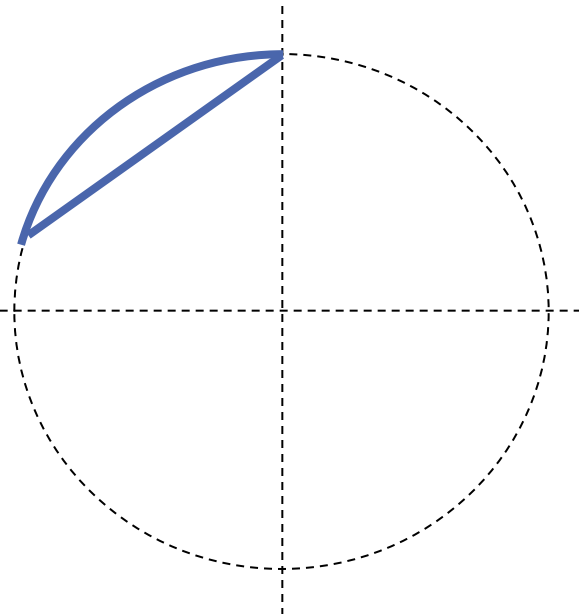
KRUHOVÉ (ELIPTICKÉ) OBLASTI



výsek

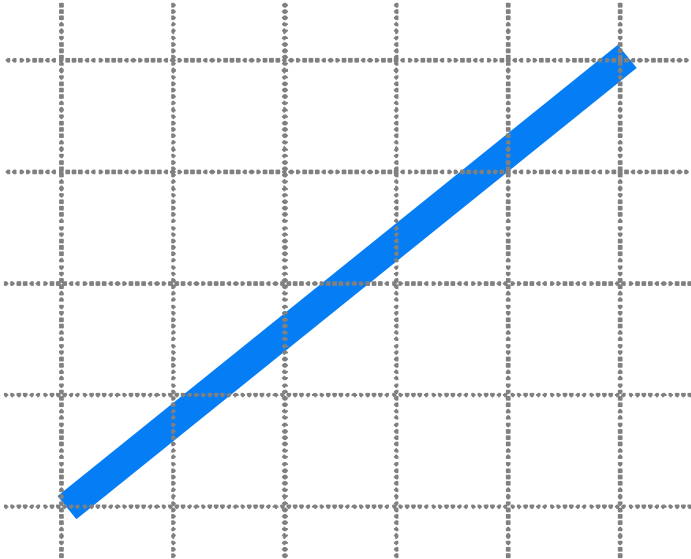


oblúk



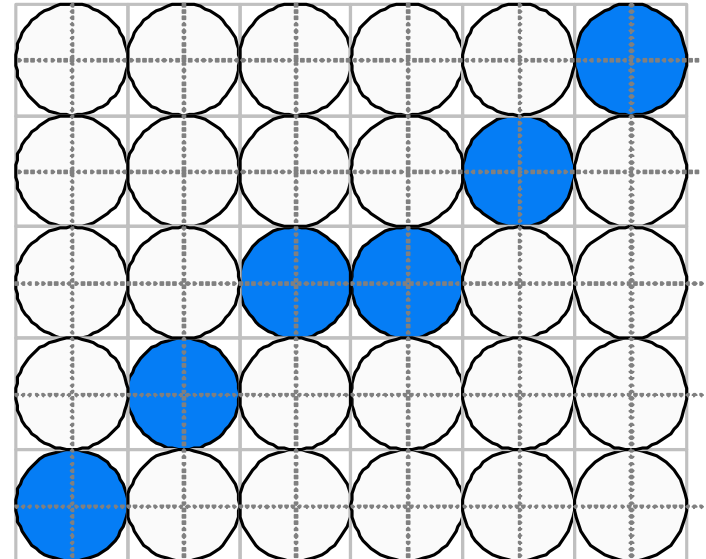
odsek

ANTIALIASING



úsečka

a jej



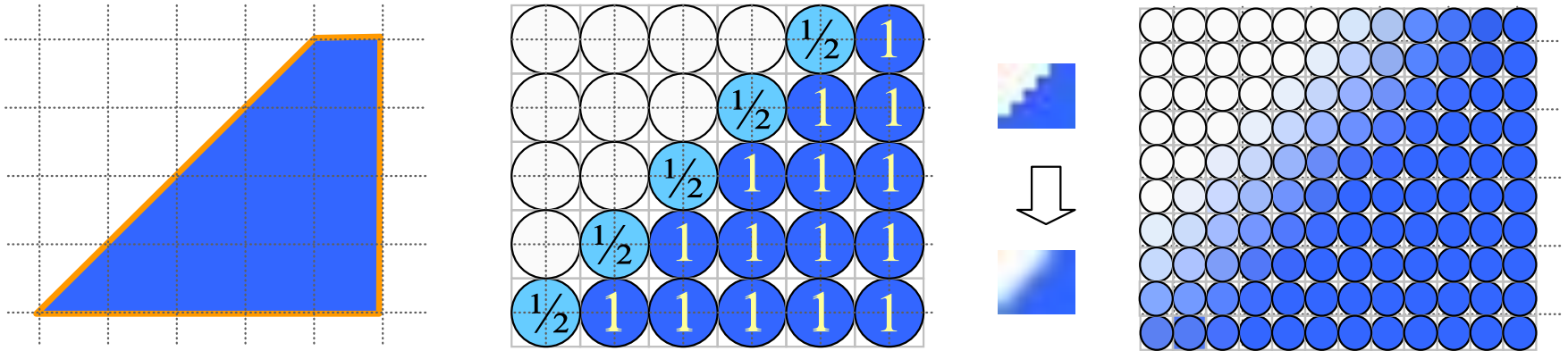
rastrová náhrada
(alias)

ANTIALIASING

Prípady:

- schodovité zobrazovanie priamych rovných čiar a hraníc polygónov na rastrových displejoch.
- zobrazovaný objekt menší ako veľkosť pixelu (obrazového bodu) alebo pri veľmi tenkých čiarach. Malé objekty potom nie sú vôbec zobrazované alebo napr. tenké čiary nie sú opticky hladké resp. ucelené (sú zobrazované ako nepravidelná postupnosť bodov).
