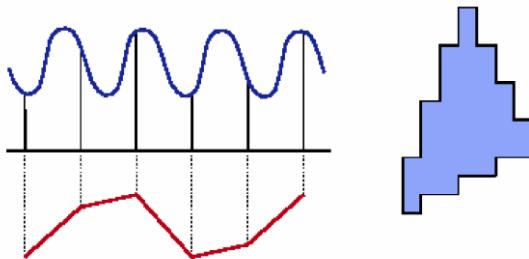


I. Rekonstrukce a vzorkování

1. Vysvětlete pojem (jev) „alias“. V jakých situacích vzniká, co je jeho příčinou a důsledkem? Uveďte metody používané pro antialiasing včetně vysvětlení základního principu?

Alias = rušivý jev vzniklý zobrazením v pravidelné diskrétní mřížce
= přidaná, nízkofrekvenční informace, která vzniká jako důsledek procesu vzorkování a rekonstrukce



Prostorový alias – vykreslíme něco v jiném místě, než by to mělo být, je to např.:

- zubaté zobrazení šíkmých linií – viz trojúhelník
- při kreslení husté sítě čar vzniká tzv. „Moiré efekt“ – viz šachovnice
- **interference** rychle se měnícího obrazu s pixelovým rasterem
 - příklad: plot v perspektivní projekci
 - příliš jemná nebo příliš vzdálená pravidelná textura (šachovnice ve velké vzdálenosti)

Časový alias

- projevuje se zejména při animaci **pomalého pohybu**, např.:
 - **blikání** na obvodu pohybujících se objektů
 - v extrémním případě se celé malé objekty objevují a opět mizí
- **interference** cyklického pohybu se snímkovou frekvencí
 - otácející se kolo se zdánlivě zastaví nebo se pomalu otačí opačným směrem

Antialiasing

- **odstranění (odfiltrování) vysokých frekvencí**
 - většinou vzorkování s vyšší frekvencí (supersampling), následná filtrace (rozmařání) a převedení zpět na původní vzorkování
 - nebo odstranění vysokých frekvencí přímo ze spojitého obrazu
- **převedení aliasu na šum (stochastické vzorkování)**
 - vzorkování s vyšší frekvencí ale **ne pravidelně** (Poisonovo rozložení, roztažení, částečné roztažení, náhodné vzorkování, nezávislé roztažení)

2. Vysvětlete pojmy vzorkování a kvantování. Co je rekonstrukční filtr? Jak ovlivňuje zobrazení vzorkovaného signálu?

Vzorkování (sampling) = zaznamenávání hodnot (vzorků) v předem daných intervalech

Kvantování = „zaokrouhllování“, probíhá v oboru hodnot obrazové fce, ten rozdělíme na intervaly, jimž je pak přidělena jediná, zástupná hodnota (reálnou hodnotu nahrazuje zástupnou hodnotou)

rekonstrukční filtr je dán vlastnostmi výstupního zařízení

Jeho kvalita (stupen polynomu, ...) urcuje ako kvalitne dojde k rekonstrukcii povodneho signalu. Je dobré ho volit podla poznatkov o povodnom signale.

3. Rekonstrukce = přechod od diskrétní fce I_i , def. pro $i = 1, 2, \dots$ ke spojité fci $f(x), x \in \mathbb{R}$

Rekonstrukční filtr = nějaká spojitá fce (plocha vymezená fcí je jednotková), při rekonstrukci

vynásobíme hodnotu vzorků v bodech I_i rekonstrukčním filtrem, získané fce sečteme a máme spojitu fci

- rekonstrukční filtr by neměl mít příliš velkou změnu v gradientu v okolí vzorku – to může vést k nepříznivému rozmazání obrazu

