

Otázky z predmetu Počítačová grafika ZS šk. rok 2021/22

1. Charakterizujte problém používania a spracovania farieb v rámci počítačovej grafiky, základné atribúty svetla, chromatický diagram.

SVETLO

Dve reprezentácie svetla:

- vlnová - elektromagnetické vlnenie v oblasti 10¹⁴ Hz.
- časticová – prúd častíc – fotónov

Typy svetla:

- Achromatické - viacfarebné, celé spektrum, „biele svetlo“
- Monochromatické - 1 zložka prevláda (100%) - prakticky nevieme urobiť 100% 1f svetlo

ZÁKLADNÉ ATRIBÚTY SVETLA

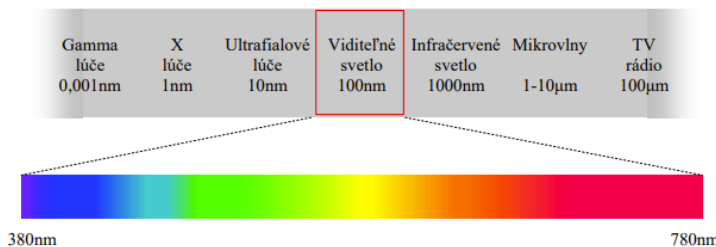
- Farba – je základným atribútom svetla a závisí od už spomínaných frekvencií (resp. vlnovej dĺžky)
- Jas - odpovedá vlastne intenzite svetla. Jasnosť zdroja svetla je v priamej úmere s intenzitou (amplitúdou)
- Sýtosť - uvádza čistotu svetla. Čím vyššia je sýtosť, tým užšie je spektrum frekvencií obsiahnutých vo svetle
- Svetlosť – je veľkosť achromatickej zložky vo svetle s určitou dominantnou frekvenciou.

FARBY

Každá farba zodpovedá určitej frekvencii vlnenia.

Rozsah farieb je:

- od červenej (7.8x10¹⁴Hz, mimo viditeľného spektra pokračuje do infračervenej oblasti)
- po fialovú (3.8x10¹⁴Hz, mimo viditeľného spektra pokračuje do ultrafialovej oblasti)



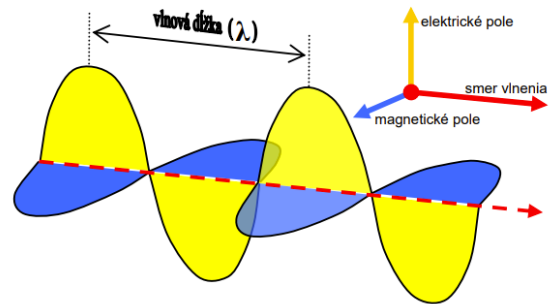
FAREBNÉ MODELY sú definované (v rámci farebnej dimenzie):

- množinou základných farieb,
- spôsobom ich miešania a
- pravidlami menenia farebných charakteristík

FAREBNÝ PRIESTOR - Oblasť farieb pokrytá možnosťami príslušného farebného modelu sa nazýva jeho farebný priestor (color space)

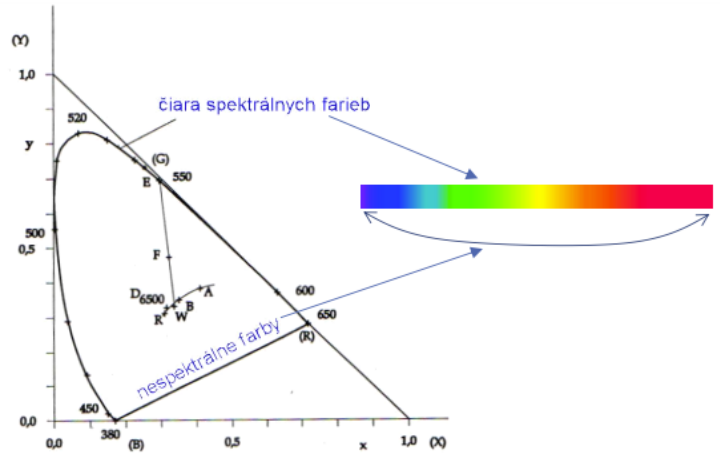
FAREBNÁ HĽBKA

- udáva sa v bitoch, počet zobraziteľných farieb farebného priestoru
- farebné režimy napr. HighColor (16bitov), TrueColor (24, 32 bitov)



CIEXY CHROMATICKÝ DIAGRAM

- farebný model vyvinutý na základe štandardného pozorovateľa, ktorého farebné videnie je zástupcom ľudskej populácie s normálnym farebným videním
- pre určenie farby sa používa trojzložkový (tristimulus) systém. Farba je určená trojicou čísel (X, Y, Z). $y = \text{jas}$
- Tieto trojzložkové hodnoty udávajú množstvo každej z troch hypotetických primárnych zložiek, ktoré definujú farbu. Hodnota Y dáva jas objektu a primárne zložky sú vybrané tak, aby boli viditeľné farby definované ako pozitívne hodnoty.



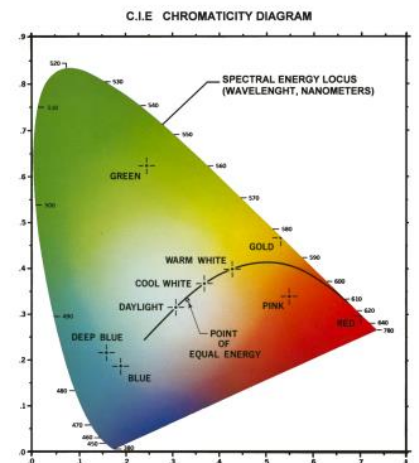
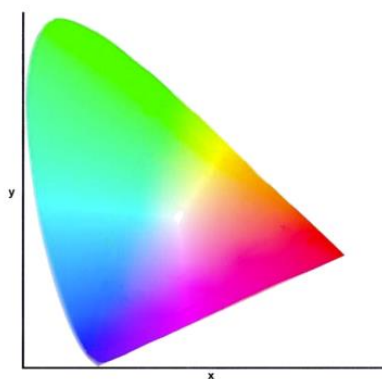
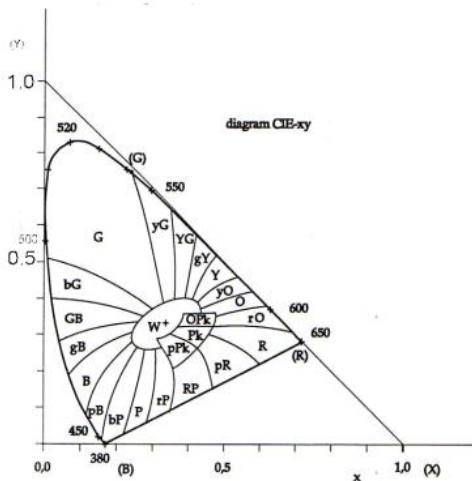
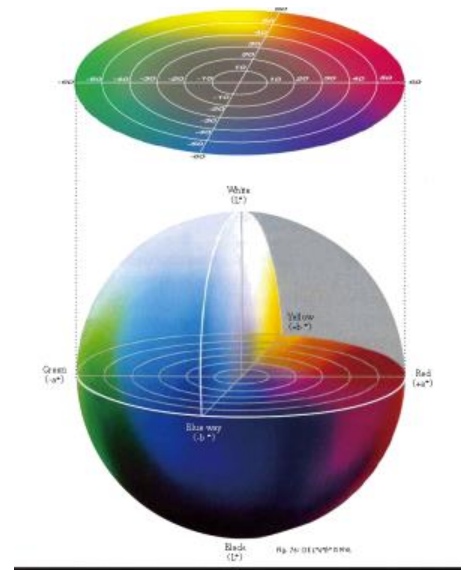
1. Typ: CIELAB CHROMATICKÝ DIAGRAM

- CIELAB – (svetlo *absorbované*; atramenty, farbivá, pigmenty) reprezentuje popis farebného priestoru založený na *subtraktívnom miešaní - tlačiaren*

2. Typ: CIEUV CHROMATICKÝ DIAGRAM

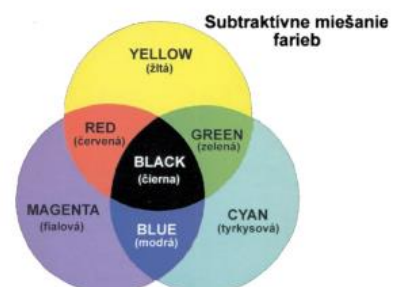
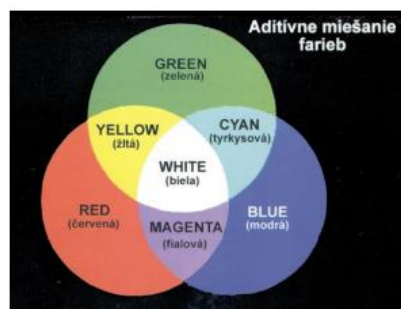
- CIEUV – (svetlo *emitované*; fosfor, farebné svetelné zdroje) reprezentuje popis farebného priestoru založený na *aditívnom miešaní - monitor*

Škála svetlosti (jasnosti) je pre oba modely rovnaká a je založená na tretej odmocnине svetivosti, ktorý dáva lineárna stupnica



MIEŠANIE FARIEB

- Aditívne – základ je čierna, pridaním všetkých zložiek => biela
- Subtraktívne – základ je biela, pridaním všetkých zložiek => čierna

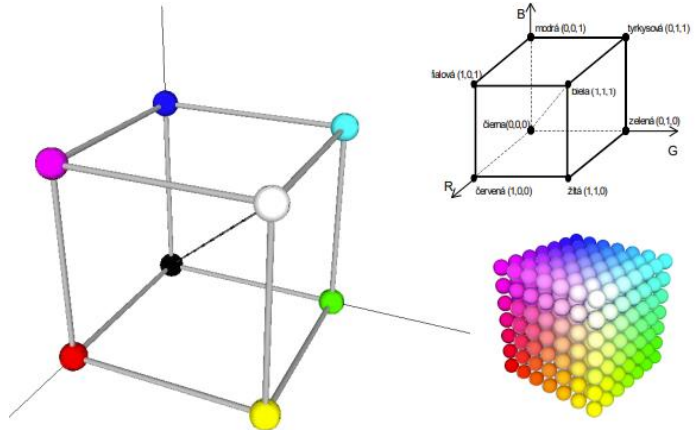


2. Charakterizujte a popíšte model farebný model RGB a RGBA.

FAREBNÝ MODEL RGB

- Zložky: red, green, blue
- Miešanie: aditívne
- Zmena zložky: lineárna

farba \ zložka	R(ed) červená	G(reen) zelená	B(lue) modrá
čierna	0	0	0
modrá	0	0	1
zelená	0	1	0
tyrkysová (azúrová)	0	1	1
červená	1	0	0
fialová (purpurová)	1	0	1
žltá	1	1	0
biela	1	1	1



FAREBNÝ MODEL RGBA

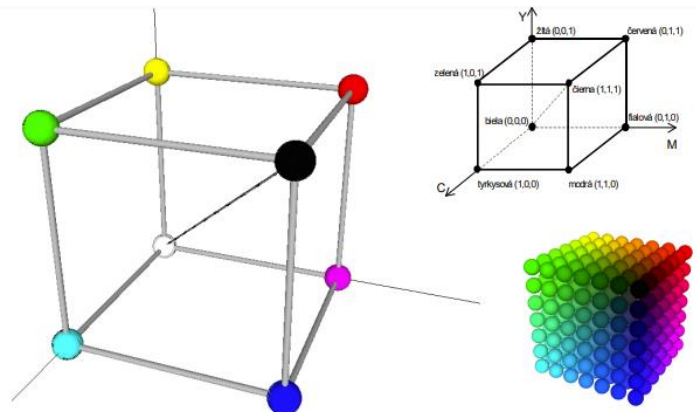
- Zložky: red, green, blue, alfa – priesvitnosť
- Miešanie: aditívne
- Zmena zložky: lineárna

3. Charakterizujte a popíšte model farebný model CMY a CMYK.

FAREBNÝ MODEL CMY

- Zložky: Cyan – tyrkysová (azúrová), Magenta – fialová (purpurová), Yellow – žltá
- Miešanie: subtraktívne
- Zmena zložky: lineárna

farba \ zložka	C(yan) tyrkysová	M(agenta) fialová	Y(ellow) žltá
biela	0	0	0
žltá	0	0	1
fialová (purpurová)	0	1	0
červená	0	1	1
tyrkysová (azúrová)	1	0	0
zelená	1	0	1
modrá	1	1	0
čierna	1	1	1



FAREBNÝ MODEL CMYK

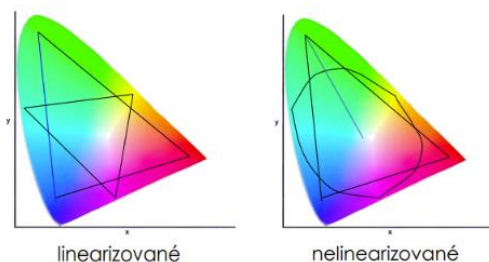
- Zložky: Cyan – tyrkysová (azúrová), Magenta – fialová (purpurová), Yellow – žltá, black – technologická čierna

- Miešanie: subtraktívne
- Zmena zložky: lineárna

PREVOD RGB ↔ CMY

$$\begin{matrix} \text{RGB} \rightarrow \text{CMY} \\ \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \end{matrix} \quad \begin{matrix} \text{CMY} \rightarrow \text{RGB} \\ \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \end{matrix}$$

POROVNANIE RGB A CMY FAREBNÉHO PRIESTORU

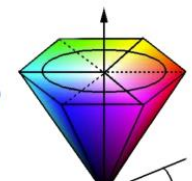
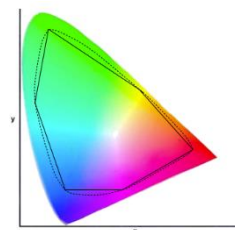
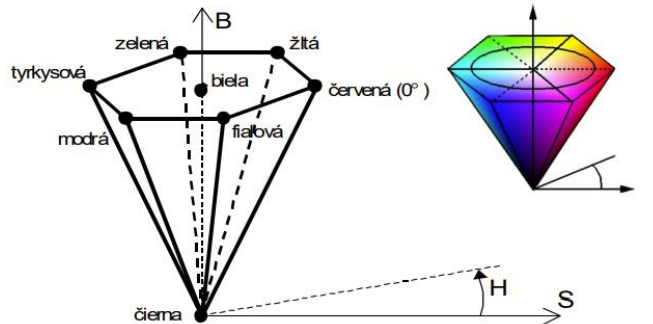
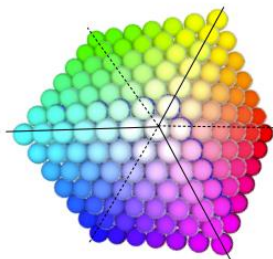
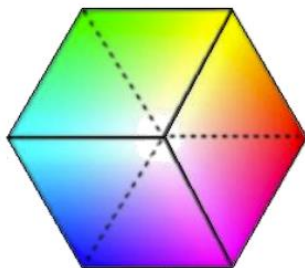


RGB vs. CMY - pokrývajú inú časť farebného priestoru - nie všetky farby sú dosiahnuteľné oboma modelmi

4. Charakterizujte a popíšte model farebný model HSB.

FAREBNÝ MODEL HSB (HSV)

- Zložky: Hue – farebný tón, Saturation – saturácia, Brightness – jas (value)
- Miešanie: aditívne
- Zmena zložky: uhlová a lineárna



RGB priemet do roviny kolmej k šedej diagonále (0,0,0) - (1,1,1)

PREVOD RGB ↔ HSB (HSV)

RGB → HSB (HSV)

1. $\max = \max(R, G, B)$
2. $\min = \min(R, G, B)$
3. $\delta = \max - \min$
4. $V = \max$
5. ak $\max = 0$ $S = 0$ inak $S = (\max - \min) / \max$
6. ak prevláda zložka R: $H = (G - B) / \delta$
7. ak prevláda zložka G: $H = 2 + (B - R) / \delta$
8. ak prevláda zložka B: $H = 4 + (R - G) / \delta$
9. $H = H * 60$, ak H je záporné číslo: $H = H + 360$

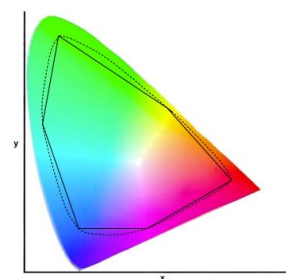
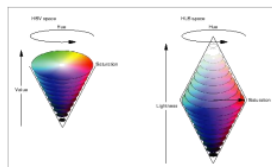
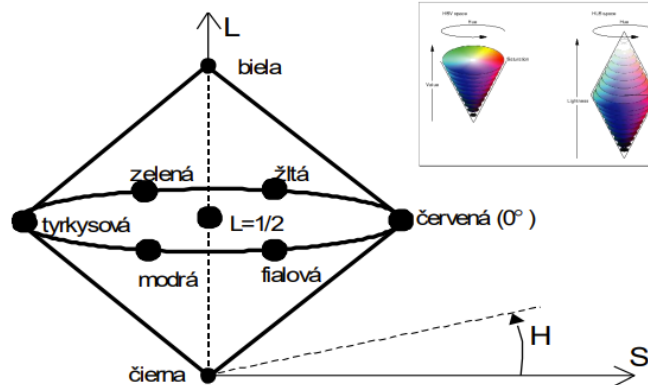
HSB (HSV) → RGB

1. ak $S = 0$, $R = V$, $G = V$, $B = V$, koniec
2. inak $H = H / 60$
3. $i = \text{celá časť } H$, $f = \text{zvyšok po delení } H$
4. $p = V * (1 - S)$, $q = V * (1 - S * f)$, $t = V * (1 - (S * (1 - f)))$
5. ak $i = 0$ $R = V$, $G = t$, $B = p$
6. ak $i = 1$ $R = q$, $G = V$, $B = p$
7. ak $i = 2$ $R = p$, $G = V$, $B = t$
8. ak $i = 3$ $R = p$, $G = q$, $B = V$
9. ak $i = 4$ $R = t$, $G = p$, $B = V$
10. ak $i = 5$ $R = V$, $G = p$, $B = q$

5. Charakterizujte a popíšte model farebný model HLS.

FAREBNÝ MODEL HLS – najlepší pre ľudské oko

- Zložky: Hue – farebný tón, Lightness – svetlosť, Saturation – saturácia,
- Miešanie: aditívne
- Zmena zložky: uhlová a lineárna



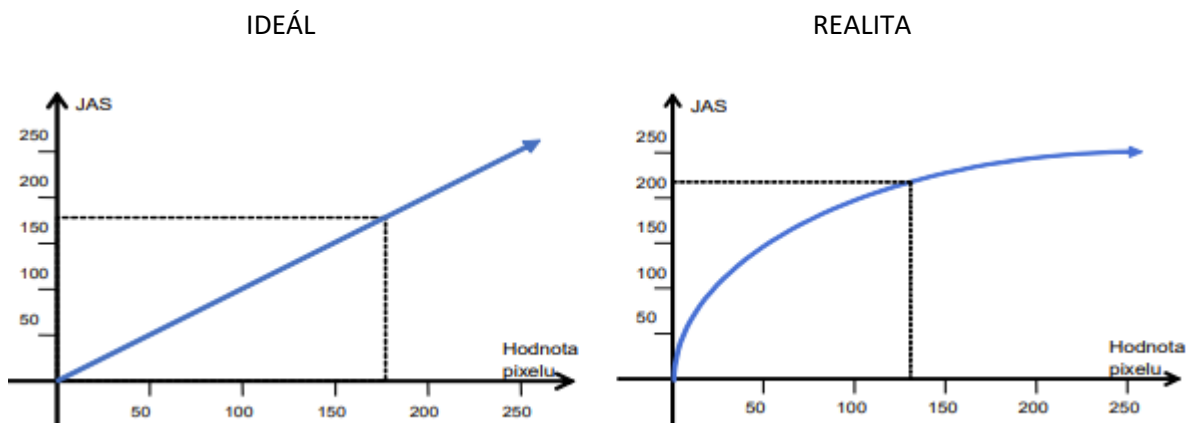
PREVOD RGB → HLS

1. $\text{minim} = \min(R, G, B)$
2. $\text{maxim} = \max(R, G, B)$
3. $L = (\text{minim} + \text{maxim}) / 2$
4. ak $L = 1$, $H = 0$, $S = 0$, koniec //biela farba
5. ak $\text{minim} = \text{maxim}$, $S = 0$, $H = 0$, koniec //odtiene šedej
6. inak $S = (1 - \text{minim}) / (1 - L) - 1$
7. ak $L > 0.5$
 $R2 = 1 + (R - 1) / (2 - 2L)$
 $G2 = 1 + (G - 1) / (2 - 2L)$
 $B2 = 1 + (B - 1) / (2 - 2L)$
inak $R2 = R$, $G2 = G$, $B2 = B$
8. ak prevláda zložka R: ak $G2 > B2$, $H = G2 / B2$ inak $H = 6 - B2 / G2$
9. ak prevláda zložka G: ak $R2 > B2$, $H = 2 - R2 / G2$ inak $H = 2 + B2 / G2$
10. ak prevláda zložka B: ak $G2 > R2$, $H = 4 - G2 / B2$ inak $H = 4 + R2 / B2$
11. $H = H * 60$, ak $H < 0$: $H = H + 360$

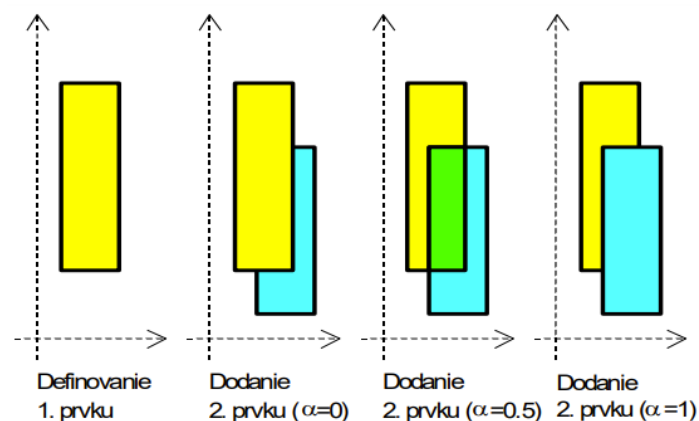
6. Charakterizujte gama korekciu a popíšte alfa-miešanie.