- pri zobrazovaní zložitejšej scény s blízkymi detailmi (napr. pri generovaní základných obrázkov z raytracingu). Tieto detaily sú buď potlačené alebo skreslené tak, že nie je možné rozoznať ich pôvodný tvar. (LOD – Level of Detail)

ANTIALIASING



Často sa používajú namiesto *antialiasingu* aj niektoré rozptyľovacie metódy.

FILTROVACIE A ROZPTYĽOVACIE METÓDY

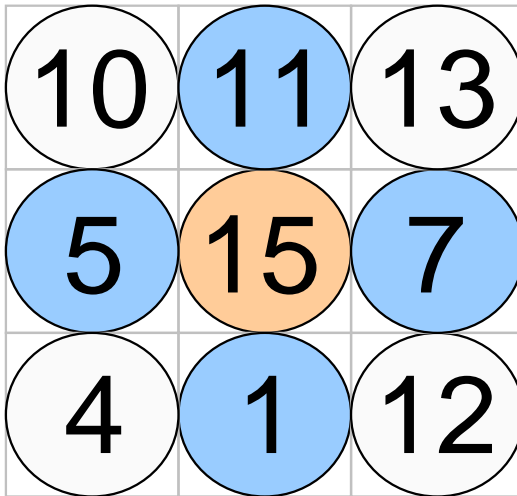
Určenie farby:

- náhodne
- *spriemernením* bodov oblasti alebo použitím *mediánovej* funkcie príp. inej funkcie.
- zistí sa početnosť výskytu farieb v oblasti a vyberie sa tá farba, ktorá sa vyskytuje najčastejšie. Ak je výskyt farieb rovnaký, vyberie sa farba podľa iného pravidla alebo ľubovoľná z vyskytujúcich sa farieb.
- vyberie sa farba, ktorá sa vyskytuje najmenej krát.
- vyberie sa farba ľavého horného bodu oblasti