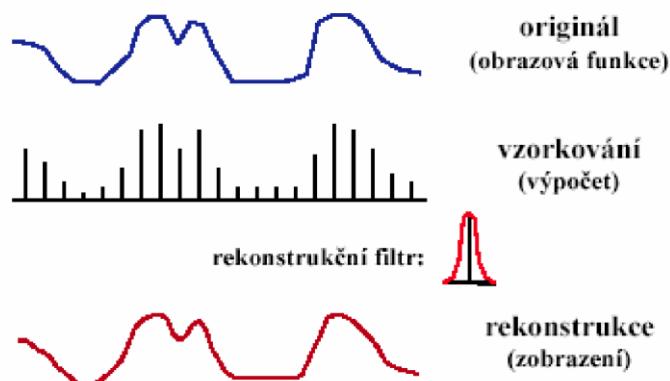
Vzorkování a rekonstrukce

- **vzorkování** nebo **výpočet** obrazové funkce
 - před vzorkováním by se z obrazu měly odstranit všechny vyšší (nezobrazitelné) frekvence filtr typu dolní propustnost (průměrování v okénku)
 - při **syntéze obrazu** se mohou vyšší frekvence zanedbat přímo (vyhlazování vzorkováním plochy)
- **rekonstrukční filtr** je dán vlastnostmi výstupního zařízení
 - např. na monitoru se stopy sousedních pixelů překrývají

Podmínky rekonstrukce

Signál musí být frekvenčně omezen.

Signál může být rekonstruován ze vzorků, pokud je vzorkovací frekvence f_s větší než dvojnásobek nejvyšší frekvence f_{\max} přítomné v signálu (Nyquistova rychlosť).



4. Co je spektrum neperiodické funkce? Jak ovlivňuje pravidelné vzorkování spektrum výsledné, tj. vzorkované funkce? Jaké jsou podmínky pro úplnou rekonstrukci vzorkovaného signálu?

Spektrum neperiodické fce :

- pokud fce není perodická, spektrum fce je spojité
- Cez Fourierova transformace

$G(f)$ - frekvenční obsah, spektrum funkce $g(x)$:

$$G(f) = \int_{-\infty}^{\infty} g(x) e^{-i2\pi fx} dx \Leftrightarrow g(x) = \int_{-\infty}^{\infty} G(f) e^{+i2\pi fx} df$$

Vzorkování

před vzorkováním by se z obrázku měly odstranit všechny vyšší (nezobrazitelné) frekvence filtr typu dolní propust (průměrování v okénku)

při syntéze obrazu se mohou vyšší frekvence zanedbávat přímo (vyhlazování vzorkováním plochy)
(Vyšsие frekvencie nez vzorkovacia sa stracaju)

Podmínky rekonstrukce

$$G_s(f) = G(f) + G_{high}(f)$$

Signál musí být frekvenčně omezen.

Nyquistův vzorkovací teorém:

Signál může být rekonstruován ze vzorků, pokud je vzorkovací frekvence f_s větší než dvojnásobek nejvyšší frekvence f_{\max} přítomné v signálu (Nyquistova rychlosť, N. limita).

5. Konvoluce dvou funkcí, konvoluční teorém.

Konvoluci dvou fcí $I(x)$ a $h(x)$ označujeme operátorem * je definována jako:

$$I(x) \star h(x) = \int_{-\infty}^{\infty} I(x-\alpha)h(\alpha) d\alpha$$

kde $h(x)$ je konvoluční jádro.

Konvoluční teorém

Buď $I(t)$, $h(t)$ integrovatelné fce, a jejich Fourierovy transformace $F(\omega)$ a $H(\omega)$, platí:

$$\begin{aligned} g(t) &= \{f * h\}(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau \\ &\quad \downarrow \\ G(\omega) &= \sqrt{2\pi} F(\omega)H(\omega) \end{aligned}$$

... platí i naopak ...

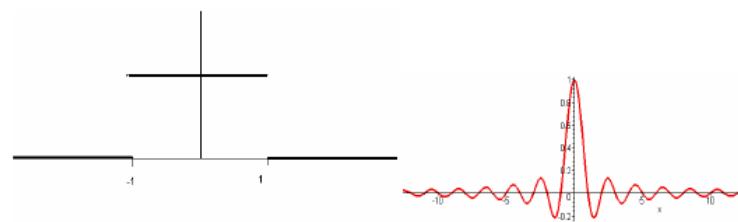
$$\begin{aligned} F(\omega) \star H(\omega) &\Leftrightarrow f(t)h(t) \\ F(\omega)H(\omega) &\Leftrightarrow f(t) \star h(t) \end{aligned}$$

tedy (konvoluce dvou obrazů = součin vzorů) a naopak (konvoluce dvou vzorů = součin obrazů)

6. Jaký tvar má ideální nízkopásmový filtr ve frekvenční doméně? K čemu se používá? Jaký konvoluční filtr je jeho obrazem? Nakreslete a vysvětlete na příkladu vzorkovaného signálu s omezeným frekvenčním rozsahem.