GAMA KOREKCIA

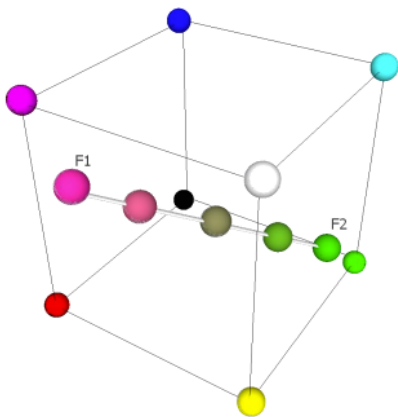
- vyrovnanie, korekcia toho, ako fyzicky viem rozsvietiť pixel oproti matematickému modelu
- na fyzickej úrovni - problem výroby (technológií) - výrobná chyba
- treba kalibrovať monitory (výšku napätia)



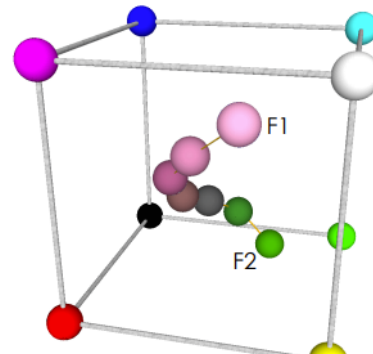
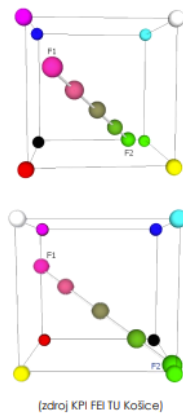
ALFA MIEŠANIE:



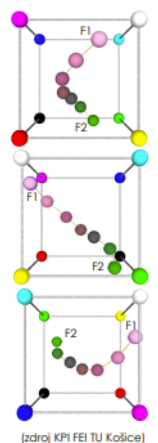
Lineárne, RGB priestor



Nelineárne, RGB priestor



nelineárny (NURBS) prechod $F(\text{arba})_1 - F(\text{arba})_2$ pri alfa miešaní, RGB priestor



7. Charakterizujte problematiku ľudského vizuálneho vnemu a jeho spracovania v relácii s počítačovou grafikou.

ĽUDSKÝ VIZUÁLNY VNEM

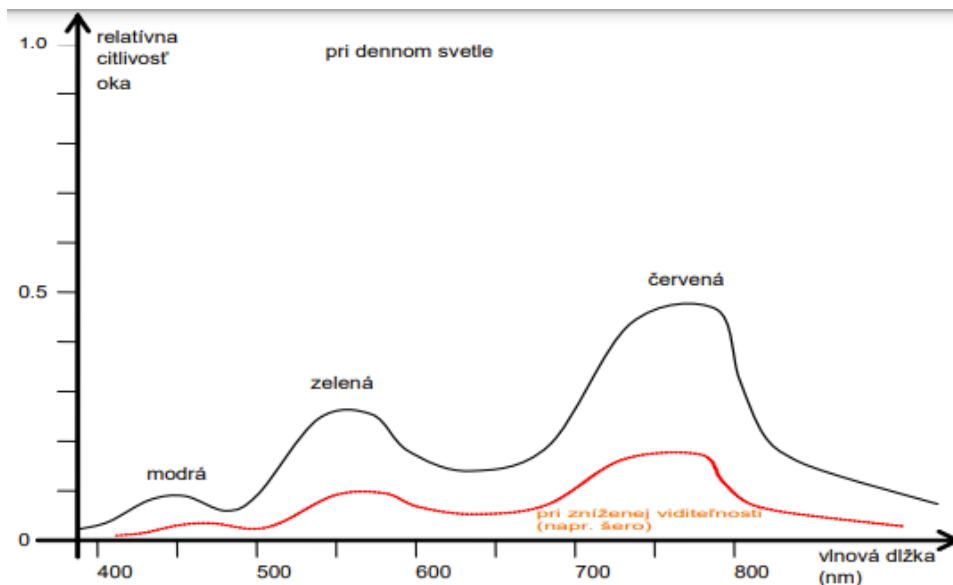
- Hlavným ľudským orgánom prijímania obrazových informácií je oko
- Hlavným ľudským orgánom spracovania obrazových informácií je mozog

Týmto je determinované použitie technológií počítačovej grafiky

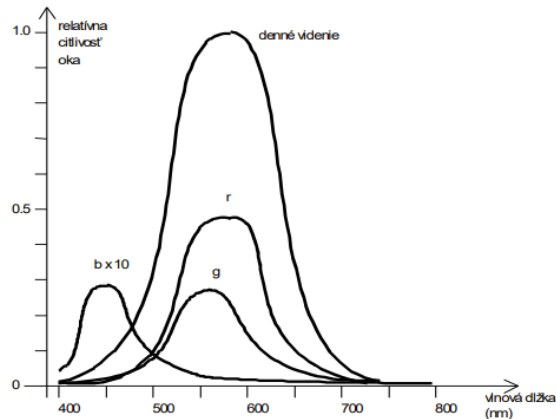
Receptory oka:

- Čapíky – (kuželovité-cones) počet cca 6-7 mil. sa nachádzajú v strede sietnice a sú citlivé na farby. Senzitívnosť týchto receptorov je delená do dvoch skupín, ktoré rozlišujú farebný rozdiel spektra:
 - červená-zelená (RG-cones) – citlivosť, dominantné
 - modrá-žltá (BY-cones)
- Tyčinky - (tyčinkovité-rods) počet cca 75-150 mil. rovnomerne pokrývajú celú sietnicu a umožňujú vnímať všeobecné obrazové informácie ako obrysy prípadne jas., je ich viac ako čapík.

RELATÍVNA CITLIVOSŤ ĽUDSKÉHO OKA NA PRECHODY MEDZI FARBAMI



RELATÍVNA CITLIVOSŤ ĽUDSKÉHO OKA NA FARBY



8. Charakterizujte problém miešania a rozptyľovania farieb (prevod do šedej škály, halftoning, dithering) v rámci počítačovej grafiky.

PREVOD RGB DO ODTIEŇOV ŠEDEJ

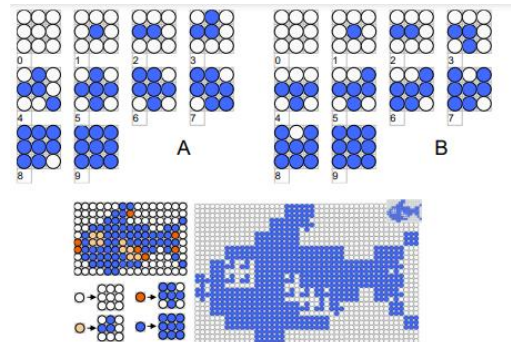
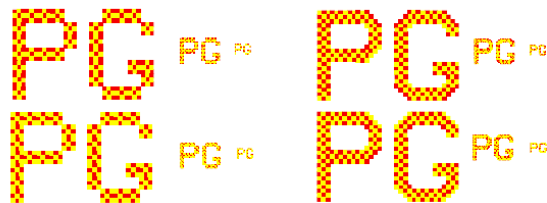
$$I = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

I - je výsledná intenzita (úroveň šedej) R, G, B - sú základné farebné zložky pôvodnej farby

MIEŠANIE FARIEB z pohľadu človeka: DITHERING



žltá a červená sa javí ako oranžová



HALFTONING: ak miešame 2 farby

- 1 pixel nahradený množinou pixelov, ale len binárnou maskou (2 farby)

- napríklad ryba viacfarebná - modrá - potrebujem vyššie rozlíšenie - monochromatická laserová tlač

Určenie farby:

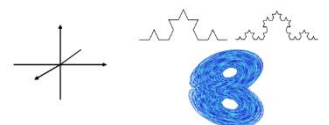
- náhodne
- spriemerňím bodov oblasti alebo použitím mediánovej funkcie:
 - zistí sa početnosť výskytu farieb v oblasti a vyberie sa tá farba, ktorá sa vyskytuje najčastejšie. Ak je výskyt farieb rovnaký, vyberie sa farba podľa iného pravidla alebo ľubovoľná z vyskytujúcich sa farieb.
- vyberie sa farba,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

9. Charakterizujte dimenziu priestoru a dimenziu objektu, štruktúra dimenzie.

DIMENZIA (ROZMER) PRIESTORU

A. Číselná

- Celočíselná (topologická) dimenzia (0,1,2,3,4), celočíselná metrika



- Neceločíselná dimenzia (2.6 a pod.)

B. Nečíselná - (informačná, dim. sebedobnosti)

- Topologická • Hausdorffova • Farebná • Fraktálna • dimenzia sebedobnosti • Kapacitná
- Informačná • dimenzia Rotácií ...

ŠTRUKTÚRA DIMENZIE (ROZMERU) PRIESTORU

Príklad štruktúry 3 hodnotovej dimenzie:

$$3 = 3 + 0 \text{ (geometria + čas)}$$

$$3 = 2 + 1 \text{ (geometria + čas)}$$

$$D(\text{imenzia}) = M + N + \dots$$

↑ veličina 1
 ↑ veličina 2
 ↑ veličina n

ROZMER OBJEKTOV

- 0-rozmerný objekt (bod)
- 1-rozmerný objekt (priamka, úsečka)
- 2-rozmerný objekt (plocha)
- 3-rozmerný objekt (teleso)

Všetky geometrické objekty, sa dajú spojitou transformáciou previesť na vyššie uvedené objekty ak majú rovnakú topologickú dimenziu. Všetky geometrické objekty, môžu reprezentovať vyššie uvedené objekty, ak majú rovnakú alebo nižšiu topologickú dimenziu.

POPIS A REPREZENTÁCIA OBJEKTOV

Pri spracovaní objektov sú zaujímavé dve hľadiská: • popis objektu • reprezentácia objektu

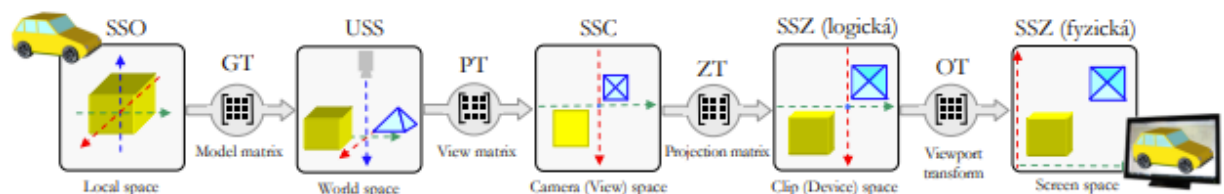
10. Popíšte vrstvy vizualizačného procesu.

VIDENIE (VISION) - je komplementárny problém, ktorý analyzuje obrazy na vytvorenie počítačových modelov sveta

VYKRESĽOVANIE (RENDERING) - je proces tvorby obrazu pomocou počítačových algoritmov alebo tiež oblasť štúdia zaoberajúceho sa syntézou obrazov z modelov sveta v počítači.

VIZUALIZÁCIA

- je proces transformácie popisu modelu virtuálneho sveta do výstupného obrazu na zobrazovacom zariadení
- základnou úlohou grafických systémov je generovať grafický výstup pre pripojené zobrazovacie zariadenie



VRSTVY VIZUALIZAČNÉHO PROCESU

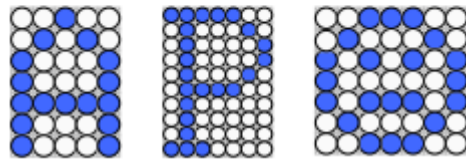
1. Definovanie/spracovanie modelu (reprezentácia, súradnicové systémy)
2. Transformácie nad objektami (geometrické)
3. Riešenie viditeľnosti
4. Tieňovanie
5. Osvetľovanie
6. Realistické zobrazovanie
7. Kompozícia a Vykresľovanie

11. Charakterizujte grafickú informáciu po objektivej aj typovej stránke.

- 1. Krok vizualizačného procesu

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 – TYP

1. Vektorová (spojitý priestor) - krajšia
2. Rastrová (nespojité priestor) – ľahšie vyrobiteľná

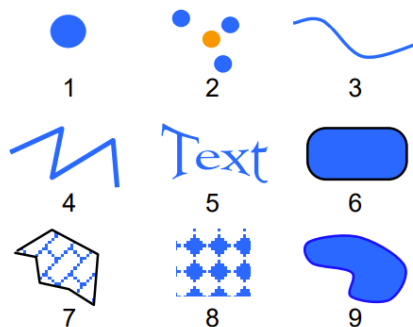
GRAFICKÉ OBJEKTY: - ZVACSENIE:

1. Vektorové - Žiadna strata
2. Rastrové - Strata informácií

12. Vymenujte základné grafické primitíva a ich atribúty.

GRAFICKÉ PRIMITÍVA: - prvé 2 sú raster aj vektor - všetky sú vektor

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



GRAFICKÉ PRIMITÍVA – ATRIBÚTY -Je nimi možné riadiť ich konečný tvar.

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

Atribúty môžu byť jednotlivým elementom priradené:

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

13. Popíšte spracovanie bodu a sledu bodov v rámci počítačovej grafiky,

BOD

- elementárny (atomárny) objekt, 0 rozmerný
- Základné atribúty sú: poloha a farba príp. čas (v prípade heterogénnej štruktúry dimenzie priestoru, ktorej je bod prvkom)
- Typy:
 - o Pixel – obrazový bod charakterizovaný dvomi súradnicami polohy a svojou farbou, najmenšia jednotka digitálnej rastrovej grafiky.
 - o Voxel – objemový bod charakterizovaný tromi polohovými súradnicami a svojou farbou.
 - o Texel – bod textúry, polohy v rámci určitej súradnicovej sústavy + poloha v rámci výplňového vzoru (textúry) + relácia 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)

SLED BODOV – POLYMARKER

- rozširujúci prvok a priamo nadväzuje na bod
- logicky zviazaná množina bodov na základe určitej relácie medzi prislúchajúcimi atribútmi týchto bodov.
- Operácia sa deje nad všetkými bodmi spolu.
- Relácia medzi atribútmi bodov polymarkra môže byť:
 - o homogénna - je relácia definovaná medzi rovnakými atribútmi jednotlivých bodov, napr. medzi atribútmi polohy.
 - o heterogénna - je definovaná medzi rôznymi atribútmi jednotlivých bodov, napr. farba niektorého bodu je závislá od polohy druhého.

KRIVKA, LOMENÁ ČIARA

- 1D útvar

- definovaný vrcholmi a hranami:

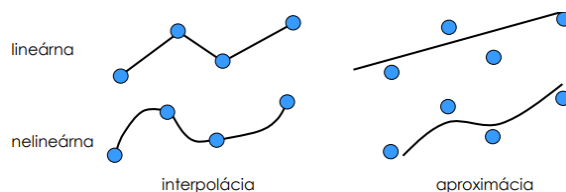
- a) neuzavretý (acyklický) - krivka (polyline)
- b) uzavretý (cyklický) - plocha (polygon)

- podľa typov hrán:

- lineárny - všetky segmenty sú lineárne
- nelineárny - stačí, že 1 segment je nelineárny

- podľa vplyvu vrcholov na tvar

- interpolačné - vrcholy sú súčasťou krivky
- aproximačné - vrcholy nemusia byť súčasťou krivky ale vplyvajú na jej tvar



14. Popíšte spracovanie úsečky v rámci počítačovej grafiky a uveďte základné metódy jej generovania

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

- Algoritmus založený na výpočte oboch súradníc –
 - o štandardný algoritmus
 - o vzorec - dáme cyklus pre x a vypočítame y podľa rovnice úsečky - potom ostatné body
- DDA - digital differential analyzer
- Bresenhamov algoritmus

15. Vysvetlite DDA algoritmus.

- prírastkový algoritmus (DDA - digital differential analyzer) založený na postupnom pripočítavaní konštantných prírastkov k oboj súradniciam x a y
- Rozlišujeme výpočet pre priamku:
 - o so smernicou menšou ako 1 – úsečka klesá
 - o so smernicou väčšou ako 1 – úsečka narastá

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

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

-potom 1 sa pripočítava k riadiacej osi, ktorá môže byť x alebo y a príslušná diferencia sa pripočítava k opačnej osi

- **py** je posun/prírastok k y osi
- **px** je prírastok/posun k x osi
- tam, kde max (=1, to druhé bude < 1) - riadiaca os
- px/py môžu byť float - spomalenie výpočtu - snaha obísť neceločíselný výpočet

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

16. Vysvetlite 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
- - zaokrúhľovanie - ak je výsledok do 0,5 ostáva sa v danom riadku, ak je väčší tak nový riadok

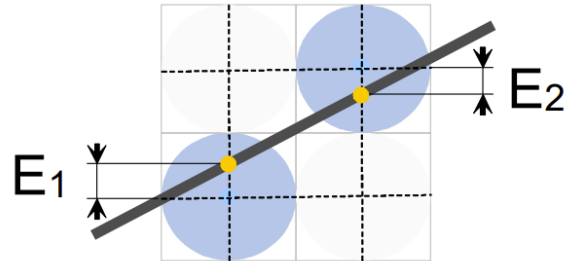
$$dx = |x_B - x_A|$$

$$dy = |y_B - y_A|$$

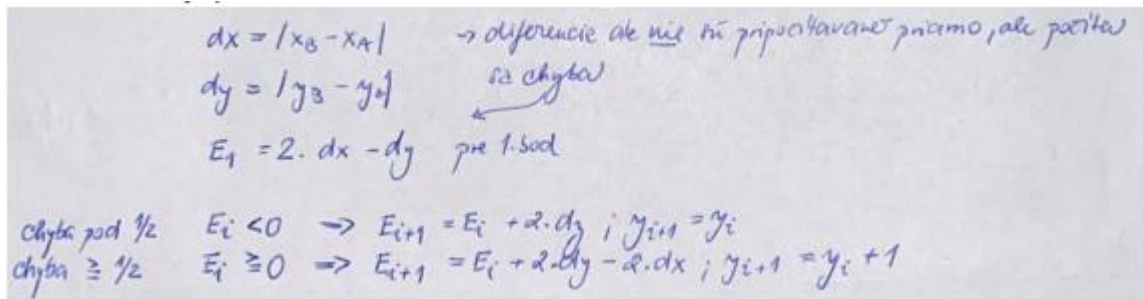
$$E_1 = 2 \cdot dx - dy \quad \text{pre prvý bod}$$

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

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



Stanovenie chybového člena ED pri Bresenhamovom algoritme



17. Popíšte spracovanie kružnice a elipsy v rámci počítačovej grafiky a uveďte základné metódy jej generovania.

KRUŽNICA - množina bodov rovnako vzdialených od stredového bodu $y = y_S \pm \sqrt{r^2 - (x - x_S)^2}$

METÓDY GENEROVANIA:

- Algoritmus kreslenia kružnice na základe parametrického vyjadrenia

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

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

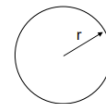
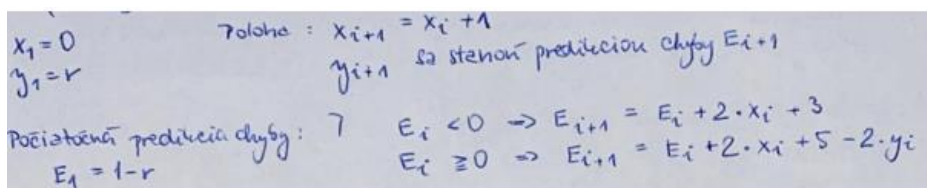
- Algoritmus kreslenia kružnice podľa predikcie chyby

$$x_1 = 0 \quad \text{Poloha:} \quad x_{i+1} = x_i + 1$$

$$y_1 = r \quad y_{i+1} \text{ sa stanoví predikciou chyby } E_{i+1}$$

$$\text{Počiatočná predikcia chyby:} \quad E_i < 0 \Rightarrow E_{i+1} = E_i + 2 \cdot x_i + 3$$

$$E_1 = 1 - r \quad E_i \geq 0 \Rightarrow E_{i+1} = E_i + 2 \cdot x_i + 5 - 2 \cdot y_i$$



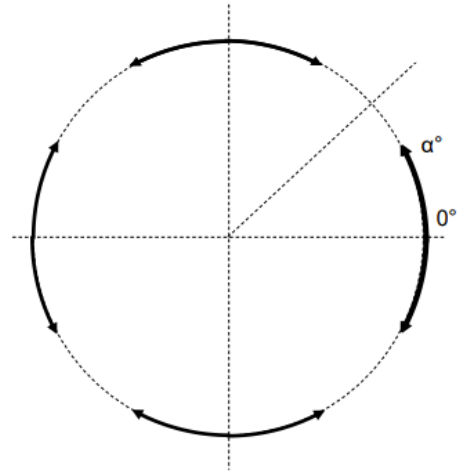
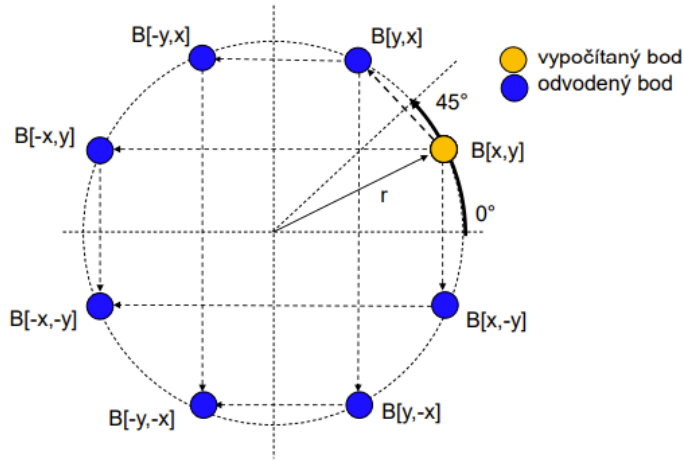
$$y = y_S \pm \sqrt{r^2 - (x - x_S)^2}$$

↓
polomer

S[x_s, y_s] ... stred

- float sa zaokrúhľuje - buď sa kreslí v novom alebo pôvodnom riadku (stĺpci) ako Bresenham
- rýchlejší ako prvý algoritmus

- Algoritmus kreslenia kružnice využitím osovej súmernosti
- najrýchlejšia varianta
- kružnica je oso súmerná - netreba počítať celých 360°, stačí do 45°
- zvyšok nakreslíme podľa osovej súmernosti (odvodíme)



ELIPSA - množina bodov rovnako vzdialených od dvoch ohnísk

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

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



- 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}

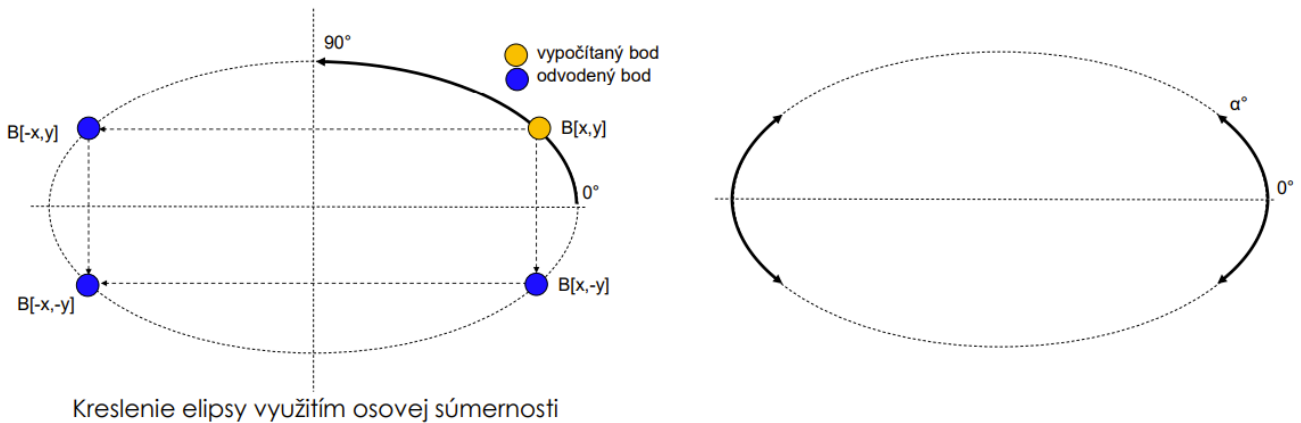
Počiatková predikcia chyby:

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

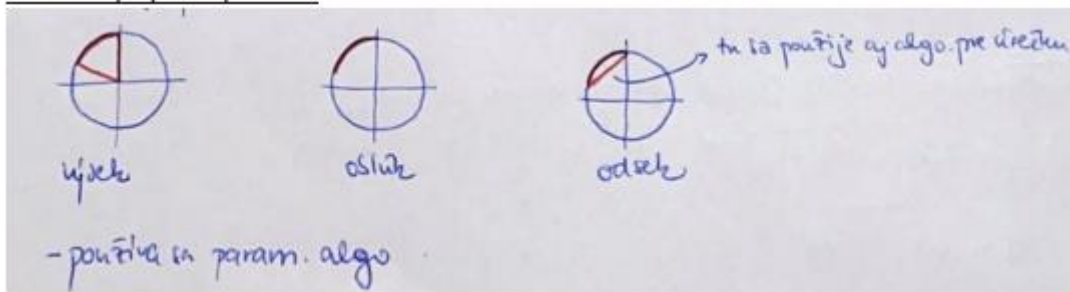
$$E_i < 0 \Rightarrow E_{i+1} = E_i + b^2 \cdot (2 \cdot x_i + 1)$$

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

- Algoritmus kreslenia elipsy podľa osovej súmernosti – do 90°



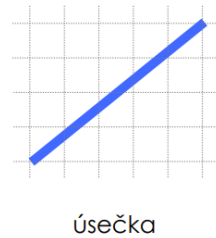
KRUHOVÉ (ELIPTICKÉ) OBLASTI:



18. Charakterizujte a popíšte antialiasing.

ANTIALIASING

- ak máme vektorový objekt, ale ideme ho vykresľovať na rastrový monitor - vytvorí sa nám rastrová náhrada (alias)
- proces, ktorý vytvára kostrbatosť, schodíkovosť rastrovej náhrady
- prípady:



- schodovité zobrazovanie priamych rovných čiar a hraníc polygónov na rastrových displejoch
- zobrazovaný objekt je menší ako veľkosť - malé objekty potom nie sú vôbec zobrazené alebo napr. tenké čiary nie sú opticky hladké resp. ucelené (sú zobrazené ako nepravidelná postupnosť bodov).