FILTROVACIE A ROZPTYĽOVACIE METÓDY

SPRIEMERNENIE

4-susedstvo



(11,5,15,7,1)

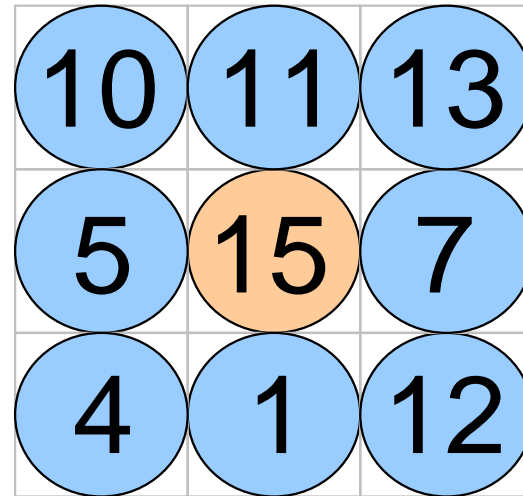


$(11+5+15+7+1=39)/5$



8

8-susedstvo



(10,11,13,5,15,7,4,1,12)



$(10+11+13+5+15+7+4+1+12=78)/9$

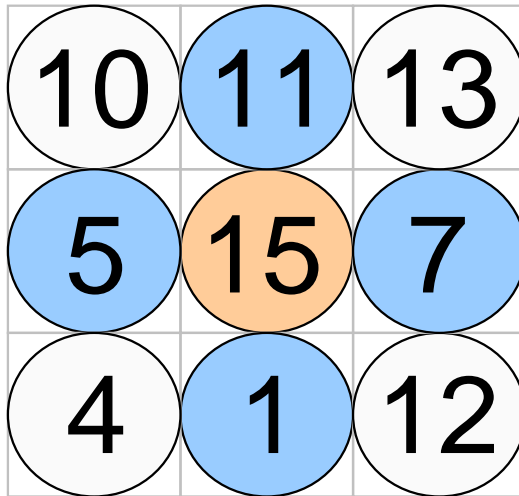


9

FILTROVACIE A ROZPTYĽOVACIE METÓDY

MEDIÁNOVÁ FUNKCIA (MEDIÁN FILTER)

4-susedstvo



(11, 5, 15, 7, 1)

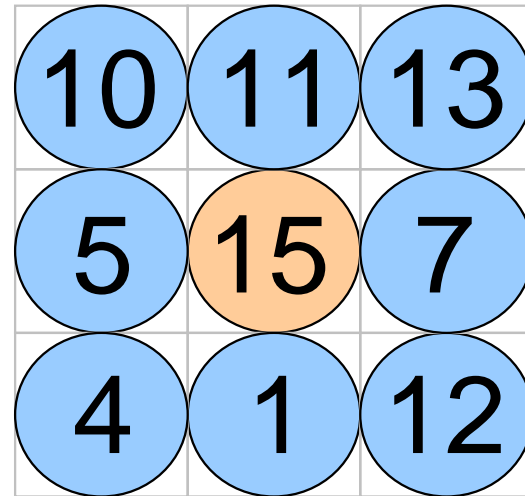


(1, 5, 7, 11, 15)



7

8-susedstvo



(10, 11, 13, 5, 15, 7, 4, 1, 12)



(1, 4, 5, 7, 10, 11, 12, 13, 15)



10

ROZPTYĽOVACIE METÓDY

Metódy na riešenie problému rozdielu medzi požadovanou kvalitou výsledných obrazov a obmedzenými možnosťami dostupného farebného priestoru zariadenia alebo techniky pre podporu *antialiasingu*



ROZPTYŤOVACIE METÓDY

Rozptyľovacie metódy využívajú tú vlastnosť ľudského oka, že z farieb niekoľkých blízkych bodov vytvára dojem jediného bodu, ktorého farba je daná aditívnym zlúčením farieb pôvodných bodov (napríklad pre prevod obrazu so 16 úrovňami šedej na binárny obraz sa použije intenzita jasů vstupného bodu $I_{IN} \in \langle 0, 15 \rangle$ a výstupného bodu $I_{OUT} \in \langle 0, 1 \rangle$).