- pri zobrazení 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)

- môžu sa miesto neho použiť aj filtrovacie a rozptyľovacie techniky

Filtrovacie a rozptyľovacie techniky - riešenie problému rozdielu medzi požadovanou kvalitou výsledných obrazov a obmedzenými možnosťami dostupného farebného priestoru zariadenia alebo techniky pre podporu antialiasingu

- rozptyľovanie - dithering - použitie viacerých farieb
- poltonovanie - halftoning - len 2 farby

Mediánová funkcia



(11,5,15,7,1)

(1,5,7,11,15)

7



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

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

10

19. Popis a reprezentácia objektov v počítačovej grafike, priestor a jeho parametre.

- sme stále na 1. úrovni vizualizačného procesu

PRIESTOR A JEHO PARAMETRE

Podľa charakteru priestoru:

1. Translačný priestor
2. Rotačný priestor
3. Kombinovaný priestor

Podľa typu dimenzie:

1. Celočíselné - topologické
2. Neceločíselné

Podľa štruktúry dimenzií (primárne geometria) a ich počtu:

1. Homogénna štruktúra ($N+0$) – ND priestor - iba 1 zložka na definovanie štruktúry dimenzie
2. Heterogénna štruktúra ($N_1+N_2+\dots+N_n$) – ($N_1+N_2+\dots+N_n$)D priestor - 2 a viac zložiek - väčšinou geometria + čas (niekedy aj farba)

Príklady: celočíselný translačný priestor, celočíselný rotačný priestor

Objekty v PG

1. 0-rozmerný objekt (bod)
2. 1-rozmerný objekt (priamka, úsečka)
3. 2-rozmerný objekt (plocha)
4. 3-rozmerný objekt (teleso)

- Všetky geometrické objekty, sa dajú spojitou transformáciou previesť na vyššie uvedené objekty ak majú rovnakú topologickú dimenziu.

- Všetky geometrické objekty, môžu reprezentovať vyššie uvedené objekty, ak majú rovnakú alebo nižšiu topologickú dimenziu

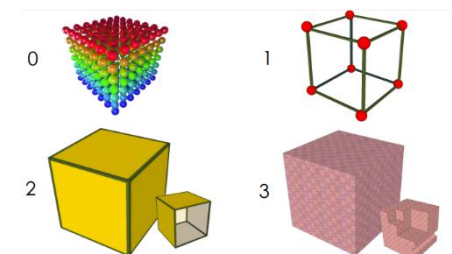
POPIS A REPREZENTÁCIA OBJEKTOV

Pri spracovaní objektov sú zaujímavé dve hľadiská:

- popis objektu
- reprezentácia objektu

Pre modelovanie telies sú používané tri základné spôsoby popisu:

- hraničná reprezentácia a jej štruktúrovaná derivácia B-rep
- konštruktívna geometria telies (CSG)
- vypočítavanie obsadených častí priestoru - už sa nepoužíva
- systémy založené na množine bodov - mračná bodov (point clouds) 0
- systémy založené na drôtovom modeli (wire frame model) 1
- systémy založené na povrchovom modeli (surface model) 2
- systémy založené na objemovom modeli (solid model) 3

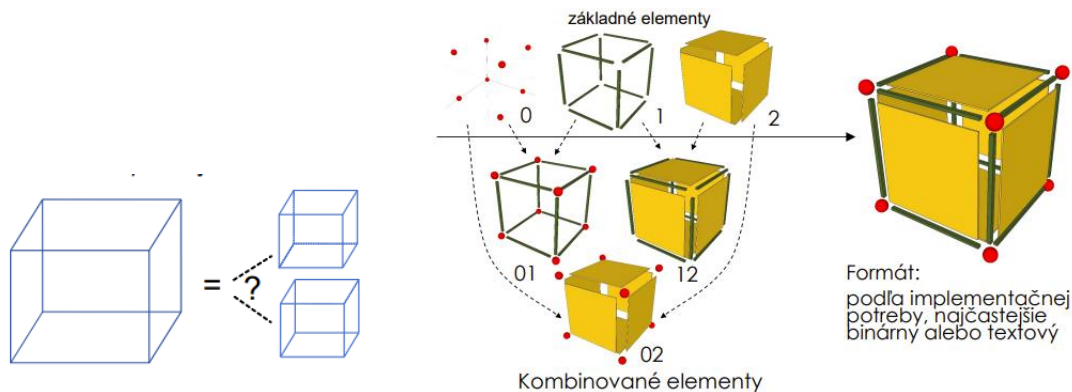


20. Hraničná reprezentácia.

HRANIČNÁ REPREZENTÁCIA

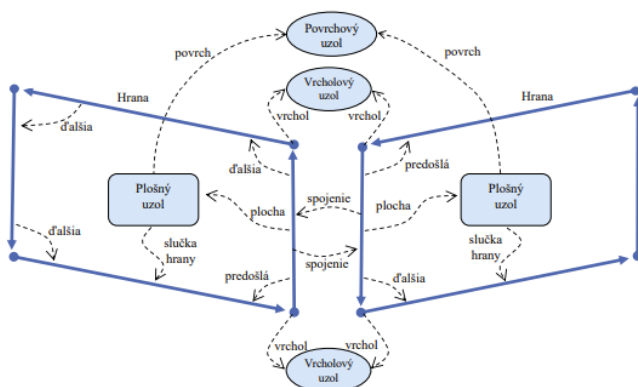
- najvýznamnejšou časťou telesa sú jeho hraničné elementy ako napr. hrany alebo povrch, ktorý tvorí hranicu medzi hmotou telesa a okolitým priestorom.
- hraničné plochy delíme: - časti rovín - analytické plochy - špeciálne parametrické plochy

- najjednoduchšia metóda popisu hranice telies: stanovenie hrán a vrcholov na povrchu telesa (drôtový model) - môže byť nejednoznačné



HRANIČNÁ REPREZENTÁCIA – B-REP METÓDA

- B-rep - definuje objekt svojim povrchom
- zložený zo stien, ktoré sa môžu dotýkať iba na spoločných hranách, každá hrana je orientovaná, jednoznačne určená či je vonkajšia alebo vnútorná
- objekt reprezentovaný dynamickou údajovou štruktúrou - Wired Edge Structure
- 4 druhy uzlov: - vrchol (Vertex) - hrana (Edge) - stena, plocha (Face) - teleso (Solid)

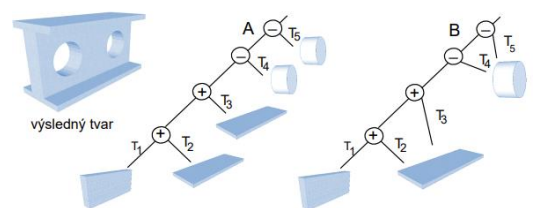


Rozšírená hranová (winged-edge) štruktúra

21. Konštruktívna geometria telies. - (CSG - Constructive Solid Geometry)

- def. abstraktnou údajovou štruktúrou - strom:
 - o -listy - atomárne elementy z ktorých sa objekt skladá
 - o uzly - operácie medzi atomárnymi elementami / objektami / subobjektami na danej úrovni stromu (zjednotenie, rozdiel, prienik)
 - o hrany - transformácie atomárných elementov / subobjektov vstupujúcich do rodičovského uzla
 - o koreň - celý objekt

- strom môže byť:
 - cyklický (B) - môže byť viac časovo náročný
 - acyklický (A) - viac pamäťovo náročný
- 1 objekt môže vzniknúť viacerými postupmi (rôzne stavy)
 - nevie povedať na začiatku, či to bude to isté alebo nie
- strom vyhodnocujem zdola nahor



22. Vymenujte a v krátkosti popíšte súradnicové sústavy používané v počítačovej grafike.

- Súradnicová sústava umožňuje parametrizovať priestor a definovať jeho počiatkový bod
 1. Počiatok (stred súradnicovej sústavy) - (ORIGIN)
 2. Os – definuje smer rozvoja fyzikálnej veličiny v príslušnej dimenzii priestoru popisovaného súradnicovou sústavou - (nemusi byt len priamka, môže byť krivka)
 3. Súradnice (parametre) – definujú jednoznačne polohu v rámci súradnicovej sústavy podľa charakteru a štruktúry dimenzií priestoru
 1. Počet súradníc – podľa počtu dimenzií priestoru

Súradnicové sústavy

Podľa rozvoja hodnôt veličín príslušných dimenzií priestoru ktorému je súradnicová sústava priradená:

- 1. Lineárne súradnicové sústavy
- 2. Nelineárne súradnicové sústavy

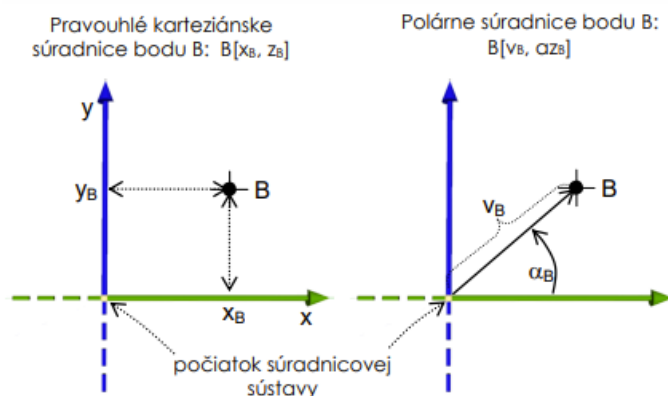
Podľa vzťahu osí súradnicového systému:

- 1. Pravouhlé
- 2. Nepravouhlé

-Pre počítačovú grafiku sa primárne používajú celočíselné translačné alebo rotačné lineárne súradnicové sústavy - - väčšinou lineárne pravouhlé kombinované

- typy:

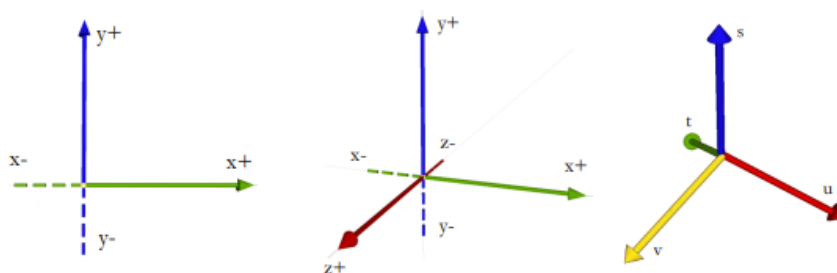
1. karteziánska - pravouhlá, translačná, lineárna
2. polárna (3D-sférická) - rotačná, lineárna



A. 2D (topologicky, 2+0) translačná pravouhlá lineárna súradnicová sústava

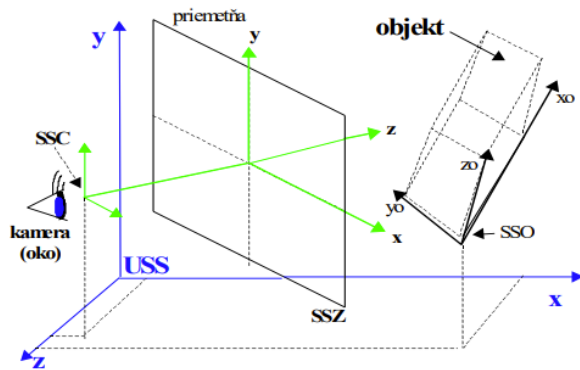
B. 3D (topologicky, 3+0) translačná pravouhlá lineárna súradnicová sústava

C. 4D (topologicky, 4+0) translačná pravouhlá lineárna súradnicová sústava (projekcia do 3D, disfenoid)



- základné súr. sústavy:

1. USS - Univerzálna (používateľská, globálna) súr. Sústava
2. SSO - Súradnicová sústava objektu
3. NSS - Normalizovaná súradnicová sústava (kontrola limitov)
4. SSZ - Súr. sústava zariadenia (monitor, hologram)
5. SSC - Súr. sústava kamery
6. SST - Súr. sústava textúry



23. Charakterizujte transformácie a transformačné zobrazovacie reťazce v rámci počítačovej grafiky.

-transformácie – 2. fáza vizualizačného procesu

Transformácia je proces, ktorý mení (transformuje) vstupný objekt (jeho parametre) na objekt výstupný

- transformácie vieme vykonávať:

- a) paralelne - neovplyvňujú sa navzájom
- b) sekvenčne

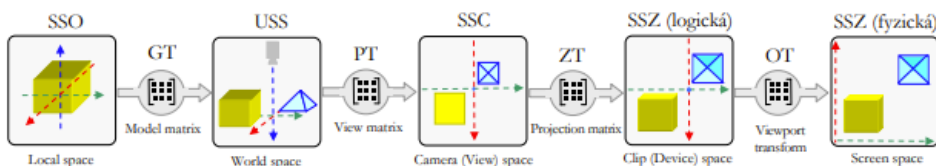
- ak sa transformácia skladá z viacerých - aby bola lineárna tak všetky dielčie transformácie musia byť lineárne

- Z hľadiska použitého typu sústavy je možné príslušnú transformáciu z hľadiska jej vonkajšieho prejavu v rôznych sústavách vnímať rozdielne.

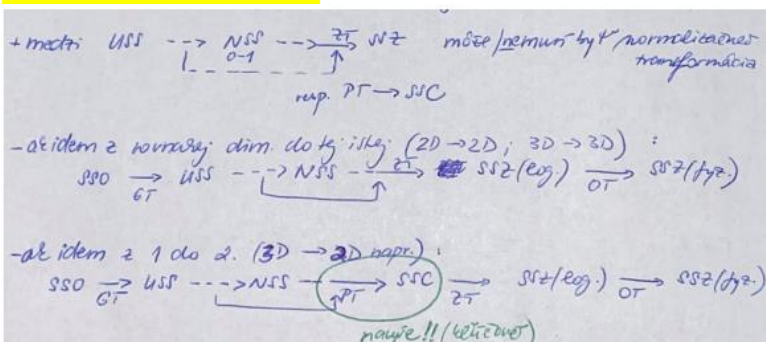
- niekedy je ľahšie popísať transformáciu v inej súradnicovej sústave a potom vyjadriť transformáciu medzi sústavami

MEDZISÚSTAVOVÉ TRANSFORMÁCIE

- Globálna (geometrická) transformácia GT,
- Pohľadová transformácia PT,
- Zobrazovacia (premietacia) transformácia ZT,
- Orezávací transformácia OT

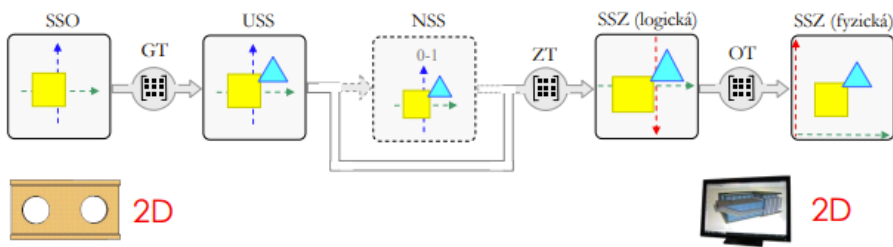


TRANSFORMAČNÉ REŤAZCE

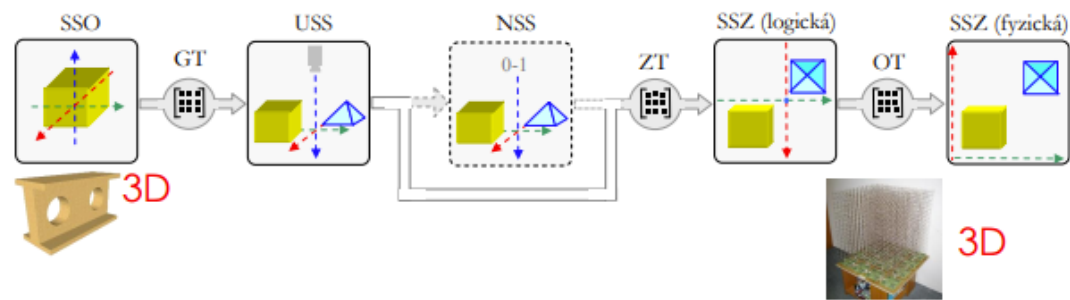
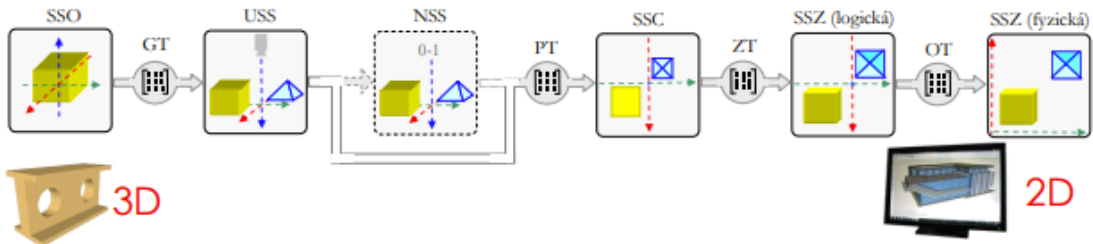


- kameru chápame ako prevodové zariadenie medzi dimenziam

TRANSFORMAČNÉ REŤAZCE 2D

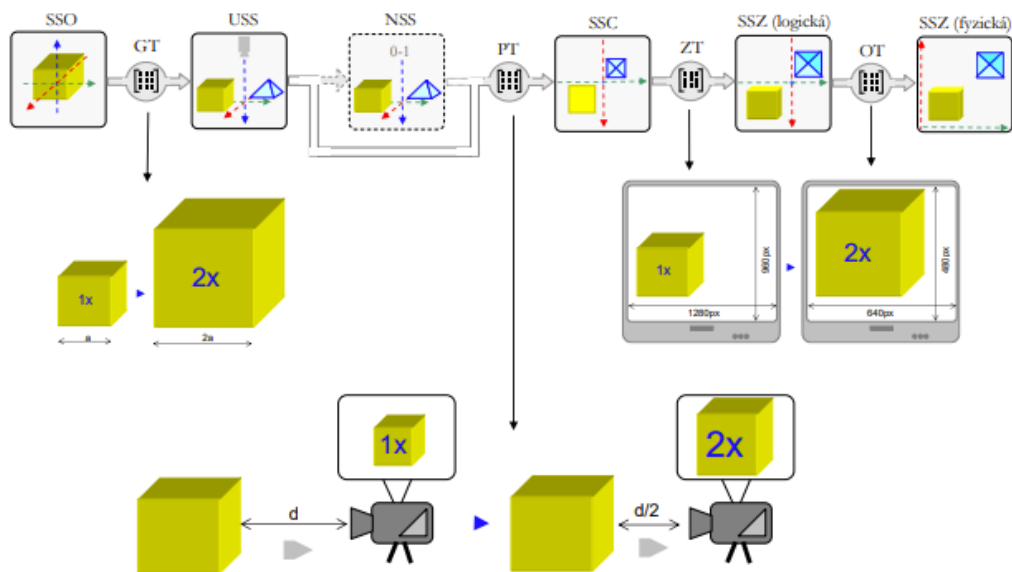


TRANSFORMAČNÉ REŤAZCE 3D



24. Transformácie v rámci zobrazovacích reťazcov, transformačné matice a homogénne súradnice.

APLIKÁCIA TRANSFORMÁCIÍ V ZOBRAZOVACOM REŤAZCI



- transformáciu (požadovanú) môžeme vykonávať v rôznych bodoch reťazca - napr. mam kocku, kt. chcem 2x zväčšiť - môžeme to spraviť na úrovni GT (zväčším geometricky) , PT (priblížim kameru), ZT/OT

25. Charakterizujte geometrickú transformáciu posunutia.

POSUNUTIE (TRANSLÁCIA):

2D : $x'_B = x_B + p_x$
 $y'_B = y_B + p_y$

$$T_P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ p_x & p_y & 1 \end{pmatrix}$$

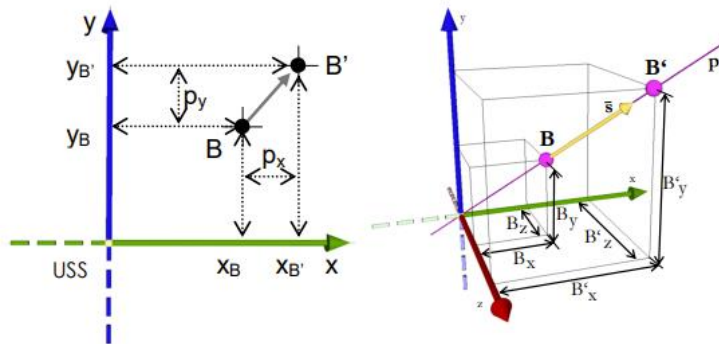
koefficienty posunutia

3D : $x'_B = x_B + p_x$
 $y'_B = y_B + p_y$
 $z'_B = z_B + p_z$

$$T_P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_x & p_y & p_z & 1 \end{pmatrix}$$

transformácia bez m (tam realizujeme posunutie)

$1 \cdot x + 0 \cdot y + 0 \cdot z + p_x = x'$

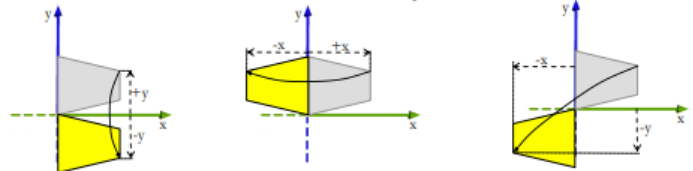


26. Charakterizujte geometrickú transformáciu zrkadlenia.

Zrkadlenie (2D)

- spočíva vo výmene znamienok
 - vzhľadom na os x: $x' = x$; $y' = -y$
 - vzhľadom na os y: $x' = -x$; $y' = y$
 - vzhľadom na stred: $x' = -x$; $y' = -y$

vzhľadom na os x vzhľadom na os y vzhľadom na stred



- môžeme zrkadliť nad objektom, kt. dimenzia je nižšia ako objektu and ktorým robím zrkadlenie - koľko znamienok mením toľko je rozdiel dimenzií týchto objektov
- RASTER - nemôžeme zrkadliť naraz
 - vymieňam buď po stĺpcoch alebo riadkoch
 - používam ak chcem otočiť len objekt ale pozadie nie

Zrkadlenie (3D)

$$T_{zx} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

wci x

$$T_{zxy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

wci xy
plocha

$$T_{zst} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

wci stred
homogenizujúci faktor

27. Charakterizujte geometrickú transformáciu zmeny mierky, zväčšenie rastrového objektu

$$x'_B = M_x \times x_B$$

$$y'_B = M_y \times y_B$$

↳ toto je heterogénny spôsob
 - homogénny spôsob → ar $M_x = M_y = M$ (= M_z pri 3D)

$$T_M = \begin{bmatrix} M_x & 0 & 0 \\ 0 & M_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x'_B = M_x \times x_B$$

$$y'_B = M_y \times y_B$$

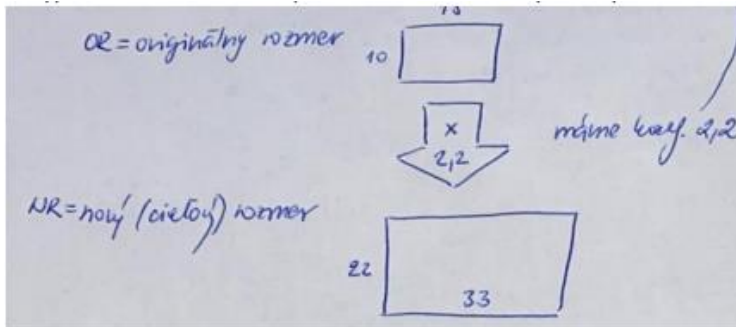
$$z'_B = M_z \times z_B$$

$$T_M = \begin{bmatrix} M_x & 0 & 0 & 0 \\ 0 & M_y & 0 & 0 \\ 0 & 0 & M_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

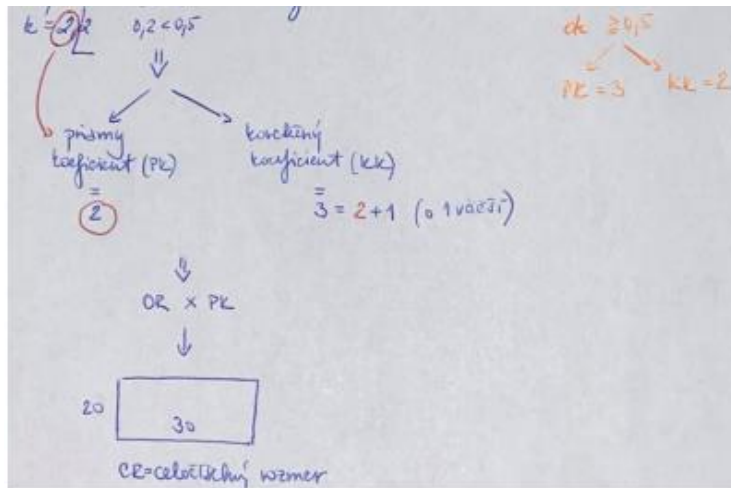
RASTER

a) zväčšenie (ak je desatinná časť koeficientu menšia ako 0,5 a väčšia ako 0,2):

1. výpočet nového rozmeru podľa koeficientu zmeny mierky:



2. výpočet priameho a korekčného koeficientu a nového celočíselného rozmeru



3. výpočet rozdielu nového rozmeru a celočíselného rozmeru, následne výpočet krokov aplikácie korekčného koeficientu - idem vypočítať, koľko pixelov ešte musím dopracovať v každom smere

$$\Delta x = |NR_x - CR_x| = |33 - 30| = 3$$

$$\Delta y = |NR_y - CR_y| = |22 - 20| = 2$$

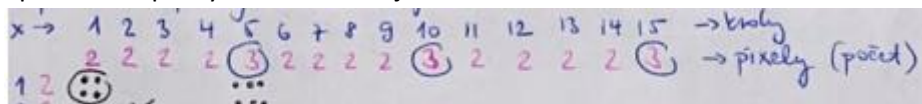
-zistím v ktorom kroku mám aplikovať tento korekčný krok:

$$CK_x = \text{int}(OR_x / \Delta x) = 15 / 3 = 5 \rightarrow \text{každý 5. krok musím korigovať}$$

$$CK_y = \text{int}(OR_y / \Delta y) = 10 / 2 = 5 \rightarrow \text{...}$$

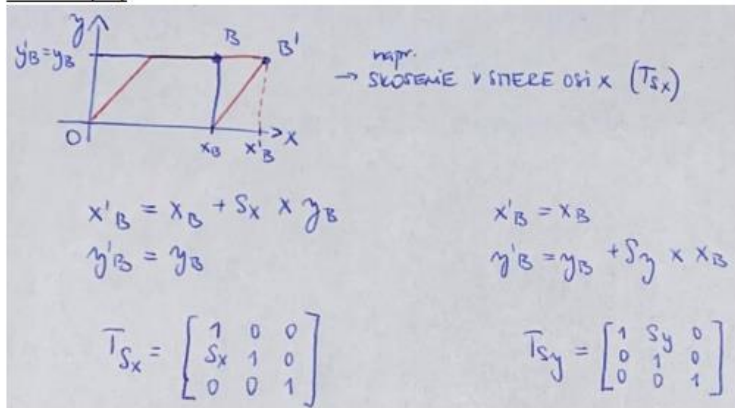
↳ toto užito rovnako

4. aplikácia na pixely rastrového objektu:

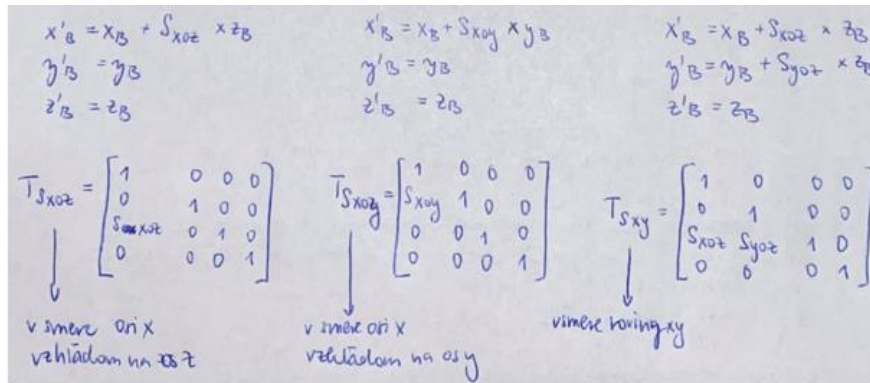


29. Charakterizujte geometrickú transformáciu skosenia.

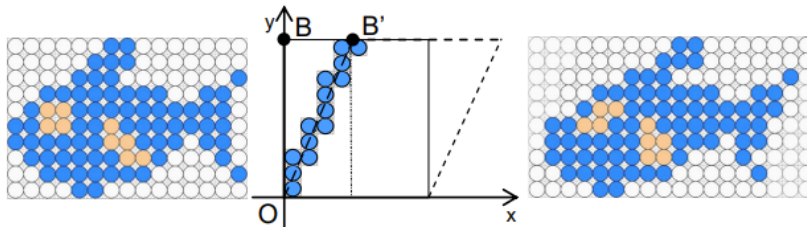
SKOSENIE (2D)



SKOSENIE (3D)



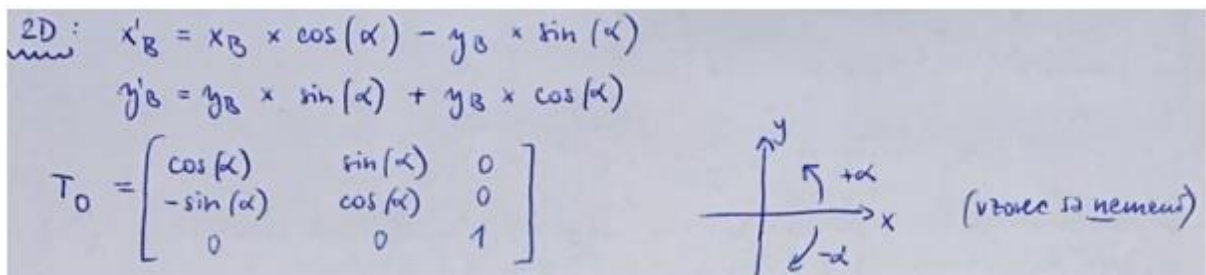
SKOSENIE RASTROVÝCH OBJEKTOV



30. Charakterizujte geometrickú transformáciu otočenia.

- otáčanie definované Eulerovými uhlami a reprezentované všeobecnými transformačnými maticami - vždy musíme definovať okolo ktorej osi sa deje čiastková rotácia - neumožňuje všeobecnú rovnicu pre rotáciu
- otáčanie definované Eulerovým teorémom a reprezentované quartermiónmi (vymyslené 1841) - v súčasnosti používané

OTÁČANIE (ROTÁCIA, 2D)



OTÁČANIE (ROTÁCIA, 3D)

3D: • okolo osi x : $x'_B = x_B$
 $y'_B = y_B \times \cos(\alpha) - z_B \times \sin(\alpha)$
 $z'_B = y_B \times \sin(\alpha) + z_B \times \cos(\alpha)$

$$T_{Ox} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

• okolo osi y : $x'_B = x_B \times \cos(\alpha) + z_B \times \sin(\alpha)$
 $y'_B = y_B$
 $z'_B = -x_B \times \sin(\alpha) + z_B \times \cos(\alpha)$

$$T_{Oy} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

• okolo osi z : $x'_B = x_B \times \cos(\alpha) - y_B \times \sin(\alpha)$
 $y'_B = x_B \times \sin(\alpha) + y_B \times \cos(\alpha)$
 $z'_B = z_B$

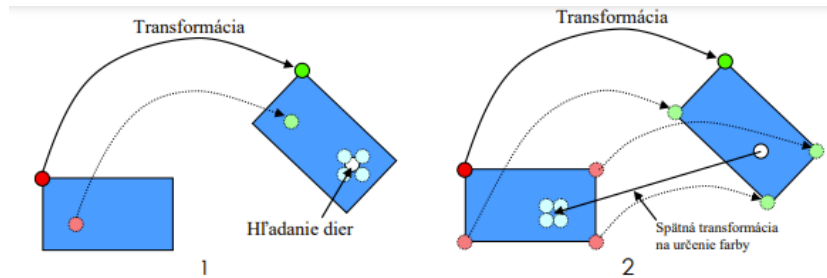
$$T_{Oz} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- nesmiem to spraviť naraz v 1 matici (mením 2 súr.)
- robí sa to krok po kroku
- dá sa kombinovať s posunom (neprekrývajú sa koeficienty)

OTÁČANIE RASTROVÝCH OBJEKTOV

a) priame otáčanie - otáčanie s interpoláciou protiľahlých bodov

- idem po pixeloch a hľadám ich nové pozície
- existuje problém: v rastru viem pracovať len s celými číslami a keď vzniknú float hodnoty tak sa môže pri zaokrúhľovaní stať, že na 1 miesto padnú 2 pixely



- 2 stratégie: farba prvého / farba posledného
- taktiež sa môže stať, že vzniknú diery - riešenie je masková matica kam zapisujem či sú pixely obsadené alebo nie (0/1), tam kde ostane 0 tak sa doplní farba (väčšinou medián)

b) spätné otáčanie - otáčanie s interpoláciou všetkých bodov

- vezmem vzor a otočím len jeho hraničné body (vrcholy)
- potom vypočítam medzi vrcholmi úsečky (Bresenham) - dostanem hrany
- v ohraničenej oblasti počítam spätnú transformáciu, aby som správne určil farby
- použijem inverznú maticu
- vedie jednoznačne k cieľu

31. Rotácia okolo všeobecnej priamky využitím Eulerových uhlov.

- len v prípade použitia Eulerových uhlov (pri quartermiónoch sa to robí naraz)

$$T_P \times T_{Oa} \times T_{Ob} \times \boxed{T_{O}} \times T_{Ob}^{-1} \times T_{Oa}^{-1} \times T_P^{-1}$$

1. posunieme sústavu / objekty tak aby os otáčania prechádzala počiatkom SS (T_P)
2. celé to (vrátane osi) sklopím do pôdorysu (T_{Oa})
3. potrebujem os otáčania stotožniť s niektorou z osí SS (T_{Ob})
4. vykonám rotáciu okolo požadovanej osi (T_O)

5. ,6. ,7. návrat do pôvodnej polohy (inverzie) aby os otáčania bola tam kde bola na začiatku problémy:

1. inverzná matica nemusí existovať

2. uhly a, b a náš požadovaný uhol sa teoreticky môžu rovnať (alebo aspoň 2 z nich) a vznikne $\sin A / \cos A = \tan A$.. tangens nemá spojité obor hodnôt - nedef. výsledok

- stáva sa to pri uhloch blízkyh 0° a 90°

- napríklad rubikova kocka - neviem dotočiť - „gimbal lock“

- strata stupňa voľnosti = Degree of Freedom (DOF)

- počet transformácií vzhľadom na počet nezávislých súradníc z hľadiska pohyblivosti objektu v danej SS

- v kartezianskej sústave pre transláciu DOF = 3, pre rotáciu DOF = 3 = 6 DOF

Je ťažké predvídať ako sa postupné rotácie okolo základných osí navzájom ovplyvnia. (maticová reprezentácia Eulerových uhlov má prirodzenú jedinečnosť v parametrizácii) Je možné vytvoriť takú postupnosť rotácií, že vo výslednej rotácii sa stratí jeden stupeň voľnosti. Táto situácia sa nazýva gimbal lock (strata stupňa voľnosti).

(Stupeň voľnosti (Degree of Freedom (DOF)) - počet transformácií (najčastejšie translačných a rotačných) vzhľadom na počet nezávislých súradníc (dimenzií) z hľadiska pohyblivosti objektu v danej súradnicovej sústave)

32. Vymenujte a v krátkosti popíšte 2D premietacie transformácie používané v počítačovej grafike, charakterizujte kolmú projekciu.

Projekcie

- špeciálne transformácie

- medzi USS a SSC (kamery)

- väčšinou medzi rôznymi dimenziami (najčastejšie 3D -> 2D)

- 2. vrstva vizualizačného procesu

- pracovná oblasť sa delí:

- logická oblasť SSZ

- zviazaná s USS - fyzická oblasť SSZ - podľa fyzického zariadenia (čo máme k dispozícii)

- zorné pole (FOV)

- premieta sa to na stred projekčnej šošovky

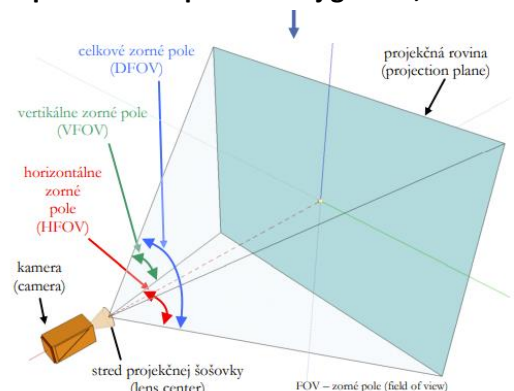
-3 základné zorné polia:

- vodorovné (HFOV)

- zvislé (VFOV)

- celkové (DFOV) - uhlopriečka - väčšinou 60° - 70°

- aplikuje sa v pohľadovej transformácii a súradnicovej sústave kamery



TYPY PREMIETACEJ TRANSFORMÁCIE

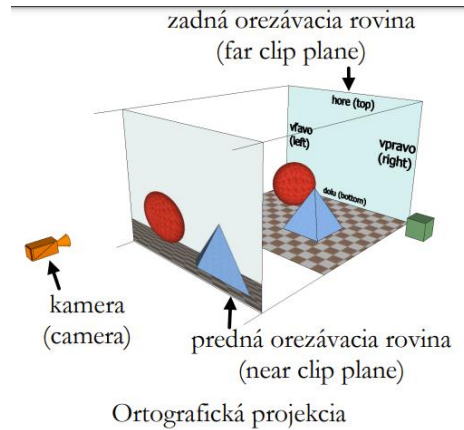
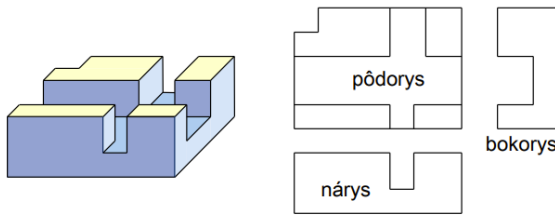
a) kolmá (ortografická)

b) axonometrická (axonometria)

c) perspektívna (perspektíva) - najčastejšie využívaná

a) **kolmá (ortografická) :**

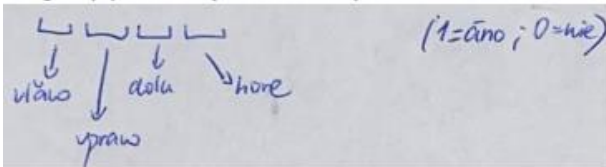
- nevytvára dojem hĺbky - musíme použiť pôdorys, bokorys, nárys (aspoň 3)
- 100% viditeľnosť len ak mám 6 pohľadov



33. **Popíšte princíp Cohen-Sutherlandovho algoritmu.**

COHEN-SUTHERLANDOV ALGORITMUS

- orezávacía transformácia
- logický priestor je rozdelený do 9 oblastí - každá je popísaná ako 4-bit číslo



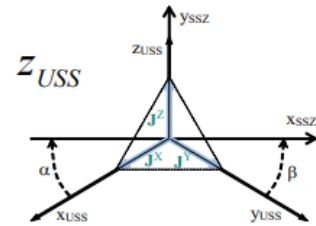
- zobrazovacie okno = fyzická oblasť = 0000 (v strede)
- ak časť úsečky leží mimo zobrazovacieho okna - počítame priesečník
 - kreslí sa potom len od viditeľného bodu po priesečník resp. od jedného priesečníka po druhý
- ak je celá úsečka mimo - nevykreslí sa

vľavo hore 1001	hore 0001	vpravo hore 0101
vľavo 1000	Zobrazovacie okno (fyzická oblasť) 0000	vpravo 0100
vľavo dolu 1010	dolu 0010	vpravo dolu 0110

- kódy oboch koncových bodov sú nulové (prázdne) potom oba koncové body ležia vo vnútri zobrazovacieho okna a je možné úsečku vykresliť bez orezania.
- jeden z kódov nenulový, potom je nutné orezanie, pretože časť úsečky určite leží mimo zobrazovacieho okna.
- obidva kódy sú nenulové. Potom sú možné dva prípady:
 - celá úsečka je mimo zobrazovacieho okna a nevykreslí sa. Toto sa deje najmä ak obidva kódy dva príslušné bity rovnaké (napr. 1000 a 1010).
 - časť úsečky je v okne. Toto sa môže stať, ak obidva kódy nemajú dva príslušné bity rovnaké (napr. 1000 a 0100). Vtedy je nutné orezanie.

34. Popíšte axonometrickú projekciu používanú v počítačovej grafike.

- stredová projekcia
- os, kt. ide nahor, má projekciu takú ako ju vidíme
- zvyšné dve osi pod uhlom
- vzdialenosť kamery nie je taká podstatná
- „axonometrický kríž (trojuholník)“



$$x_{SSZ} = -J^X \cdot \cos(\alpha) \cdot x_{USS} + J^Y \cdot \cos(\beta) \cdot y_{USS}$$

$$y_{SSZ} = -J^X \cdot \sin(\alpha) \cdot x_{USS} - J^Y \cdot \sin(\beta) \cdot y_{USS} + J^Z \cdot z_{USS}$$

- typy axonometrickej projekcie:

- izometria: $J^X = J^Y = J^Z$ a $\alpha = \beta$
- dimetria: $J^X = J^Y$ a $\alpha = \beta$
- trimetria: $J^X \neq J^Y \neq J^Z$ a $\alpha \neq \beta$
- technická axon.: $J^X = J^Y$, $J^Z = \frac{1}{2} J^X$, $\alpha = 45^\circ$, $\beta = 0^\circ$



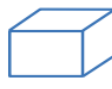
Izometria



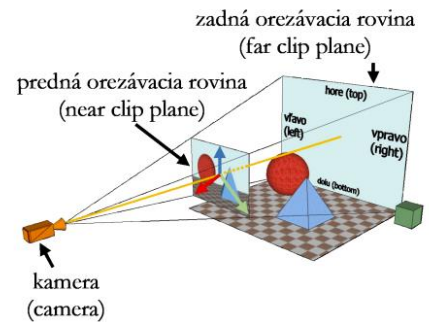
Dimetria



Trimetria



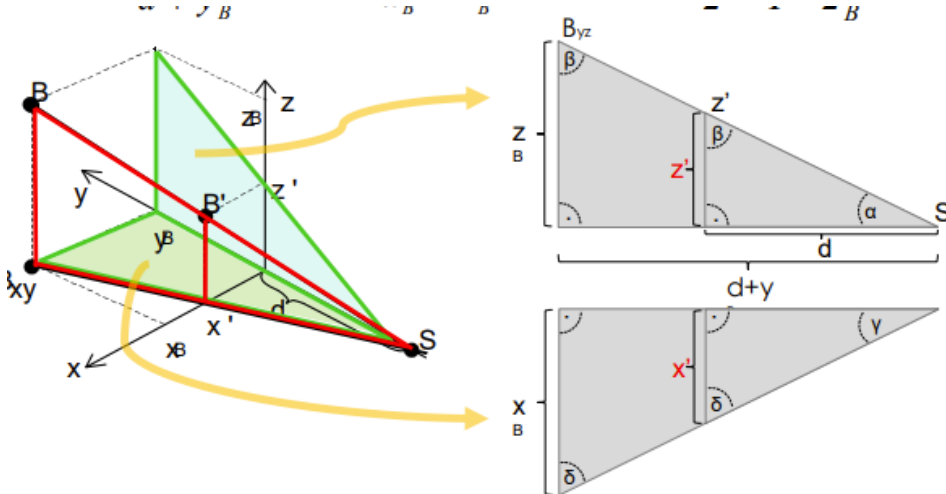
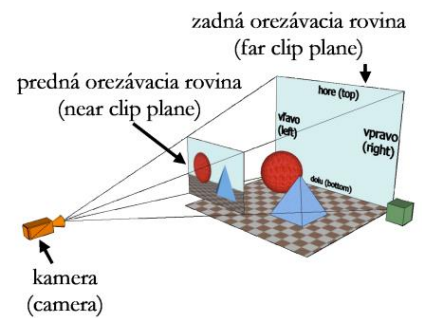
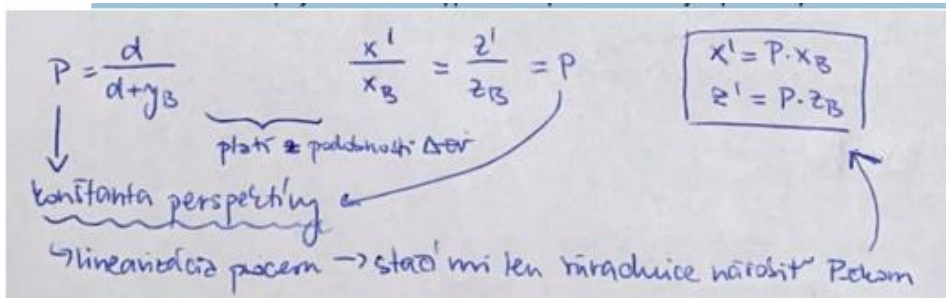
Technická axonometria