- Náhodné rozptyľovanie
- Maticové rozptyľovanie
- Rozptyľovanie distribúciou chyby

0	0	0	0	0	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1
$I_{IN} = 0$		$I_{IN} = 1$		$I_{IN} = 2$		$I_{IN} = 3$		$I_{IN} = 4$	

Príklady rozptyľovacích matíc $I_{IN} = \langle 0, 4 \rangle$

0	11	3	14
8	4	10	7
2	13	1	2
9	6	8	5

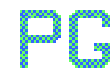
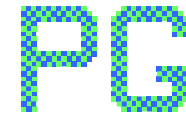
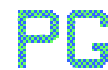
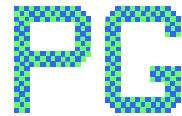
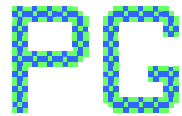
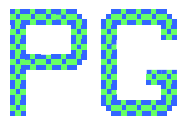
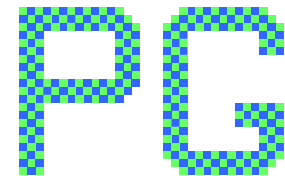
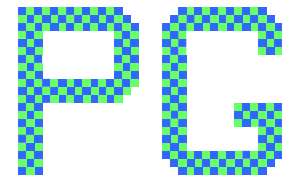
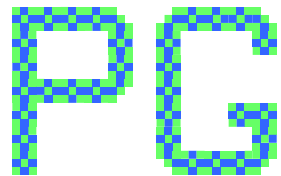
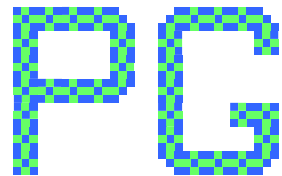
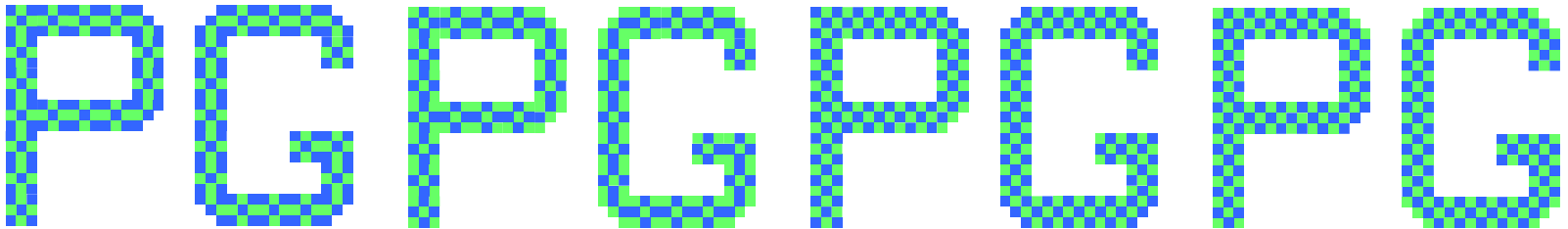
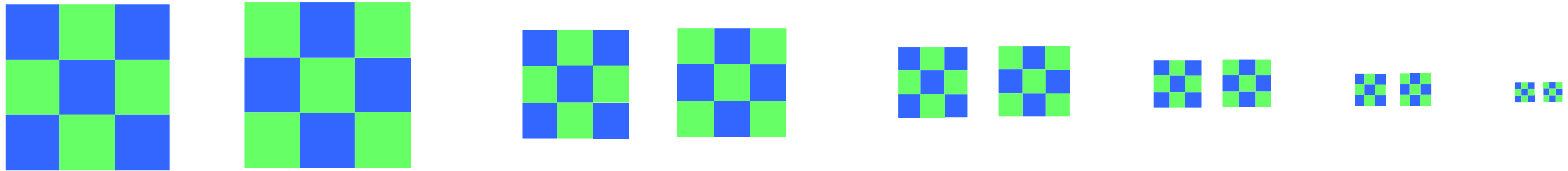
$I_{IN} = \langle 0, 15 \rangle$