Stredová projekcia

35. Popíšte perspektívnu projekciu v počítačovej grafike.

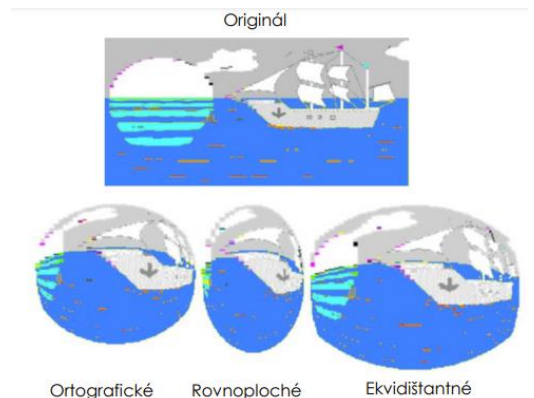
- dôležitá je vzdialenosť kamery od projekcie, ihlan pohľadu
- rovina xz je zobrazovacia
- mám bod B ... spojím ho s S a vypočítam priesečník tejto priamky s rovinou xz = B'



36. Uveďte a popíšte aspoň jeden typ nelineárnej premietacej transformácie používanej v počítačovej grafike.

RYBIE OKO (NELINEÁRNA TRANSFORMÁCIA)

- y - vzdialenosť snímku svetelného lúča a prechádzajúceho bodom O, od hlavného bodu H
- f - ohnisková vzdialenosť objektívu
- β - uhol, ktorý lúč a vytvára s optickou osou o objektívu. - = pohľadový uhol kamery
- 3 typy
 - ortografický - $y = f \cdot \sin(B)$... najčastejšie
 - rovnoploché - $y = 2 \cdot f \cdot \sin(B/2)$... najmenej používané
 - ekvidistančný - $y = f \cdot B$



37. Charakterizujte krivky používané v počítačovej grafike. 1D krivkové útvary.

- 1. a 2. vrstva vizualizačného procesu

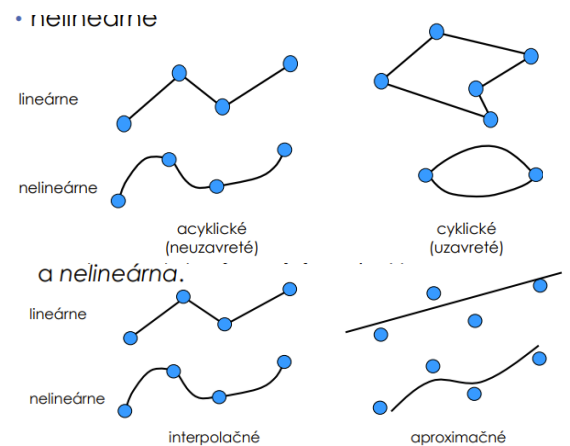
- krivky:

- krivky dané analytickým popisom
- interpolačné krivky
- aproximačné krivky
- použitie kriviek a plôch:
 - geometrická reprezentácia - sú súčasťou vykresľovaných objektov
 - riadiaca funkcia - napr. trajektória lietadla

Krivkové 1D útvary:

1. POLYLINE:

- def. vrcholmi a hranami (segmentami)
 - neuzavretý (acyklický) - min. 2 vrcholy
 - uzavretý (cyklický) - min. trojuholník
- podľa typu hrán (segmentov):
 - lineárne - ! lineárny vtedy len ak všetky segmenty sú lineárne !
 - nelineárne
- podľa vplyvu vrcholov na tvar krivky:
 - interpolačné - vrcholy súčasťou
 - aproximačné - vrcholy nemusia byť súčasťou, ale vplyvajú na jej tvar

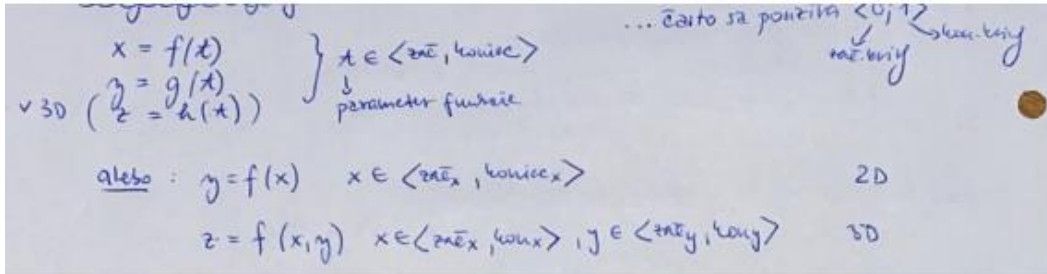


POUŽÍVANÉ KRIVKY:

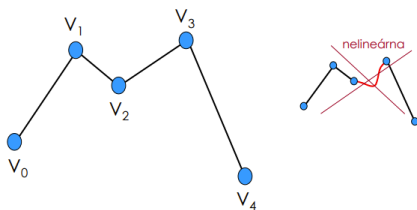
- lineárne: -lineárna interpolácia, lomená čiara - najjednoduchšie
- nelineárne:
 - Bezierove krivky stupňa n
 - racionálne Bezierove krivky stupňa $n - B$
 - spline krivky stupňa n
 - kubické B-spline krivky
 - uniformné
 - neuniformné
 - racionálne B-spline krivky stupňa n - najzložitejšie

- modifikovateľnosť kriviek:
 - zmena polohy riadiacich vrcholov
 - zmena váh riadiacich vrcholov
 - modifikácia uzlového vektora

KRIVKY DEFINOVANÉ ANALYTICKY:



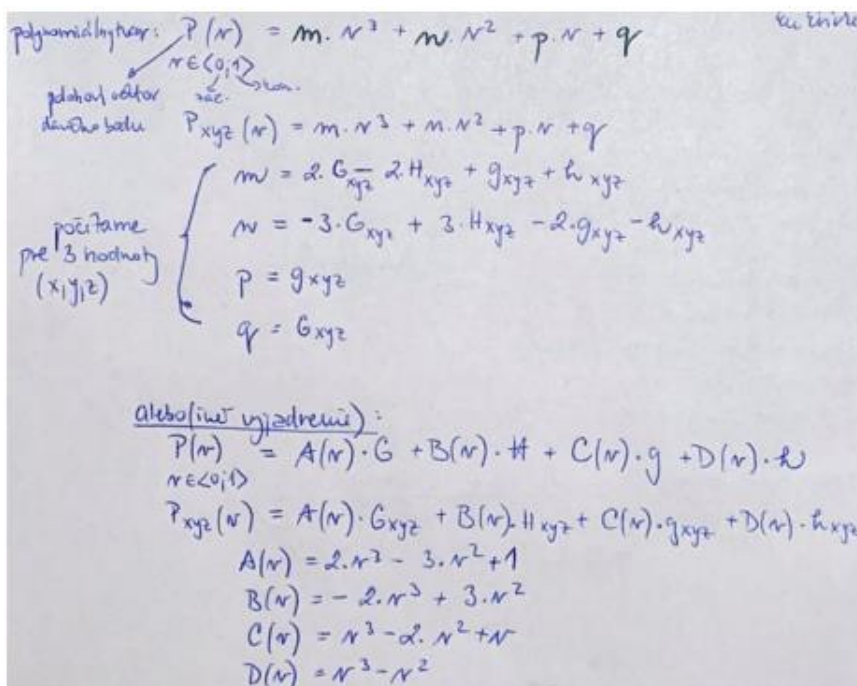
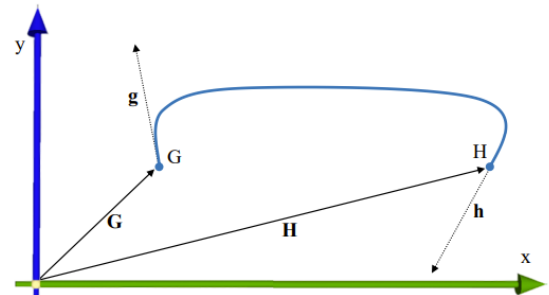
LINEÁRNA INTERPOLÁCIA, LOMENÁ ČIARA:



38. Charakterizujte a popíšte Fergusonovu krivku.

Fergusonova krivka (interpoláčna krivka)

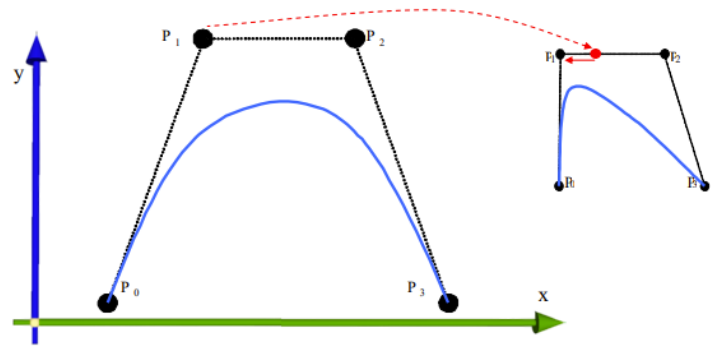
- James Ferguson (1963)
- kridlo Boeingu
- nelinearna
- prili veľka citlivost pre zaoblenie
- def. 4 zkladnymi funkciami:
 - G, H = polohove vektory za. a konca bodu
 - g, h = dotycncovy ˇstartovy, koncovy vektor z bodov G, H



39. Charakterizujte a popíšte Beziérove krivky.

Bézierova kubická krivka (aproximačná krivka):

- Pierre Bezier, 1960s
- Citroen
- kubická
- aproximačná, uniformná
- prechádza cez zač. a koncový bod, zvyšné body určujú zaoblenie
- interpoluje koncové vrcholy
- def. polynomiálnou funkciou stupňa n (n = počet riadiacich vrcholov)
- leží v konvexnom obale riadiacich vrcholov
- pseudolokálna kontrola - ak pohnem vrcholom, krivka sa zmení len v jeho okolí
- afinná invariancia - po aplikovaní afinných transformácií zachováva tvar
 - napr. ak chcem krivku otočiť, stačí mi otočiť riadiace body



Bézierova kubická krivka:

$$P(t) = P_0 \cdot B_0(t) + P_1 \cdot B_1(t) + P_2 \cdot B_2(t) + P_3 \cdot B_3(t)$$

$t \in \langle 0, 1 \rangle$

$$P_{xyz}(t) = P_{0xyz} \cdot B_0(t) + P_{1xyz} \cdot B_1(t) + P_{2xyz} \cdot B_2(t) + P_{3xyz} \cdot B_3(t)$$

$$\left. \begin{aligned} B_0(t) &= (1-t)^3 \\ B_1(t) &= 3 \cdot t \cdot (1-t)^2 \\ B_2(t) &= 3 \cdot t^2 \cdot (1-t) \\ B_3(t) &= t^3 \end{aligned} \right\} \text{pre } t \in \langle 0, 1 \rangle$$

postupne
kubická nř. \downarrow

$$R(t) = \sum_{i=0}^{m-1} P_i \cdot B_{i,m}(t)$$

$P_i \in E^3 (E^2)$
riadiace vrcholy

Bernsteinove polynomy

$$B_{i,m}(t) = \binom{m}{i} \cdot t^i \cdot (1-t)^{m-i}$$

$t \in \langle 0, 1 \rangle$

40. Charakterizujte a popíšte spline a B-spline krivku.

Spline krivka (aproximačná)

- vychádza z Bezierovej
- nemusí začínať a končiť v bodoch
- je ľubovoľne segmentovateľná (Bezier nie je)
- má spojité derivácie - hladko nadväzujúce segmenty
- najčastejšie kubické spline funkcie
- namiesto Bernsteinových polynómov sa používajú koncové polynómy
- Spline funkciou stupňa m pre daných $n+1$ bodov $X_i = (x_i, y_i)$, $i = 0..n$, $x_0 < x_1 < \dots < x_n$, nazývame funkciu $f(x)$, pre ktorú na intervale platí:
 - $f(x) = f_k(x)$ na intervale, kde f_k je polynóm stupňa m ,
 - $f(x)$ má spojité derivácie $f(0), f(1), \dots, f(m-1)$.

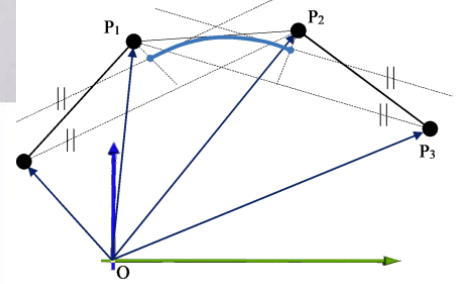
Najčastejšie sa používajú kubické spline funkcie ($m = 3$)

B-SPLINE KRIVKA

B-spline krivky sú zovšeobecnením Beziérových kriviek, miesto Bernsteinových polynómov sa používajú jednoduchšie funkcie.

kubicj Coonsov B-spline : $R(t) = \sum_{i=0}^3 P_i \cdot Co_i(t) ; t \in \langle 0;1 \rangle$
 kde $\left\{ \begin{matrix} P_i \\ i=0 \end{matrix} \right\}$ sú polohové vektory riadiacich vrcholov pre $i = 0-3$

$$\left. \begin{aligned} Co_0(t) &= -\frac{1}{6} \cdot t^3 + \frac{1}{2} \cdot t^2 - \frac{1}{2} \cdot t + \frac{1}{6} \\ Co_1(t) &= \frac{1}{2} \cdot t^3 - t^2 + \frac{2}{3} \\ Co_2(t) &= -\frac{1}{2} \cdot t^3 + \frac{1}{2} \cdot t^2 + \frac{1}{2} \cdot t + \frac{1}{6} \\ Co_3(t) &= \frac{1}{6} \cdot t^3 \end{aligned} \right\} \text{pre } t \in \langle 0;1 \rangle$$



- hladšie ako Bezier
- ťažšie na ovládanie ako Bezier
- aproximačná, uniformná
- špeciálne zadané vstupné hodnoty (násobnosti) spôsobia, že krivka začína v bode P0 a končí v bode P_{L+n-1}, kde n = počet riadiacich vrcholov a L = počet násobností riadiacich vrcholov
- pseudolokálna kontúra
- segmentovateľnosť - možné rozbiť na viac kriviek nižšieho rádu (spojité derivácie)
- afinná invariancia - (po aplikovaní afinných transformácií zachováva tvar)

41. Charakterizujte plochy používané v počítačovej grafike. 2D plošné útvary.

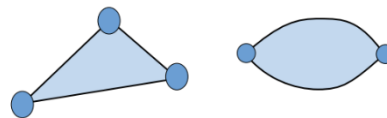
PLOCHY POUŽÍVANÉ V PG

- Plochy dané analytickým popisom
- interpolačné plochy
- aproximačné plochy

PLOŠNÉ 2D ÚTVARY

- TYPY:

polygón: - 2D útvar definovaný plochami



- lineárny -- všetky segmenty musia byť lineárne, - najjednoduchší lineárny - trigón
- nelineárny -- najjednoduchší nelineárny - digón

POUŽÍVANÉ PLOCHY:

- Lineárne:
 - Rovinné (pravítkové) plochy
- Nelineárne:
 - Bézierove plochy
 - B-spline plochy
 - Racionálne B-spline plochy
 - Uniformné
 - Neuniformné

MODIFIKOVATEĽNOSŤ PLÔCH

- Zmena polohy riadiacich vrcholov
- Zmena váh riadiacich vrcholov
- Modifikácia uzlových vektorov

Plochy def. Analyticky:

$$\left. \begin{aligned} x &= f(t) \\ y &= g(t) \\ z &= h(t) \end{aligned} \right\} t \in \langle t_{\min}, t_{\max} \rangle$$

alebo: $z = f(x, y) \quad x \in \langle x_{\min}, x_{\max} \rangle, y \in \langle y_{\min}, y_{\max} \rangle$

- príklady prav.
analytických plôch:
guľovitá, elipsovité

- príklady použitia kriviek:

Morphing vs Warping

- morphing - proces morfovania (prechod medzi tvarmi)
- warping - prebratie tvaru (cieľový tvar)
- stačí mi morfovať len riadiace vrcholy

Nelineárne alfa -miešanie vo farebnom priestore

- farebné prechody
- dobre je používať hlavne interpolačné krivky

Spracovanie trajektórií

- modelovanie pohybov osôb
- inverzná kinematika

- príklady použitia plôch:

Povrchy - digitálny terén

3D skenovanie terénu

3D skenovanie a 3D tlač

42. Charakterizujte a popíšte Coonsovu bilineárnu plochu.

BILINEÁRNA COONSOVA PLOCHA

- Pokiaľ protiľahlé strany budú úsečky, dostaneme tzv. priamkovú (pravítkovú) plochu. Coonsova bilineárna plocha je teda všeobecnejšia ako plocha priamková.

- 9 riadiacich bodov
- 2 súradnice: u, w
- výpočet: dvojitý cyklus + pre všetky súradnice (x, y, z)
- hlavná vypuklosť: bod $P(u, w)$ v strede - G je len 1

$[1-u, -1, u] * \mathbf{M} * [1-w, -1, w]^T = 0$ tvar sa prevedie na explicitný, čím dostaneme:

kde:
 u a w sú parametre ($u \in \langle 0, 1 \rangle, w \in \langle 0, 1 \rangle$), pretože bilineárny plát je určený okrajom $P(u, 0), P(u, 1), P(0, w)$ a $P(1, w)$

$$P(u, w) = [1-u, u] * [P(0, w), P(1, w)]^T + [P(u, 0), P(u, 1)] * [1-w, w]^T - [1-u, u] * \mathbf{Q} * [1-w, w]$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{P}_{xyz}(0,0) & \mathbf{P}_{xyz}(0,w) & \mathbf{P}_{xyz}(0,1) \\ \mathbf{P}_{xyz}(u,0) & \mathbf{P}_{xyz}(u,w) & \mathbf{P}_{xyz}(u,1) \\ \mathbf{P}_{xyz}(1,0) & \mathbf{P}_{xyz}(1,w) & \mathbf{P}_{xyz}(1,1) \end{bmatrix}$$

kde \mathbf{Q} je matica: $\mathbf{Q} = \begin{bmatrix} \mathbf{P}_{xyz}(0,0) & \mathbf{P}_{xyz}(0,1) \\ \mathbf{P}_{xyz}(1,0) & \mathbf{P}_{xyz}(1,1) \end{bmatrix}$

43. Charakterizujte a popíšte Beziérovú bikubickú plochu.

Základná Bézierova plocha je nazývaná aj Beziérová bikubická plocha a je daná maticou 4x4 bodov, teda 16-imi uzlami B_{ij} , $i,j=0,1,2,3$. Plocha je následne definovaná explicitnou rovnicou:

$$z = [E(u), F(u), G(u), H(u)] * \mathbf{B} * [E(w), F(w), G(w), H(w)]^T$$

kde $E(t)$, $F(t)$, $G(t)$, $H(t)$ sú kubické Bernsteinove polynómy definované nasledovne:

$$E(t) = (1-t)^3$$

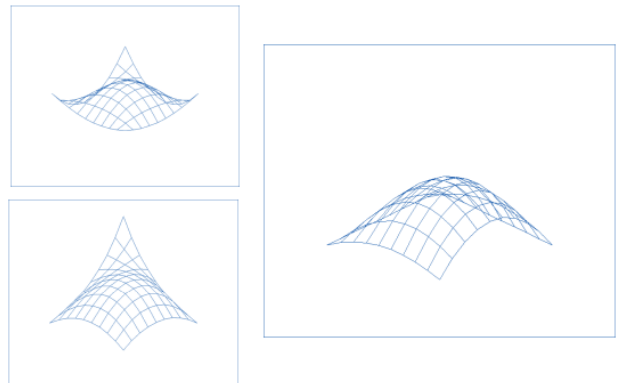
$$F(t) = 3 \cdot t \cdot (1-t)^2$$

$$G(t) = 3 \cdot t^2 \cdot (1-t)$$

$$H(t) = t^3$$

\mathbf{B} je matica vrcholov radiacej siete:

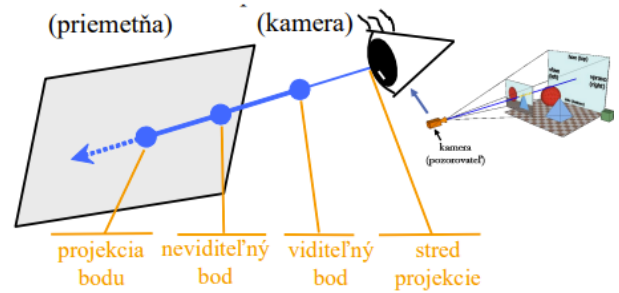
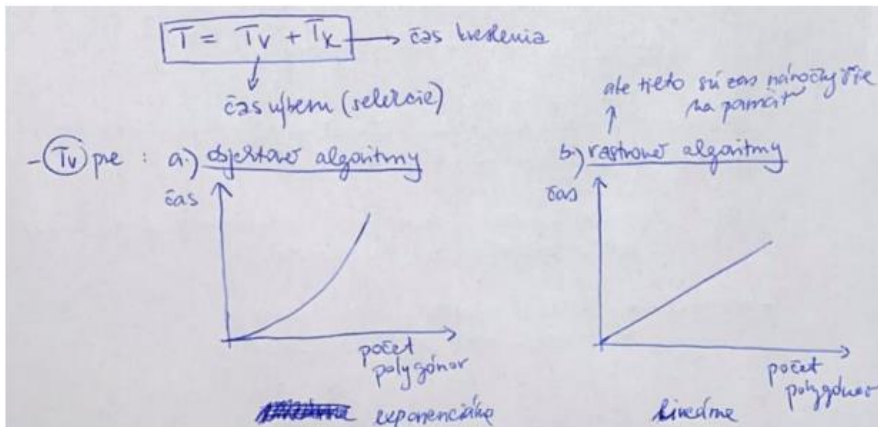
$$\mathbf{B} = \begin{bmatrix} B_{00,xyz} & B_{01,xyz} & B_{02,xyz} & B_{03,xyz} \\ B_{10,xyz} & B_{11,xyz} & B_{12,xyz} & B_{13,xyz} \\ B_{20,xyz} & B_{21,xyz} & B_{22,xyz} & B_{23,xyz} \\ B_{30,xyz} & B_{31,xyz} & B_{32,xyz} & B_{33,xyz} \end{bmatrix}$$



44. Charakterizujte problém riešenia viditeľnosti v rámci počítačovej grafiky.

- 3. vrstva vizualizačného procesu
- už dnes HW support - ale často problém s inými ako pravouhlými oblasťami
- Spočíva v odstránení (resp. odlíšení) tých častí trojrozmerných objektov, ktoré pri danom premietaní do 2D nie sú z miesta pozorovateľa viditeľné. Tým sú tieto časti akože zakryté a dostávame jednoznačné priemety želaných telies.
- KATEGORIZÁCIA:
 - podľa priestoru, kde je viditeľnosť riešená:
 - riešenie v 3D
 - riešenie v 2D priemetne
 - podľa reprezentácie objektov, ktorých viditeľnosť riešime:
 - objektovo orientované algoritmy (kde sa rieši, ktorá časť príslušného objektu je viditeľná)
 - obrazovo orientované algoritmy (kde sa rieši spätne pre každý obrazový bod, ktorý objekt je v ňom vidieť) - - rieši sa to pri premete do 2D
 - podľa toho či sa uvažuje aj osvetlenie telesa:
 - riešenie bez osvetlenia (vyhodnotenie farieb je aplikované lokálne na každý objekt)
 - riešenie s osvetlením (medzi tieto radíme aj metódu sledovania lúča (raytracing) alebo vyžarovaciu metódu (radiosity) = globálne metódy riešenia viditeľnosti s globálnou aplikáciou farieb v rámci scény t.j. napr. aj odrazy)
 - podľa vplyvu možnej chyby pri vykonávaní (napr. použitím celočíselnej aritmetiky):
 - s lokálnym vplyvom chyby na výsledok
 - s globálnym vplyvom chyby na výsledok (hl. globálne metódy)
 - podľa času potrebného na riešenie viditeľnosti:
 - riešenie mimo reálneho času
 - riešenie v reálnom čase

ČASOVÁ ZÁVISLOSŤ RIEŠENIA VIDITEĽNOSTI



- s narastajúcim počtom polygónov skôr narastá problematika výberu ako kreslenia
- o čom je viditeľnosť? - ak máme nejaké body, ktoré sa prekrývajú má byť pre kameru viditeľný ten, ktorý je k nej najbližšie (ta farba musí byť)
- väčšinou sa natiahne úsečka od kamery k scéne a na nej sa hľadajú body - počíta sa priesečník priamky s objektami

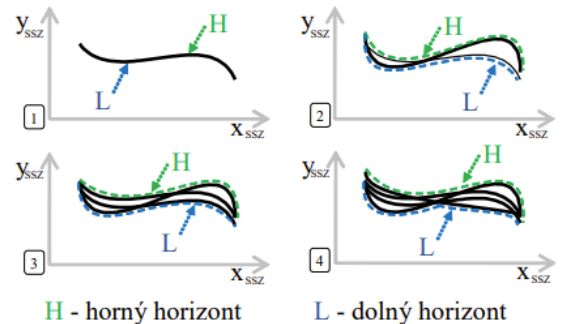
45. Uvedte postup pri získaní horizontu v rámci algoritmu plávajúceho horizontu.