0	12	3	15
8	4	11	7
2	14	1	13
10	6	9	5

$I_{IN} = \langle 0, 16 \rangle$

ROZPTYĽOVACIE METÓDY

DITHERING (DVE A VIAC FARIEB)



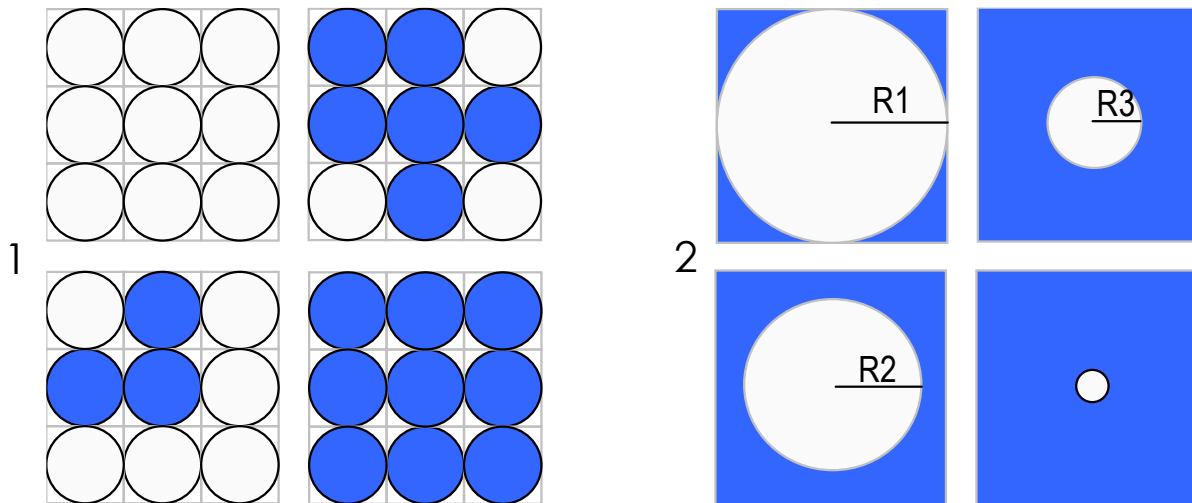
ROZPTYĽOVACIE METÓDY

POLTÓNOVANIE - HALFTONING

Viacero spôsobov ako prevádzať na dve farby. Dôležité je určiť binarizačný prah (treshhold).

Najpoužívanéjšie spôsoby:

1. *maticové* - využitím binárnych nahradzovacích matic pre spracovávanú oblasť
2. *funkcionálne* – využitím funkcií k pomerovému vyplneniu spracovávanej oblasti