ALGORITMUS PLÁVAJÚCEHO HORIZONTU

- rieši sa viditeľnosť až v priemete, nie je úplne 100%
- účinný pre kreslenie grafov funkcií (v zmysle plôch, nie kriviek)

- kroky:

- pri vykresľovaní plôch máme u a w
- zvolím si, ktoré vykresľovanie (či u alebo w) bude dominantné
- budem to kresliť v krivkových rezoch - začnem od prednej
- pre každé X (stĺpec obrazovky) sa určí, kde je horný a dolný horizont
 - pri 1. reze je to totožné (počiatočný horizont)
- pokračujem ďalším rezom (prestavím H a D horizont ak treba)
- toto sa robí pre každý rez
- ak je niektorá hodnota medzi H a D horizontom -> bod nekreslim (bod nebude viditeľný) a hodnoty horizontov sú nezmenené
- ostávajú viditeľné len body H a D horizontu
- nepatrí medzi algoritmy v reálnom čase
- ľahko implementovateľný



46. Charakterizujte a popíšte maliarov algoritmus riešenia viditeľnosti.

MALIAROV ALGORITMUS

- objektovo orientovaný algoritmus
- zložitosť narastá exponenciálne ($O(n \cdot \log 2n)$) v najlepšom prípade)
- používa sa tu algoritmus triedenia Z-sort
- kroky:

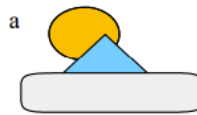
1. zoradiť objekty v scéne podľa vzdialenosti od pozorovateľa

2. kresli všetky objekty odzadu dopredu

- má problémy

- ako zoradiť niektoré objekty?

- nevie riešiť cyklický prekryv



Modelové scény pre maliarov algoritmus bez a s cyklickým prekryvom

Podmienky:

• Ak $z_{1max} < z_{2min}$, potom je polygón P1 úplne za polygónom P2.

• Ak sa neprekrývajú priemety polygónov v rovine priemietne (t.j. v SSZ), je poradie ich vykresľovania nezávislé a polygón P1 testu vyhovuje.

• Ak sú všetky vrcholy polygónu P1 pod rovinou určenou polygónom P2, potom P1 leží za polygónom P2.

• Ak sú všetky vrcholy polygónu P2 nad rovinou určenou polygónom P1, potom P1 leží za polygónom P2.

47. Charakterizujte a popíšte Freeman-Lotrellov algoritmus riešenia viditeľnosti.

- jeho dôsledky sa využívajú aj inde, nielen pri riešení viditeľnosti

- pracuje v 3D

- kroky:

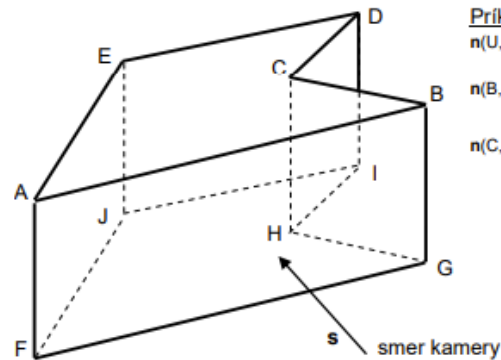
1. rozdelí steny objektov na neviditeľné a potenciálne viditeľné

2. na základe uhla φ medzi vektorom pohľadu kamery a normálou steny určí potenciálnu viditeľnosť

- ostrý uhol - otočené ku kamere - viditeľné

- tupý uhol - otočené od kamery

- podmienka: rovnaká orientácia všetkých stien v scéne (v smere hod. ruč/proti)



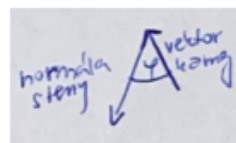
Príklad typu stien:

$n(U,V,X,Y)$ -normála príslušnej steny

$n(B,C,H,G) \cdot s > 0$ zadná (neviditeľná)

$n(C,D,I,H) \cdot s < 0$ predná (potenciálne viditeľná)

$$\varphi = \arccos \left(\frac{a_x \cdot b_x + a_y \cdot b_y + a_z \cdot b_z}{\sqrt{a_x^2 + a_y^2 + a_z^2} \cdot \sqrt{b_x^2 + b_y^2 + b_z^2}} \right)$$



48. Charakterizujte a popíšte algoritmus pamäte hĺbky (Z-buffer).

- v súčasnosti najviac používaný

- obrazovo orientovaný - lineárny čas

- 100% vždy

- vie robiť aj veci ako pohľad cez okno a

pod. (oproti iným), už je integrovaný v grafických kartách na laptopoch

- výhody:

• korektne rieši viditeľnosť

• polygóny (mnohouholníky) sa môžu navzájom pretínať

• polygóny je možné kresliť v ľubovoľnom poradí (nevyžaduje žiadne predspracovanie ani triedenie polygónov)

• vhodný pre objekty s množstvom malých polygónov

všeobecná rovnica roviny mnohoúhelníka je:

$$a \cdot x + b \cdot y + c \cdot z + d = 0.$$

$$z_i = \frac{-a}{c} \cdot (x_i - x_0) + \frac{-b}{c} \cdot (y_i - y_0) + z_0$$

-nevýhody:

- vysoké nároky na pamäť, minimálne (2.maxX .maxY) bajtov
- prekresľovanie
- je pomalý (málo efektívny) pri veľkých scénach (veľa veľkých polygónov), je ho možné však urýchliť použitím orezania na zorný ihlan.

- postup:

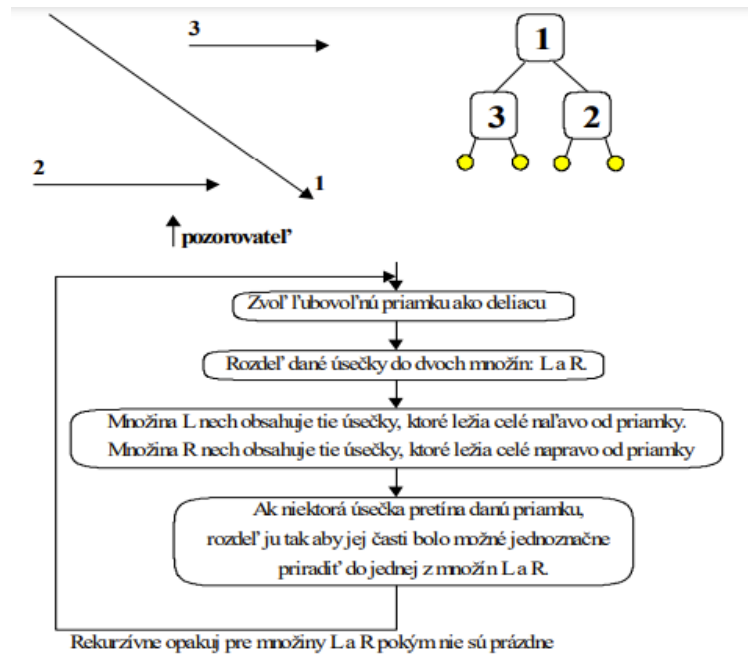
- mám kameru, mám premietňu
- počet vrchnutých lúčov je daný rozlíšením SSZ (suradnicovej sústavy zariadenia)
- vypočítame priesečníky - porovná rovnicu úsečky s rovnicou plochy
- vypočíta sa hĺbka cez Pytagorovu vetu - väčšinou vzdialenosť objektu od priemetne
- potrebuje 2 matice:
 - matica farby
 - matica hĺbky
- na začiatku inicializácia matíc (hĺbka na nekonečno a farba na farbu pozadia)
- rozlíšenie matíc je rovnaké a dané rozlíšením SSZ
- pre všetky body premietne počíta priesečníky s objektami
 - bod s najnižšou hĺbkou je zapísaný do matíc
- na záver sa matica farby využije na vyrenderovanie výsledku

49. Charakterizujte metódu **BSP stromov** pri riešení viditeľnosti v rámci počítačovej grafiky.

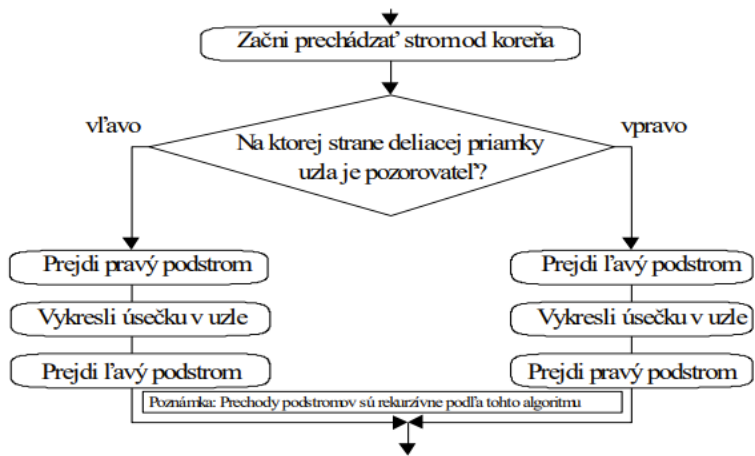
- stromy kt. binárne delia priestor
- podľa rozloženia objektov - stromy:
 - vyvážené
 - nevyvážené
- podľa rozloženia deliacich rovín:
 - statické - strom sa raz vypočíta na začiatku a potom sa takto používa celý čas
 - dynamické - môže dochádzať k úpravam stromu
 - adaptívne - je možný posun roviny
 - stratové - vie zanedbať veci
- 2 prechody stromom:
 - od koreňa k listom
 - od listov ku koreňu
- vie riešiť cyklické scény
- nevie riešiť špecifické veci (napr. kútové spájanie)

50. Popíšte tvorbu a prechod BSP stromom pri metóde BSP stromov v rámci riešenia viditeľnosti v počítačovej grafike.

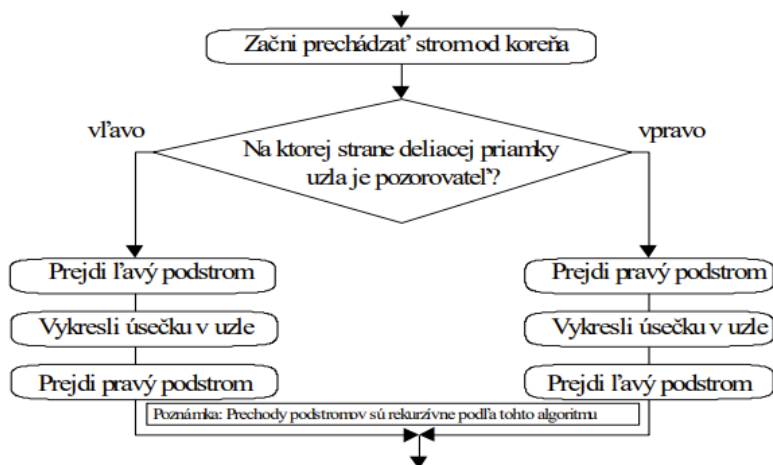
TVORBA



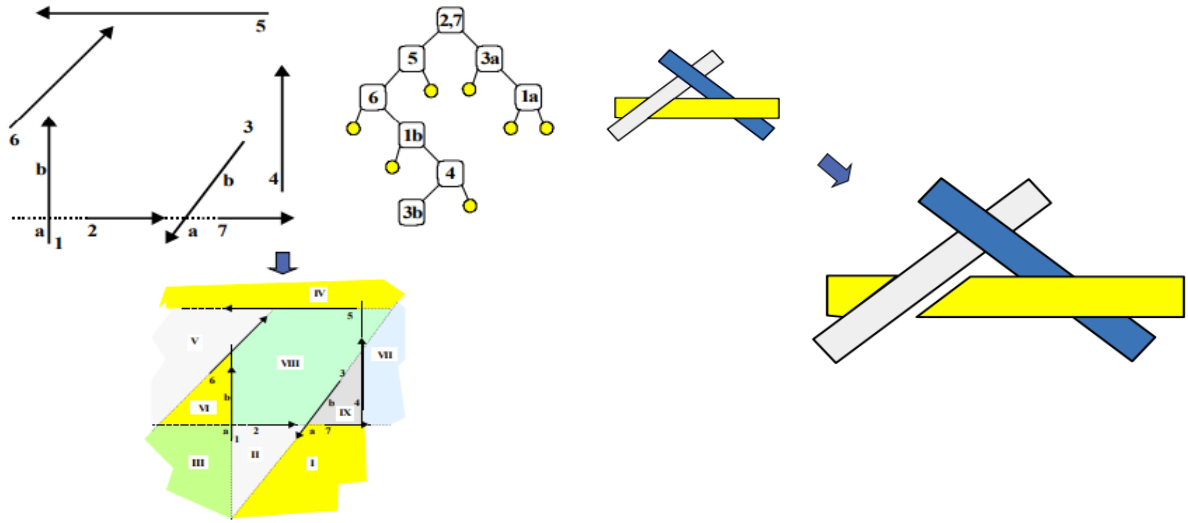
PRECHOD ZOZADU DOPREDU



PRECHOD SPREDU DOZADU



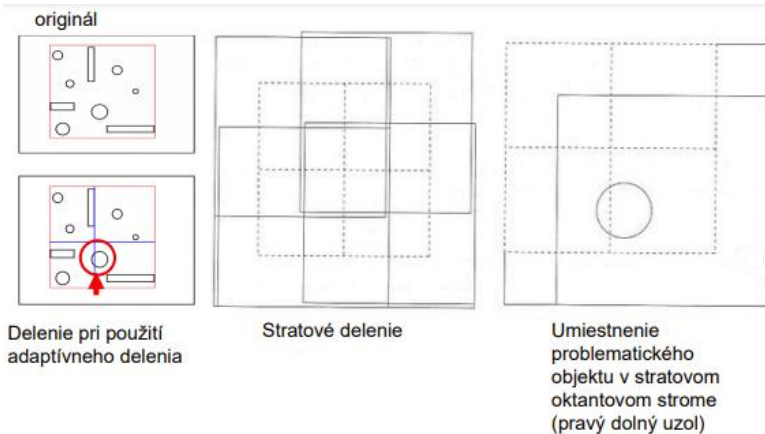
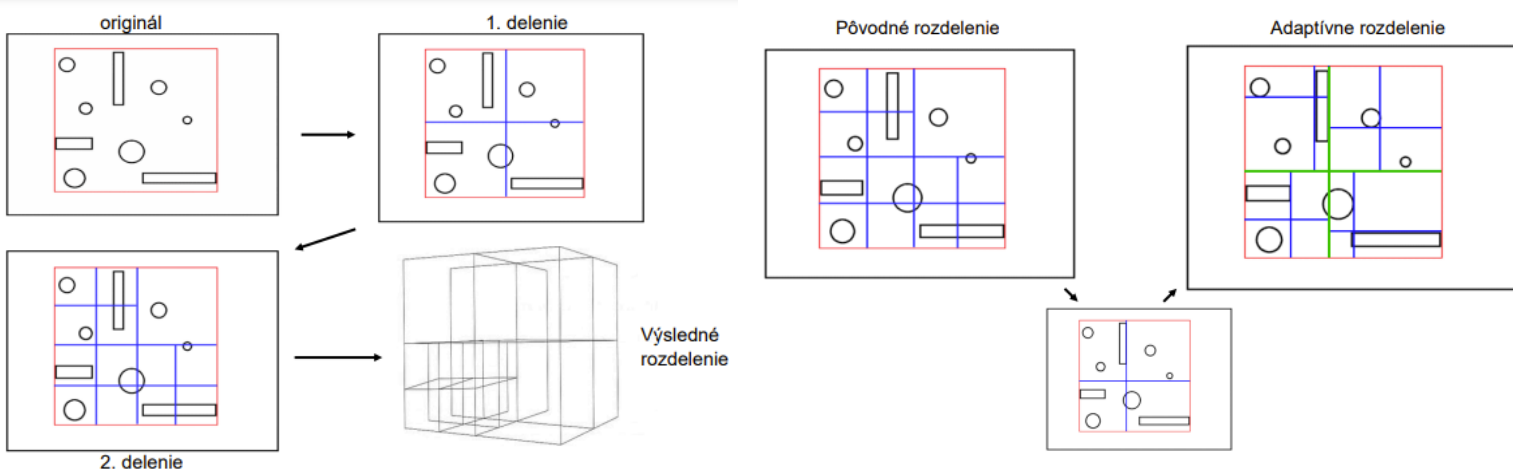
51. Popíšte vlastnosti a použitie BSP stromov v rámci riešenia viditeľnosti v počítačovej grafike.



- riešenie cyklickej scény

52. Charakterizujte metódu oktantových stromov pri riešení viditeľnosti v rámci počítačovej grafiky.

- delenie je ekvivalentné s delením pri BSP stromoch
- roviny deliace sú paralelné s rovinami SS
- statické rozdelenie nemusí byť veľmi dobré - existuje adaptívne a stratové



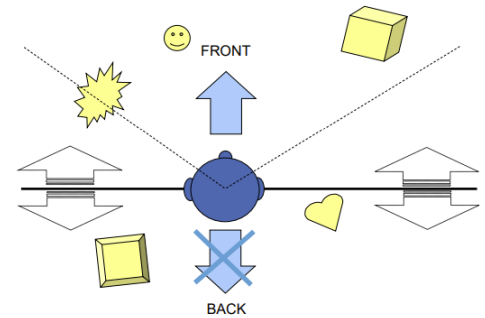
Príklad zložitejšieho delenia priestoru pomocou oktantových stromov

53. Vymenujte a v krátkosti popíšte **urýchľovacie techniky pre riešenie viditeľnosti v počítačovej grafike.**

- dajú sa kombinovať
- znižujú náročnosti výpočtu

- PATRIA TU:

- FV (Front view)/ BC (Back Cut) - asi najjednoduchšie
 - do výpočtu idú len veci, ktoré sú pred pozorovateľom
- Orezávanie na zorný ihlan
- Ohraničujúce útvary
- Sektorovanie
- Potenciál viditeľnosti
- S-buffer

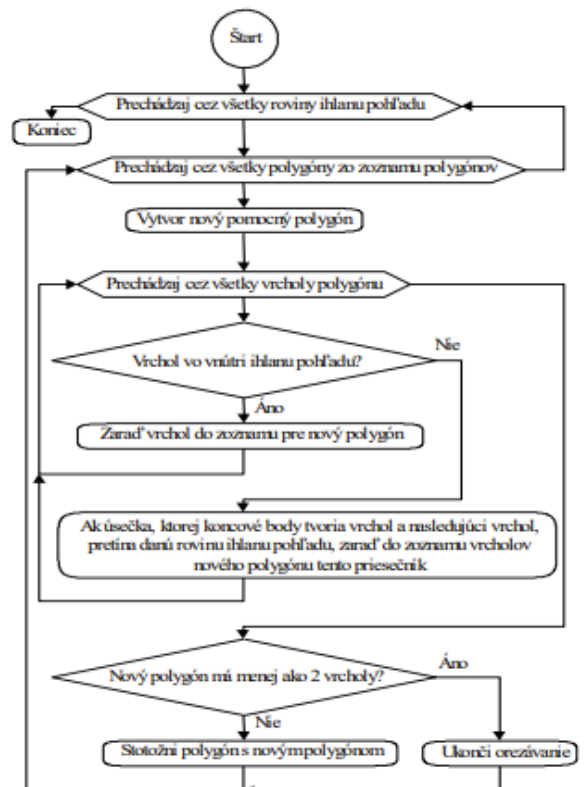


Ohraničujúce útvary

- objekty uzatvoríme do geometricky jednoduchších útvarov a počítame priesečníky s nimi
 - ak nie je priesečník nie je ani s pôvodným teleso
- ak je priesečník už musíme riešiť pôvodný objekt
- aj skupiny objektov môžeme uzavrieť
- používame ohraničujúce útvary: (znižujúca rýchlosť výpočtu, zvyšujúca presnosť)
 - hierarchia gule
 - osovo-orientovaný kváder (AABB)
 - objektovo-orientovaný kváder (OBB)
 - konvexná obálka (convex hull)

54. Popíšte spôsob **orezávania na zorný ihlan** pri riešení viditeľnosti v rámci počítačovej grafiky.

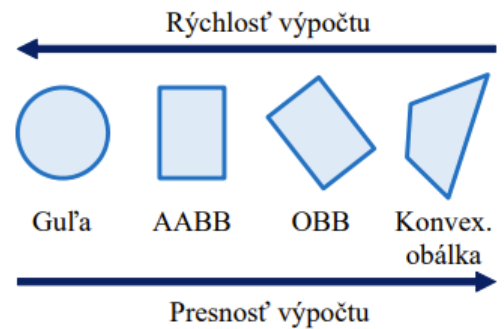
- Zorný ihlan je časť priestoru, ktorá je vymedzená H/VFOV a po projekčnej transformácii (SSC→SSZ) zobrazí do obdĺžnika na priemetni, ktorý reprezentuje tú jej časť, ktorá v konečnom dôsledku bude viditeľná na zobrazovači (obrazovke) t.j. zorný priestor.
- Orezávanie je možné robiť buď v SSC alebo v USS.



55. Porovnajete technológiu ohraničujúcich útvarov, sektorovania a potenciálu viditeľnosti pri urýchľovaní riešenia viditeľnosti v rámci počítačovej grafiky.

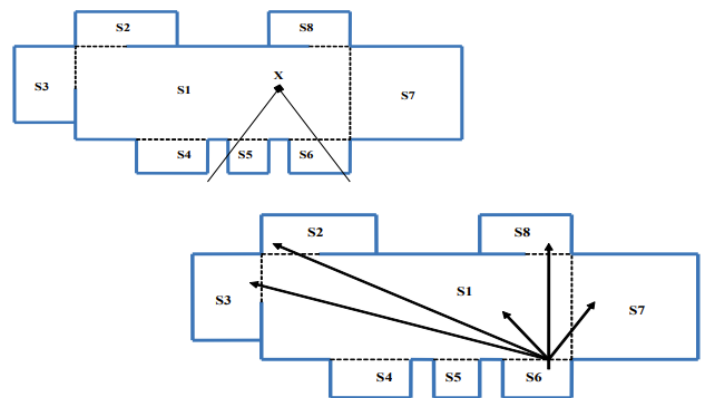
Ohraničujúce útvary

- objekty uzatvoríme do geometricky jednoduchších útvarov a počítame priesečníky s nimi
 - ak nie je priesečník nie je ani s pôvodným teleso
 - ak je priesečník už musíme riešiť pôvodný objekt
 - aj skupiny objektov môžeme uzavrieť
 - používame ohraničujúce útvary: (znižujúca rýchlosť výpočtu, zvyšujúca presnosť)
 - hierarchia gule
 - osovo-orientovaný kváder (AABB)
 - objektovo-orientovaný kváder (OBB)
 - konvexná obálka (convex hull)



SEKTOROVANIE A POTENCIÁL VIDITEĽNOSTI

- 2 techniky, ktoré k sebe patria
- sektorovanie: scénu rozdelíme na sektory
- viditeľnosť: podľa toho, kde je pozorovateľ určíme ktoré sektory majú potenciál viditeľnosti
- PC hry - dvere, portály
- PVS = Potential Visible Set
 - určuje sa, ktoré sektory sú potenciálne viditeľné z daného sektora
 - stačí že 1 bod je viditeľný - sektor je v PVS
 - potom sa uvažuje už len so sektormi ktoré patria do PVS



56. Charakterizujte vyplňovanie oblastí používané v počítačovej grafike.

- 4, 6 a 7 vrstva vizualizačného procesu
- typy vyplňovania:
 - vyplňovanie oblasti 1 farbou - najjednoduchšie
 - vyšrafovanie oblasti - vyplnenie ČB vzorom
 - vyplnenie farebným vzorom (textúrovanie)
- Rozdelenie algoritmov vyplňovania:
 - podľa spôsobu akým je zadaná hranica vyplňanej oblasti:
 - hranica definovaná geometricky
 - hranica nakreslená na zobrazovači
- Metódy vyplňovania:
 - METÓDA RIADKOVÉHO ROZKLADU
 - INVERZNÉ VYPLŇOVANIE – RIADKOVÉ A PLOTOVÉ
 - VYPLŇOVANIE SPEKTROM
 - SEMIENKOVÉ VYPLŇOVANIE – REUKRZÍVNE A NEREKURZÍVNE

57. Charakterizujte a popíšte algoritmus riadkového rozkladu pri vyplňovaní oblastí.

METÓDA RIADKOVÉHO ROZKLADU

- definovaná geometricky (hranica)

-POSTUP:

- postupujem väčšinou od 0. riadku vyššie

- zisťujem priesečník riadka s jednotlivými hranami

- usporiadajú sa priesečníky

- v každom riadku sa nakreslí čiara medzi priesečníkmi vo farbe výplne

NÁJDENIE PRIESEČNÍKOV PRI METÓDE RIADKOVÉHO ROZKLADU

$$x = a_x + \frac{c_y - a_y}{b_y - a_y} \cdot (b_x - a_x)$$

$$y = c_2$$

1. Postupuje sa od najvyššieho vrcholu oblasti k najnižšiemu, smerom zľava doprava v každom riadku.

2. Pre jednotlivé rozkladové riadky rovnobežné s osou X a s konštantnou súradnicou Y, klesajúcou s krokom -1, sa nájdu priesečníky s hranicami vyplňanej oblasti.

3. Vo výslednom zozname sú usporiadané priesečníky zľava doprava a vyfarbené úseky medzi nepárny a párny priesečníkmi. Preto je nutné aby počet priesečníkov bolo párne číslo.

58. Charakterizujte a popíšte vyplňovanie spektrom.

Vyplňovanie spektrom (prechod farieb)

- 2 spôsoby:

- riadkový rozklad

- nastavenie orezávacej oblasti - len pre Windows

- zmena farby:

- lineárna - jednoduchšia

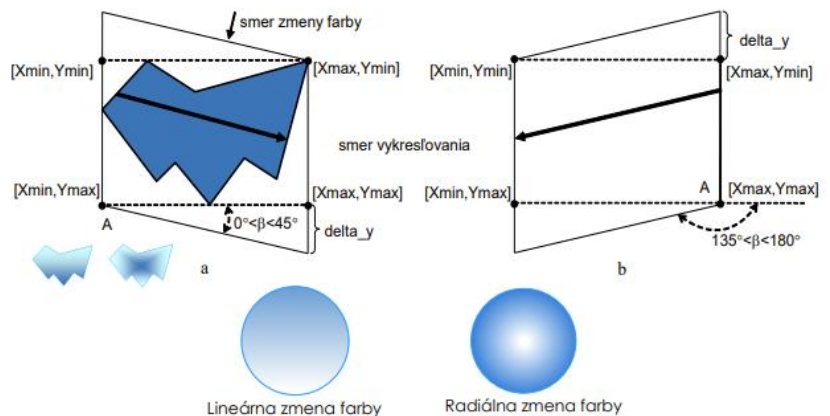
- radiálna

- prechod farieb:

- ekvivalent alfa miešania

- lineárne/nelineárne

- RGB priestor



Prvý spôsob: využíva metódu **riadkového rozkladu** popísanú skôr.

• Na začiatku sa musí otočiť mnohoúholník tak, aby smer vykresľovania priamok bol rovnobežný s osou x, t.j. o uhol β a metódou riadkového rozkladu sa nájdu priesečníky.

• Pri vykresľovaní úsekov medzi priesečníkmi sa musia úseky spätne otočiť o uhol β .

• Rozdiel oproti riadkovej výplni je v tom, že oblasť nie je vyplňaná jednou farbou, ale farba postupne plynule prechádza z prvej farby do druhej.

Druhý spôsob: využíva možnosť **nastavenia orezávacej oblasti**, ktorú poskytuje napr. grafické rozhranie MS Windows.

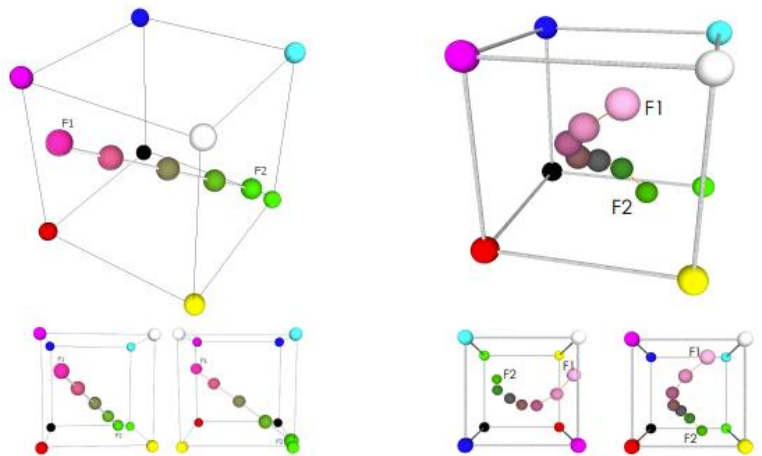
• Na začiatku sa nastaví orezávacia oblasť na celú oblasť mnohoúholníka a vypočítajú sa súradnice X_{min} , X_{max} , Y_{min} , Y_{max} .

• Podľa veľkosti uhla β je vyplnený príslušný rovnobežník a orezávacia oblasť zabezpečí vyplnenie len v oblasti mnohoúholníka.

Plynulý prechod farieb:

- Zistenie rozdielu jednotlivých RGB zložiek prvej a druhej farby.
- Vypočítanie počtu úsečiek v rovnobežníku.
- Vypočítanie prírastkov jednotlivých zložiek ako pomer rozdielu zložiek prvej a druhej farby a počtu úsečiek, napr. pre zložku R
- alebo prechod v RGB modeli

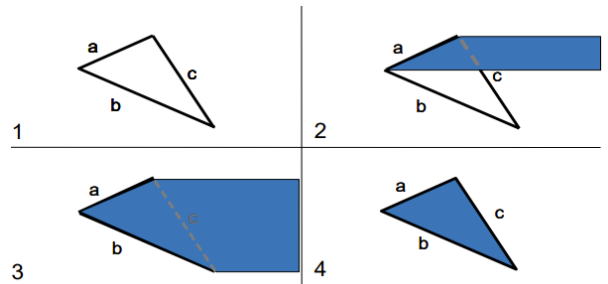
Ekvivalent alfa miešania (lineárne aj nelineárne, RGB priestor)



59. Charakterizujte a popíšte inverzné a plotové vyplňovanie.

INVERZNÉ VYPLŇOVANIE

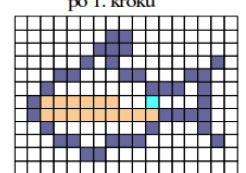
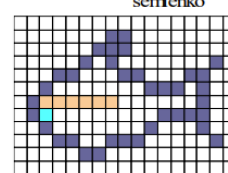
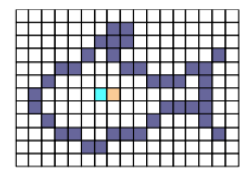
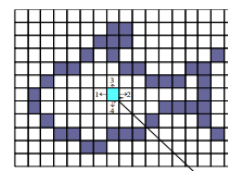
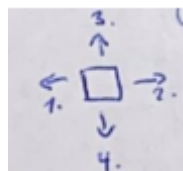
- Využíva metódu XOR ($0 \vee 0 = 1 \vee 1 = 0$, $0 \vee 1 = 1 \vee 0 = 1$)
- Dva typy:
 - o vodorovné (riadkové) vyplňovanie
 - o zvislé (plotové) vyplňovanie - pomalšie (offsetové odskoky)
- krátky algoritmus (5 riadkov v C)
- rekurzívny
- hranica na zobrazovači
- implementovateľný len na zariadeniach ktoré majú dynamický výstup (monitor)
 - o na tlačiareň sa nedá implementovať
 - o je to preto lebo potrebuje mazať (prepisovať obsah)
- postup:
 - o metódou XOR sa zafarbí všetko od hrany po pravý (ľavý) okraj obrazovky
 - o pokračujem s ďalšími hranami
 - o keď už niečo je zafarbené (1) a chcem to znova farbiť (1) --- $1 \text{ XOR } 1 = 0$ (odfarbí sa)
 - o 100% nájde výsledok ALE nie je 100% pri nalezovaní výplne



60. Charakterizujte a popíšte rekurzívne aj nerekurzívne semienkové vyplňovanie.

REKURZÍVNE SEMIENKOVÉ VYPLŇOVANIE

- hranica definovaná na obrazovke
- je pixel vyfarbený farbou hranice? výplne?
 - a nie tak vyplním
- keď ani jeden smer - rekurzívne vraciam
- až kým sa nevrátim k počiatočnému semienku
- potrebuje veľa pamäte, bez úprav je prakticky nepoužiteľný

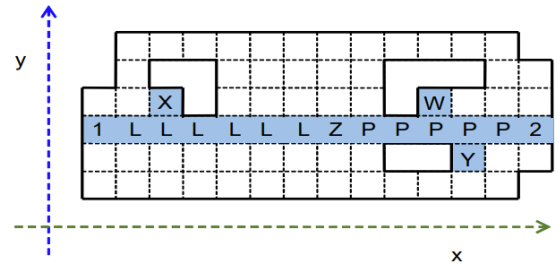


po 6. kroku

po 15. kroku

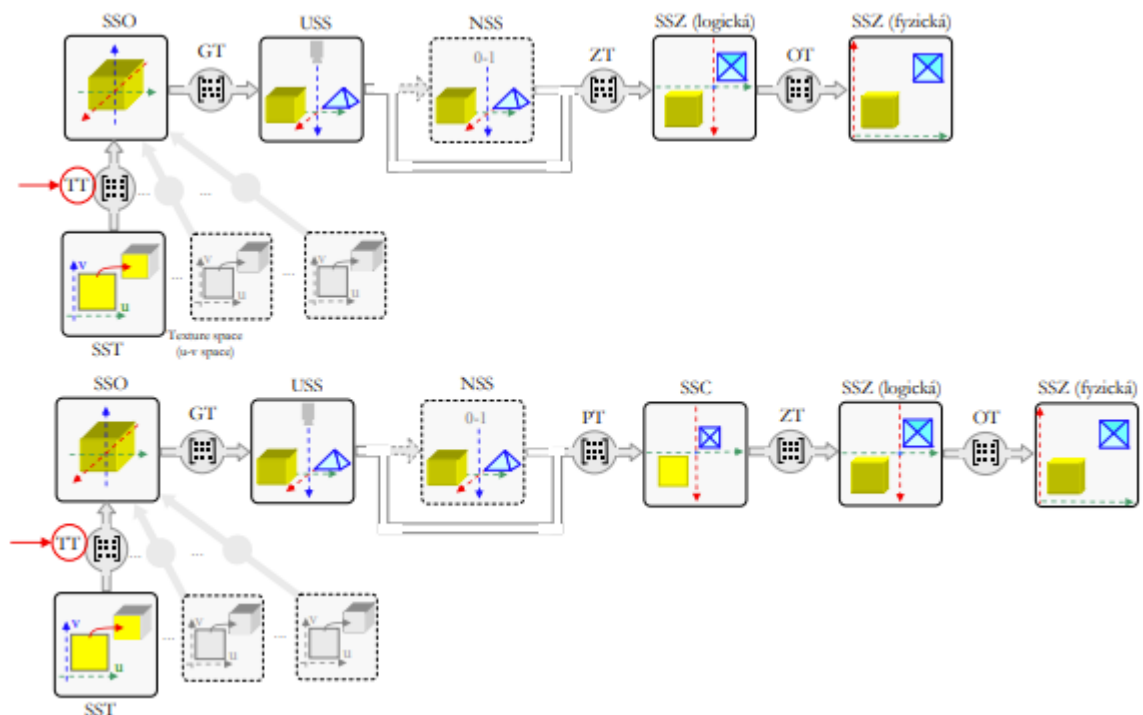
NEREKURZÍVNE SEMIENKOVÉ VYPLŇOVANIE

- ukladajú sa body návratu
- sami si riadime zásobník - 1. voľný bod v ňom je ďalší bod rozvoja
- má menšie pamäťové nároky



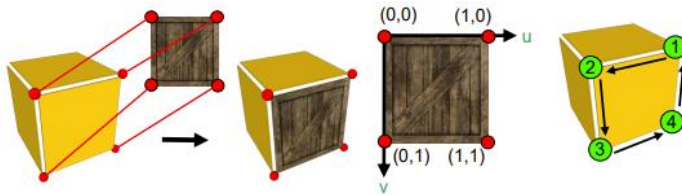
61. Charakterizujte textúrovanie a a jeho vzťah k zobrazovacím reťazcom.

- Proces (Textúrovacia transformácia, TT) nanášania obrazových vzoriek (textúr, tapiet) na povrch objektov za účelom získania vizuálneho dojmu, že objekt je z istého materiálu (napr. drevo, kameň, kov a pod.).
- Typy textúr podľa topologického rozmeru: • 1D, 2D, 3D
- Typy textúr podľa spôsobu nanášania:
 - o Statické
 - o Dynamické
 - o Procedurálne - textúra je vypočítaná podľa nejakej funkcia (napr. fraktálnej)
 - o Animačné - klasická 2D animácia (väčšinou)
- transformačný reťazec:
 - o SST (súradnicová sústava textúry) - naviazaná na SS objektu
 - o textúry sa mapujú na objekt
 - o TT = proces textúrovania
 - o SST - väčšinou 2D sústava
 - o miesto $x, y \rightarrow u, v \rightarrow$ „UV priestor“



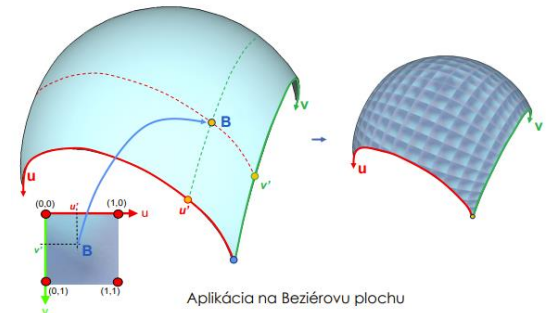
62. Charakterizujte bilinéarne textúrovanie.

- textúrovanie bilinéarne - základný spôsob, UV textúrovanie
- Pri tomto procese nanášania obrazového formátu na plochy trojrozmerného objektu sa najčastejšie používa ako vzor obrázok (textúra).
- Obrazový formát (obrázok, textúra) využíva dvojrozmernú súradnicovú sústavu SST [u,v], ktorej hodnoty súradníc u,v zodpovedajú jednotkovej miere obrazového formátu.
 - Súradnica „u“ zodpovedá x-ovej osi obrázka (textúry),
 - súradnica „v“ zodpovedá y-ovej osi textúry.
 - Jedna jednotka = dĺžka/šírka obrázku textúry. (u/v)



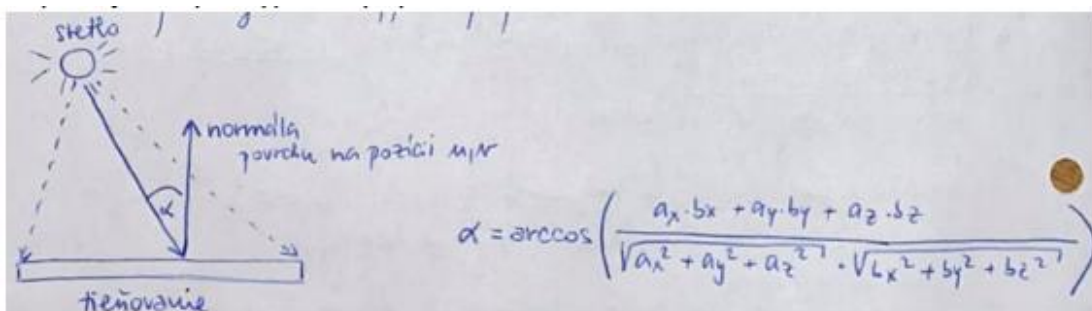
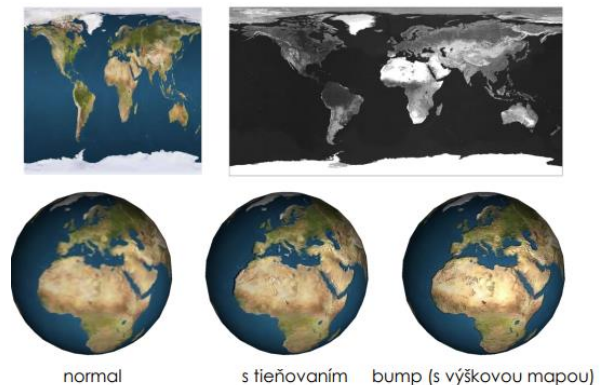
- bilinéarne UV textúrovanie pri aplikácii na Bezierovu plochu (nelineárnu oblasť):

- u a w nie sú lineárne
- súradnica [u,w] je mapovaná na [x,y] súradnicu plochy
- takto sa dá textúra aplikovať na ľubovoľnú plochu
- časová náročnosť vysoká
- ak máme viac plôch treba zabezpečiť aj naväzovanie textúr (1 plocha - bázická)



63. Charakterizujte bump-map textúrovací proces a použitie tieňovania a osvetľovania textúry.

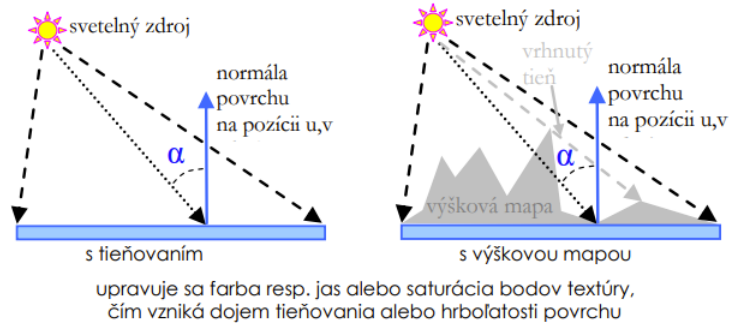
- s výškovou mapou - nad objektom je svetelný zdroj a vplyv svetla je ovplyvnený aj výškou
- s tieňovaním
- upravuje sa farba (jas, saturácia) bodov textúry, dojem hrboľatosti povrchu (zdá sa nám, že je 3D)
- vieme simulovať aj pohyb svetelného zdroja
- vyžaduje sa vyšší výpočtový výkon



Tieňovanie a osvetľovanie

- 4,5 a 6 vrstva vizualizačného procesu (5 a 6 nie je povinné ale vítané)
- proces vplyvu svetelného zdroja na objekt (najčastejšie na jeho povrch)
- ak mám viac zdrojov svetla, musím s tým počítať
- typy tieňovaní:
 - konštantné (flat) (najjednoduchšie, najrýchlejšie)
 - gourardovo – interpolácia farby
 - Phongovo – interpolácia normály (najťažšie, najpomalšie)
- podporné technológie:
 - poltónovanie (halftoning)
 - rozptyľovanie (dithering)

$$\alpha = \arccos \left(\frac{a_x \cdot b_x + a_y \cdot b_y + a_z \cdot b_z}{\sqrt{a_x^2 + a_y^2 + a_z^2} \cdot \sqrt{b_x^2 + b_y^2 + b_z^2}} \right)$$



Konštantné tieňovanie



Gourardovo tieňovanie

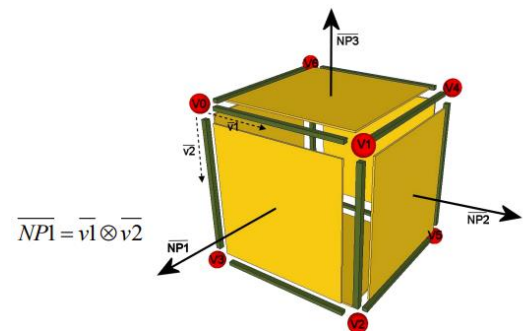
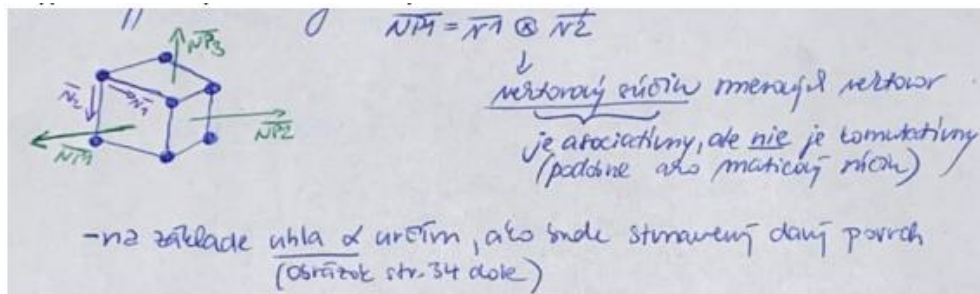