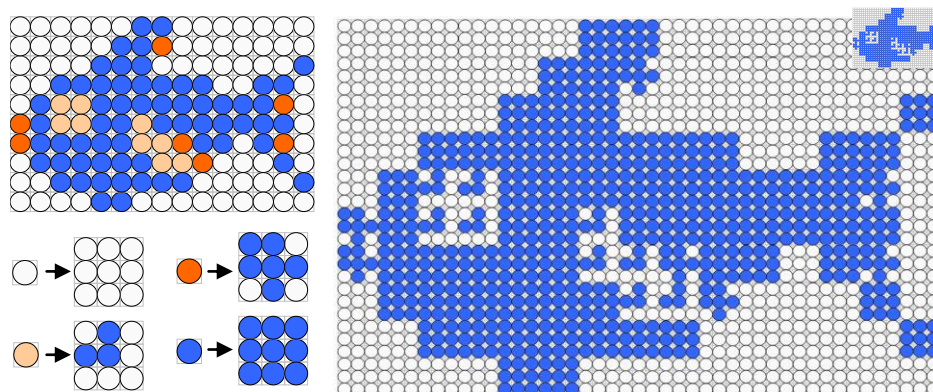
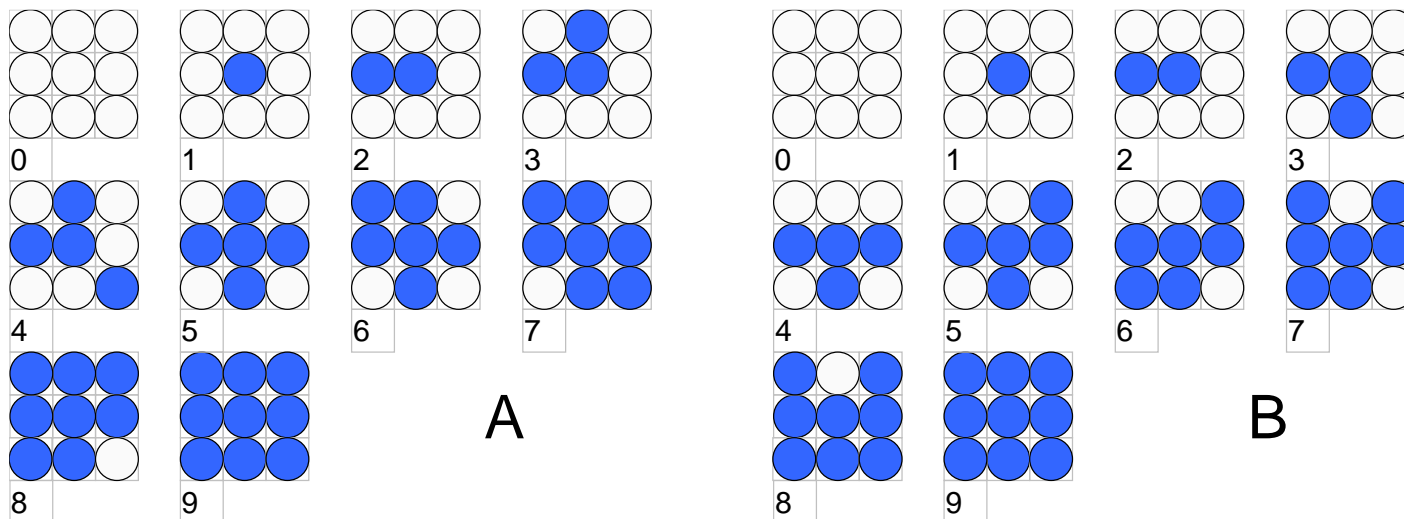


Príklady rôznych spôsobov poltónovania

ROZPTYĽOVACIE METÓDY

POLTÓNOVANIE - HALFTONING

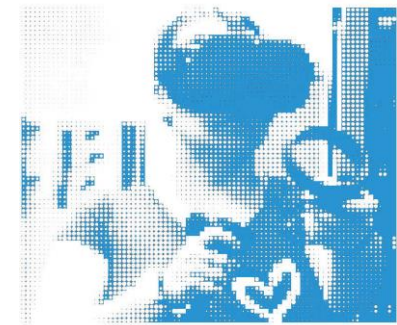
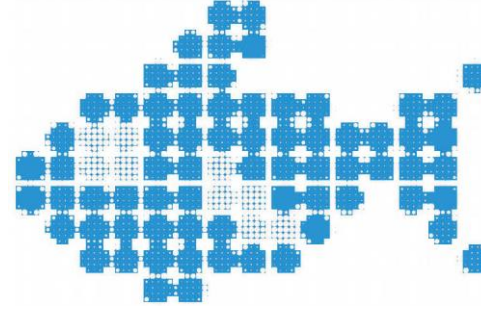
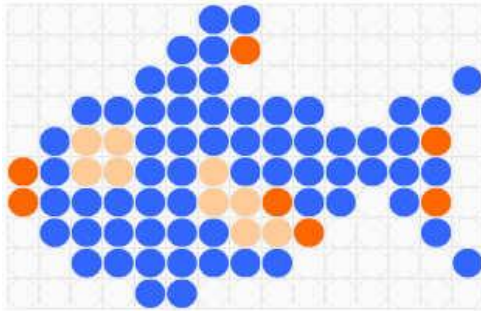
- Rôzne nahradzovacie matice:



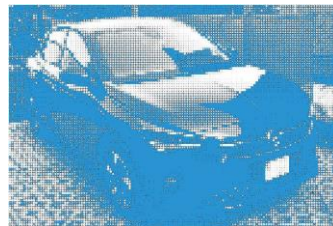
ROZPTYŤOVACIE METÓDY

POLTÓNOVANIE - HALFTONING

- Príklady funkcionálneho poltónovania (rôzne parametre):



a) →



b) →

ROZPTYŤOVACIE METÓDY

NA ZÁKLADE DISTRIBÚCIE CHYBY

1. distribúcia chyby - Floyd-Steinberg (koef./16)
2. distribúcia chyby – Stucki (koef./42)
3. distribúcia chyby – Burkes (koef./32)
4. distribúcia chyby – Sierra (koef./32)
5. distribúcia chyby - Jarvis, Judice, Ninke (koef./48)
6. distribúcia chyby - Stevenson, Arce (koef./200)
7. ...

	x	7
3	5	1

1

		x	8	4
2	4	8	4	2
1	2	4	2	1

2

		x	8	4
2	4	8	4	2

3

		x	5	3
2	4	5	4	2
	2	3	2	

4

		x	7	5
3	5	7	5	3
1	3	5	3	1

5

ROZPTYŤOVACIE METÓDY

DISTRIBÚCIA CHYBY - FLOYD-STEINBERG (KOEf./16)

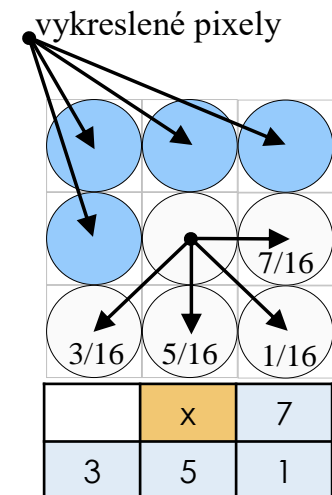
Je to veľmi rýchla metóda, pretože pri nej dochádza k deleniu jednotlivých koeficientov šestnástimi, čo je možné veľmi jednoducho implementovať bitovou rotáciou.

Intenzita vstupného prvku obrazu (bodu) je na výstupe postupne nahradzovaná jednotkou alebo nulou, Vstupná intenzita I_{IN} je použitá nielen k nájdeniu najbližšej zodpovedajúcej intenzity I_{OUT} , ale zanedbaná hodnota, vznikajúca zaokrúhlením I_{IN} z väčšieho rozsahu do rozsahu $<0, 1>$ je využitá na modifikáciu hodnôt susedných prvkov obrazu (Ak má byť napríklad spracovaný prvok obrazu so vstupnou hodnotou 4, výstupnému prvku obrazu sa priradí hodnota 0 a zanedbaná hodnota 4 sa pripočíta k hodnotám susedných prvkov obrazu. Distribuovaná chyba je rozdelená medzi susedné prvky obrazu v závislosti od použitej metódy).

Pred vykreslením každého prvku obrazu je ku vstupnej hodnote I_{IN} pripočítaná doteraz získaná chyba (z pamäte chyby) a až potom je výsledná hodnota prevedená do výstupného rozsahu. Zanedbaná hodnota sa opäť rozdelí a pripočíta do pamäte chyby (nutné zabezpečiť, aby nedochádzalo k strate celočíselným delením).

Postup pre použitie rozptyľovacích metód pre farebné obrazy je nasledujúci:

1. Pre daný obraz sa vytvorí paleta podľa požiadaviek niektorou z metód popísaných vyššie a algoritmus sa nastaví na prvý bod obrazu.
2. Vyberie sa bod a nájde sa k nemu najbližšia farebná hodnota vo vytvorenej palete.
3. Pre daný bod sa určí zanedbaná hodnota, ktorá vznikne zaokrúhlením pôvodnej farby na farbu z palety.
4. Na základe zanedbanej hodnoty a použitej metódy sa rozhodne o prípadnom zvýšení výslednej intenzity farby výsledného bodu.
5. Ak ešte nie je spracovaný celý obraz, posunie sa na ďalší bod a pokračuje sa ďalej krokom 2.



Q & A

branislav.sobota@tuke.sk

Katedra počítačov a informatiky, FEI TU v Košiciach

© 2024