Phongovo tieňovanie

64. Charakterizujte a popíšte konštantné (flat) tieňovanie v rámci počítačovej grafiky.

KONŠTANTNÉ TIEŇOVANIE (FLAT)

- vychádza z predpokladu, že normála, ktorú potrebujem na výpočet je rovnaká na celej ploche
- výpočet normály: z dvoch smerových vektorov hrán

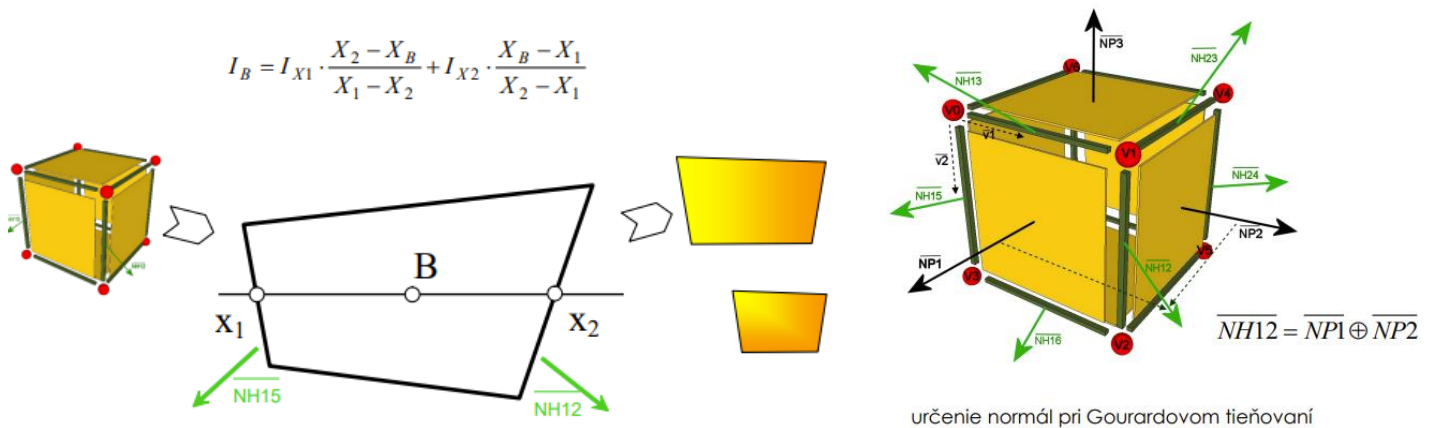


určenie normál pri konštantnom tieňovaní

65. Charakterizujte a popíšte tieňovanie interpoláciou farby (Gourard) v rámci počítačovej grafiky.

Gourardovo tieňovanie (interpolácia farby)

- pracujeme aj s normálami hrán
- kroky:
 1. to isté ako pri konštantnom – vypočítam normály plôch
 2. vypočítanie normály hrán medzi plošnými uzlami (medzi NP1 a NP2) dostanem NH12 (vektorový súčet)
 3. prehlásim, že táto normála je rovnaká pre celú hranu
 4. určí sa intenzita svetla podľa uhla alfa - lineárna interpolácia - intenzita svetla sa určuje podobne ako výplň spektrom farieb



určenie normál pri Gourardovom tieňovaní

určenie farby interpoáciou pri Gourardovom tieňovaní

66. Charakterizujte a popíšte tieňovanie interpoláciou normály (Phong) v rámci počítačovej grafiky.

Phongovo tieňovanie (interpolácia normály)

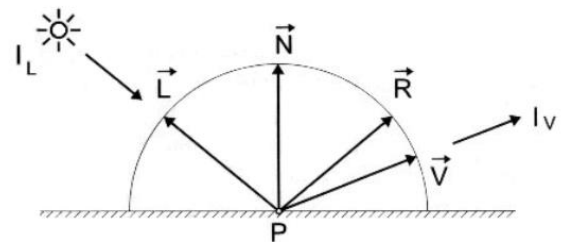
- používa empirický osvetľovací model – Phong, 1977:

- 2 základné zložky svetla:

- difúzna – v zmysle normály N

- reflexná (spekulárna) – v zmysle vektora R

- kombináciu N a R dostávame výslednú intenzitu V (resp. I_v)



Empirický osvetľovací model (Phong, 1977)

- „vrcholové tieňovanie“ – k interpolácií dochádza až smerom k vrcholom

- stupne (postup):

1. rovnaký ako konštantné (výpočet normály plôch)

2. Guardovo (výpočet normály hrán)

3. výpočet vrcholových vektorov (normál): $NV1 = NH13 + NH12 + NH23$... vektorový súčet normál hrán

4. ďalej klasika: výpočet uhla alfa medzi zdrojom svetla a normálou

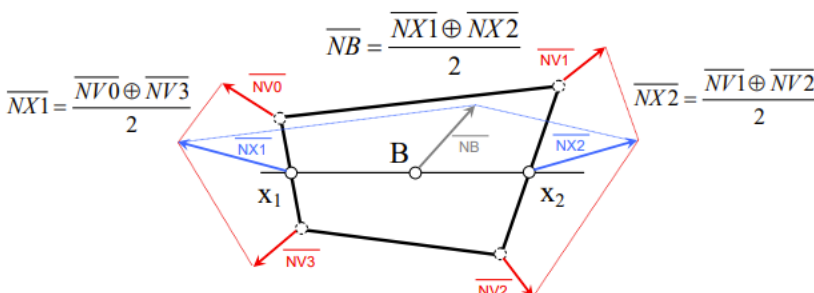
5. určenie intenzity svetla

- interpoluje sa normála každého bodu v zmysle 4 normál, ktoré sú vypočítané na vrchole

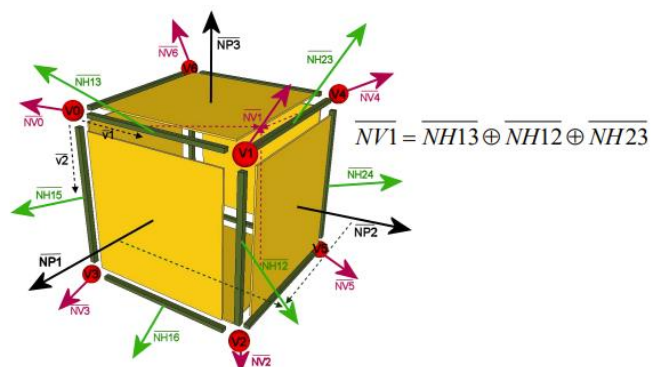
- už nie je jedna normála pre celú hranu

- normála je daná interpoláciou medzi normálami vrcholov: $NX1, NX2$

- veľa výpočtov aj keď sú jednoduché



určenie normál pri Phongovom tieňovaní



určenie normál pri Phongovom tieňovaní

67. Popíšte problematiku osvetľovania a osvetľovacích modelov, svetelné zdroje, osvetľovacie mapy.

OSVETĽOVANIE

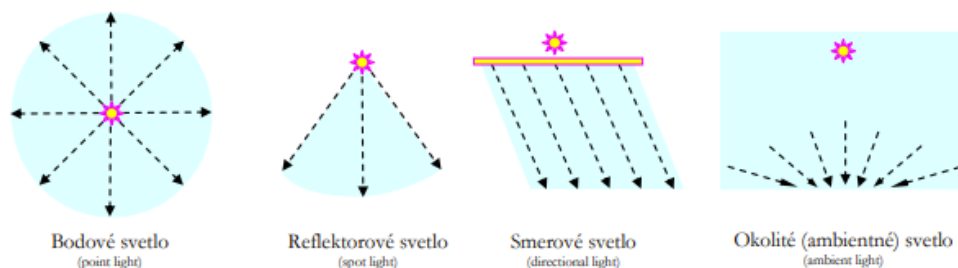
- Proces vplyvu svetelného zdroja/zdrojov, materiálu a iných objektov na svoje okolie resp. na iné objekty – vrhanie tieňov
- typy osvetľovaní:
 - o statické – svetlá sa nehýbu
 - o dynamické – svetlá sa pohybujú – každý pohyb sa musí prejavíť v scéne
- PROCES: 1. Osvetľovacia mapa 2. Dodanie textúr 3. Dodanie svetelných zdrojov

OSVETĽOVACÍ MODEL

- model, ktorým môžeme sledovať vlastnosti povrchu ako je farba, lesklosť, matnosť, drsnosť a podobne
- Základom osvetľovacieho modelu je odrazová funkcia (reflection function).
 - o vyjadruje intenzitu svetelného lúča rozptýleného svetla v závislosti na jeho smere a smere intenzity a vlnovej dĺžky dopadajúceho lúča
 - o Čím lepšie popisuje odrazová funkcia chovanie skutočného svetla, tým lepší je vizuálny vnem

SVETELNÉ ZDROJE

- typy svetelných zdrojov:
 - a) podľa farby: - monochromatické - achromatické (napr. biele) – viacero svetiel
 - b) podľa spôsobu vyžarovania: - bodové - reflektorové (spot) - plošné
 - c) podľa kinematiky: - statické – dynamické



OSVETĽOVACIE MAPY (LIGHTMAPS)

- nastavba - celkový počet vplyvov svetelných zdrojov na scénu

68. Spracovanie osvetľovacích modelov, zložky svetla podľa Phongovho osvetľovacieho modelu.

- zložky svetla (Phong):

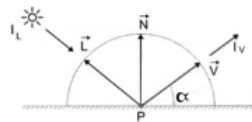
- ambientná zložka (ambient light)
 - hlavne pri empirických odrazových rovniciach
 - celkové okolité svetlo, rozptýlené
 - spravidla achromatické (biele)
 - vznik: mnohonásobné odrazy od všetkých telies, rozptyl – molekuly vzduchu
- difúzna zložka (diffuse)
 - nezáleží na smere pohľadu
 - po viacnásobnom odraze a lome – smer náhodný, rovnaka pravdp všetkých smerov
 - veľkosť zložky je úmerná kosínusu uhla dopadu
 - nesie informácie o farbe povrchu

- zrkadlová zložka (specular)
 - závisí na smere pohľadu
 - veľkosť zložky je závislá od uhla dopadu a optických vlastnostiach povrchu
 - viacnásobný odraz a lom spôsobia útlm lúča
 - pri odraze sa väčšinou riadi štatistickým rozdelením (Gaussova krivka)

Lambertov osvetľovací model – jednoduchý model

• Difúzna zložka

$$I_d = L_L r_d (\vec{L} \cdot \vec{N})$$



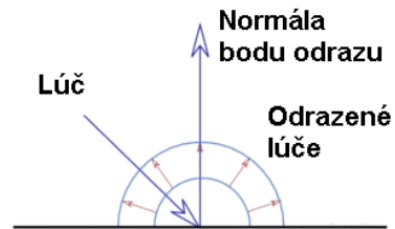
• Ambientná zložka $I_a = I_A \cdot r_a$

• Výsledok:

$$I_v = I_d + I_a$$

$$I_v = I_A r_a + \sum_{k=1}^M I_{L_k} \left[r_d (\vec{L}_k \cdot \vec{N}) \right]$$

Difúzny povrch



→ len difúzna a ambientná zložka
 → zdieľať len od dopadajúceho lúča a normály odrazu
 - difúzna zložka: $I_d = L_L r_d (\vec{L} \cdot \vec{N})$
 - ambientná zložka: $I_a = I_A \cdot r_a$
 - výsledok: $I_v = I_d + I_a$

$$I_v = I_A r_a + \sum_{k=1}^M I_{L_k} \left[r_d (\vec{L}_k \cdot \vec{N}) \right]$$
 ↓
 suma sa dáva, len ak je použitá distribúcia funkcia

Phongov osvetľovací model – zložitejší

- difúzny povrch + zrkadlový povrch = skutočný odraz zložiek svetla ako Lambertov
- výpočet odrazených lúčov – podľa štatistickej funkcie

- difúzna zložka: $I_d = L_L r_d (\vec{L} \cdot \vec{N})$
 - zrkadlová zložka: $I_s = I_L r_s (\vec{V} \cdot \vec{R})^p$
 - ambientná zložka: $I_a = I_A \cdot r_a$
 - výsledok: $I_v = I_s + I_d + I_a$... suma musí dosahovať 100%

$$I_v = I_A r_a + \sum_{k=1}^M I_{L_k} \left[r_s (\vec{V} \cdot \vec{R}_k)^p + r_d (\vec{L}_k \cdot \vec{N}) \right]$$

Difúzny povrch + Zrkadlový povrch = Odražajúce sa zložky diffuse + specular = skutočný odraz zložiek svetla



69. Charakterizujte problém realistického zobrazovania a globálne osvetľovacie techniky v rámci počítačovej grafiky.

FOTOREALISTIKA- čo najvernejšie zobrazenie priestorových scén a objektov vrátane osvetlenia a riešenia viditeľnosti

Globálne osvetľovacie techniky slúžia na riešenie zobrazovacej rovnice (vrátane riešenia viditeľnosti).

- najvyšší stupeň - cez textúrovanie, tieňovanie až po osvetľovanie

- Ich riešením je:

- Výpočet osvetlenia všetkých plôch v scéne – pohľadovo nezávislé.
- Výpočet osvetlenia pre určitý smer – pohľadovo závislé.

- Rozdelenie globálnych osvetľovacích techník:

- Metódy vychádzajúce od pozorovateľa
 - sledovanie lúča
 - sledovanie cesty
- Metódy vychádzajúce od pozorovateľa zdroja
 - sledovanie fotónov
 - Monte Carlo sledovanie svetla
- Obojsmerné metódy
 - Obojsmerné sledovanie cesty
 - Fotónové mapy
- Vyžarovacia metóda (Radiosity)
 - opticky najlepšia
 - ťažká
 - nevie si poradiť so zrkadlami

70. Uvedte a v krátkosti popíšte fotorealistické metódy vychádzajúce od pozorovateľa a od svetelného zdroja.

Od pozorovateľa

- pohľadovo závislé metódy, ktoré zhromažďujú svetelnú informáciu (energiu), ktorú lúč akumuluje (gathering methods) po svojej trajektórii
- spätné sledovanie trajektórie svetla.
- Sledovanie lúča (Ray tracing)
 - základný algoritmus pracuje s bodovými svetlami
 - poskytuje len ostré tieňe
 - transport svetla je obmedzený na zrkadlové odrazy
 - Je schopný popisovať ľubovoľný počet zrkadlových odrazov a lokálny osvetľovací model s difúznou zložkou
 - Nepočíta sa kompletne riešenie zobrazovacej rovnice (napr. efekty ako kaustika, plošné svetelné zdroje, difúzne odrazy a pod.).
- Distribuované sledovania lúča (Distributed ray-tracing)
 - rozšírenie Raytracingu o odraz lúča, aj do difúznych smerov, v závislosti od osvetľovacieho modelu.

Od svetelného zdroja

- energiu vystreľujúce metódy (shooting methods)

- riešia problémy predchádzajúcich metód, ako sú kaustika, skryté svetelné zdroje a pod., avšak za cenu zaťaženia šumom.
- Tieto algoritmy hľadajú riešenie osvetľovacej rovnice náhodným sledovaním dráh lúča zo svetelného zdroja.
- Photon tracing
 - duálna metóda k metóde sledovania lúča
 - rovnaké druhy odrazov no v opačnom smere
 - trajektória sa dá popísať ako $LS \times DE$
 - ľubovoľný počet zrkadlových odrazov zo svetla, ktoré ústia do difúzneho odrazu (závislého napr. od hĺbky rekurzie).
- Light tracing
 - Metóda duálna k metóde sledovania cesty (path tracing)
 - Náhodným spôsobom sa určujú odrazy lúčov v scéne, ktoré začínajú zo svetelného zdroja.
 - Pokiaľ je bod odrazu lúča viditeľný pozorovateľom, určí sa jeho príspevok pre daný pixel (všetky príspevky pre daný pixel sa spremerujú)
 - Trajektória sa ukončí, pokiaľ bude príspevok svetla dostatočne malý.

71. Uveďte a v krátkosti popíšte algoritmy fotorealistických metód.

Algoritmy fotorealistických metód

a) algoritmy zobrazujúce povrchy (surface based)

- nepriama metóda
- vytvárajú pomocnú geometrickú reprezentáciu povrchu
- napr.: - sledovanie obrysov (Contour tracing)
 - marching cubes (pochodujúce kocky)
 - marching tetrahedra (4-steny)
 - dividing cubes - opaque cubes
- snažia sa z objemových dát aproximovať povrch
- informácie o vnútri objektu sa stráca

b) objemové algoritmy

- využívajú plnú priestorovú informáciu na vykreslenie obrazu
- nezávislé od zložitosti scény
- náročné na pamäť a procesor
- používajú sa tu napr. oktánové stroje, binárne stroje.
- Binárne - pokrývajú voxel úplne alebo vôbec
 - povrchovo-orientované
- Pravdepodobnostné - priradujú voxelom % podiel objektu
 - stratové stromy

c) algoritmy pracujúce v obrazovom priestore

- pre každý pixel výsledného obrazu sa hľadajú voxely v objektovom priestore, ktoré prispievajú do výslednej farby bodu
- inteprolácia
- Trasovanie lúčov (raytracing, raycasting)
- Sábellova metóda

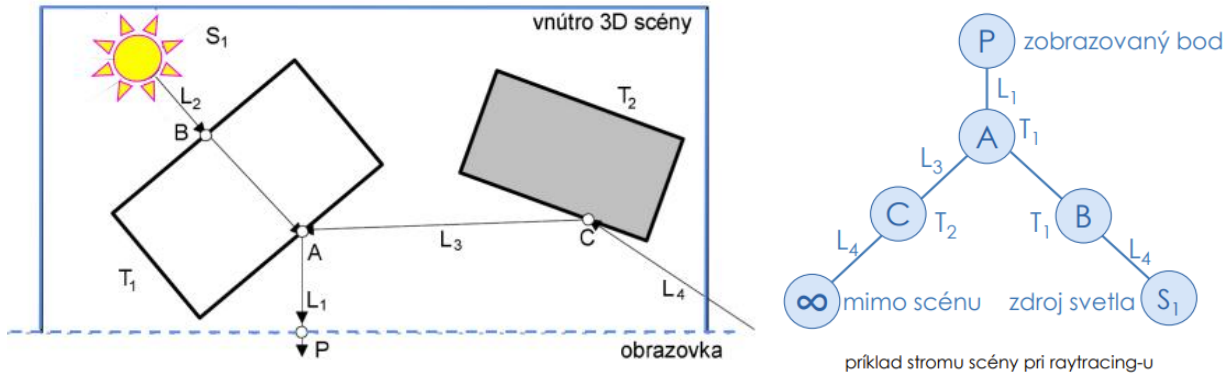
d) algoritmy pracujúce v objektovom priestore

- pre každý voxel sa hľadajú pixely V-Buffer Splatting

e) algoritmy pracujúce na hybridnom princípe

- nebude na skúške Shear-Warp (posuň a pokriv) – ekvivalent plávajúceho horizontu

72. Charakterizujte a popíšte metódu raytracing.



$$I_V = I_a + I_d + I_s + I_r + I_L$$

kde:

- I_V je výsledná intenzita
- I_a je zložka okolitého svetla
- I_d je difúzna zložka
- I_s je zrkadlová zložka (týka sa zdroja svetla)
- I_r je zložka z odrazeného lúča (týka sa telesa)
- I_L je zložka z lomeného lúča

- konštrukcia: koreň -> listy

- spätné sledovanie lúča: listy -> koreň

príklad výpočtu intenzity pri raytracing-u

73. Uveďte a v krátkosti popíšte fraktály a časticové systémy.

Fraktály

- lat. frangere – zlomiť (to break)
- množina, ktorej Hausdorffova dimenzia je väčšia než dimenzia topologická
 - neceločíselná, nadradená topologicky
 - hovorí o miere členitosti
- typy fraktálov:
 - L-Systémy
 - IFS – Iterated Function Systems
 - Dynamické množiny/systémy (dynamic sets)
 - vizuálne nenie niektorých funkcií komplexných čísel, ale v špecifickom intervale
 - len v tomto intervale dôjde k vizuálnemu efektu

1. L- systémy

- najjednoduchšie fraktály
- popis objektov prírody (napr. rastliny) – 2D aj 3D
- aplikuje pravidlá nezávislé na poradí
- definované stavom a tabuľkou akcií

- stav – poloha a orientácie L systému
- grafická interpretácia – korytnačia grafika

2. IFS

- generatívny typ fraktálov (systém iterovaných funkcií)
- napr. fraktálna komprimácia údajov
- na obrazovke sa iteratívne kreslia body
- napr. : Pascalov trojuholník, Sierpinského trojuholník, Sierpinského koberček
Kochova vložka, Dračia krivka, C krivka, Mengeleho huba

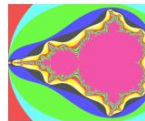
- Sierpinského koberček



- Kochovej krivka

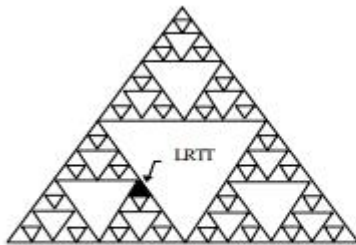


- Mandelbrotova množina

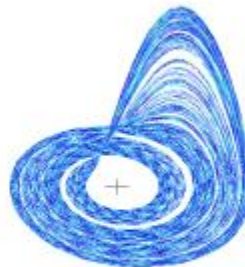


3. Dynamické množiny:

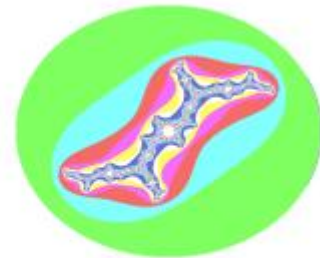
- rovinné obrázky 3D objektov
- 3. rozmer – farba – výška v danom bode
- komplexné čísla
- Juliova množina – je sebestodobná – pozostáva z kópií samej seba pomocou nelineárnych transformácií, ohraničená kružnicou
- Mandelbrotova množina
- plazma – simulácia oblakov, ohňa



1.



2.



3.

Časticové systémy (particle systems)

- Častica je objekt v scéne, ktorý má svoj vlastný životný cyklus. Niekde sa „narodí“, počas celého svojho života sa vyvíja podľa prideleného kódu a na konci „umiera“. To, čo vykonáva alebo ako sa vyvíja častica je plne v rúžii programátora. Časticové systémy (particle systems) sa pre niektoré prírodné javy používa častejšie ako fraktály.
- použitie: simulácia správania častíc (mraky, dym, oheň, ...)
- využívajú efekt použitia veľkého množstva malých časticových objektov
- 2 základné typy správania častíc:
 - usporiadané (pohyb, rotácia, zmena farby)
 - chaotické (výbuch, chemická reakcia, vietor)