MPG:

Ideální filtr vysokých frekvencí má ve Fourierově prostoru tvar skokové funkce se středem v počátku (-1..1->1, jinak 0). Jemu v obrazové doméně odpovídá funkce sinc(x). Protože je funkce sinc(x) neomezená, musí se v praxi approximovat.



skoková funkce(Fourierův prostor) sinc(x) (obrazová doména)

Ideální rekonstrukce

$$H(f) \leftrightarrow \text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

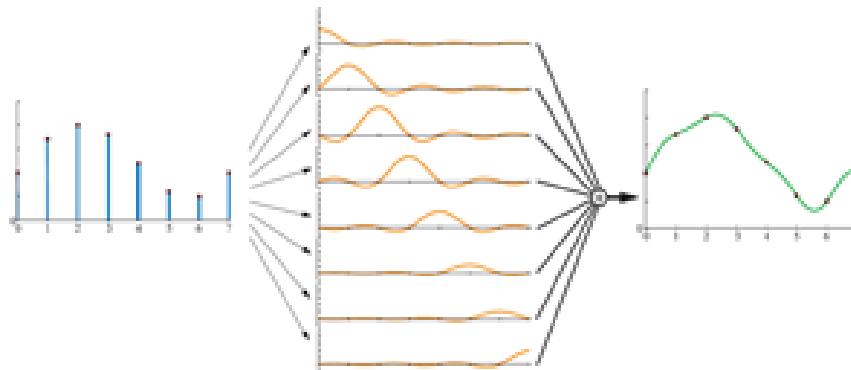
$$A/2W, 2W \leftrightarrow A \frac{\sin(\pi x/W)}{\pi x/W}$$

Rekonstrukce (podle konvol. teorému):

$$g(x) = \text{sinc}(x) * g_s(x)$$

$$= \int_{-\infty}^{\infty} \text{sinc}(\lambda) g_s(x - \lambda) d\lambda$$

Obrázek



7. Vysvětlete principy a rozdíly zpracování signálu před-filtrováním, post-filtrováním, super-vzorkováním a stochastickým vzorkováním.

Antialiasing – přístupy

- **prefiltering** – nejjednodušší a zároveň nejméně praktické řešení, pomocí nízkofrekvenčního filtru odstraníme z obrazu informaci, kterou nelze vzorkovat (redukujeme šířku pásma)
- **postfiltering ~ supersampling** - vzorkování, kdy se spočítá n x n roztažených, pravidelných nebo náhodných vzorků a pak se to zprůměruje
 - k průměrování použijeme filtr pro stanovení váhy každého vzorku, sečteme všechny vážené vzorky a vypočítáme barvu pixelu
- příklad filtru: $\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$
- **stochastické vzorkování** – také můžeme mít více vzorků, ale mezi jednotlivými vzorky NENÍ konstantní vzdálenost, jejich pozice je náhodná (Poissonovo rozložení, roztažení, ...)

Poznámka: často se používá kombinace 2 a 3, tedy mám více vzorků, ale náhodně rozmístěných, vyhnu se tak pravidelným chybám, na které je lidské oko citlivé

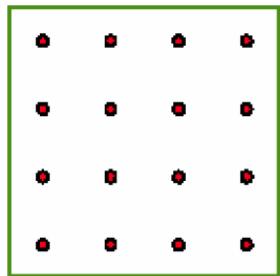
8. Jaké vlastnosti by měl mít dobrý vzorkovací algoritmus? Vysvětlete pomocí příkladů vzorkovací algoritmy, jejich principy a vlastnosti.
(random, jitter, semijitter, N-rooks, Poisson)

Vlastnosti vzorkovacího algoritmu:
- rovnoměrné pokrytí dané oblasti

- absolutní pravidelnost je nežádoucí (interference)
- efektivní výpočet

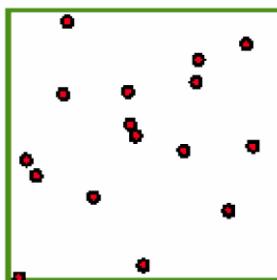
Vzorkovací algoritmy:

- Pravidelné vzorkování
 - o Neodstraňuje rušivé interference (pouze je přesune do vyšších frekvencí)



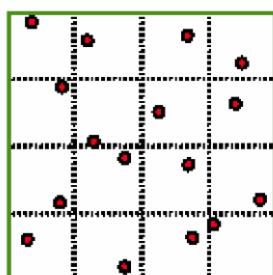
- Náhodné vzorkování

- o N nezávislých náhodných pokusů s rovnoměrným rozložením pravděpodobnosti
- o Vzorky mohou vytvářet **větší shluky**
- o Velký podíl **šumu** ve výsledku

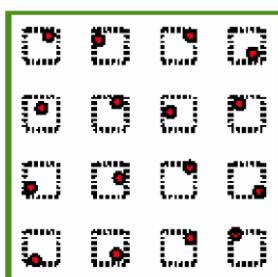


- „roztřesení“ („jittering“)

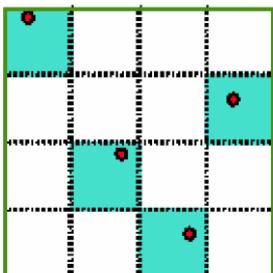
- o $K \times K$ nezávislých náhodných pokusů v $K \times K$ shodných subintervalech (pokrývajících původní interval beze zbytku)
- o Omezení pravděpodobnosti **velkých shluků**
- o **Rovnoměrnější pokrytí** vzorkovaného intervalu ..



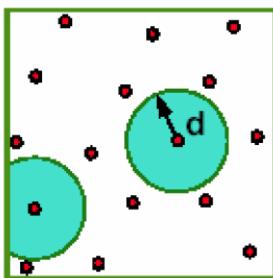
- Částečné „roztřesení“ („semijittering“)



- N věží (nezávislé „roztřesení“)



- Poissonovo diskové vzorkování



9. Praktické vyhlazovací metody při vykreslování grafických primitiv v rastru.

Metody vyhlazování

1. úprava signálu – **prefiltering**

redukce šířky pásmo signálu pomocí nízkofrekvenčního filtru
nejkvalitnější, ale často nepraktické řešení

2. úprava vzorků pomocí **supersampling**

pomocí více vzorků zvýší Nyquistovu frekvenci
jednoduchá, často používaná metoda

3. úprava vzorků pomocí **stochastického vzorkování**

relativně jednoduchá metoda, používá se v kombinaci s 2.

Vzorkovací metody

- **předpis:** $k \dots [x_k, y_k]$

- výběr vzorku z dané oblasti:
nejčastěji tvaru obdélníka, čtverce nebo kruhu v 2D
- vzorkování ve vyšších dimenzích (řádově do dim=10)

- požadované **vlastnosti** vzorkovacího algoritmu

- rovnoměrné pokrytí dané oblasti
- absolutní pravidelnost je nežádoucí (interference)
- efektivní výpočet

10. Kanál alfa, model pokrytí pixelu, alfa-míchání obrazů a antialiasing.

KANÁL ALFA

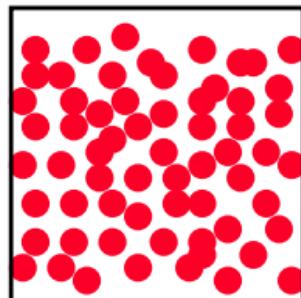
- procentuální pokrytí pixelu barvou
- doplněk průhlednosti:
 - pro $\alpha = 0$ je pixel zcela průhledný (nemá vliv na výsledek)
 - pro $\alpha = 1$ je pixel neprůhledný ("nic za ním neprosvitá")
- ukládání hodnoty α v každém pixelu
 - často se používá celočíselná reprezentace (0-255)

- čtverice [R,G,B, α $\alpha\alpha$ α]

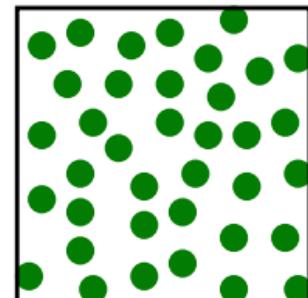
MODEL POKRYTÍ PIXELU

pixel $[A, \alpha_A]$ je náhodně pokryt barvou A
s rovnoměrně rozloženou pravděpodobností α_A
- skládání geometricky nezávislých tvarů
- vyhovuje ve většině případů

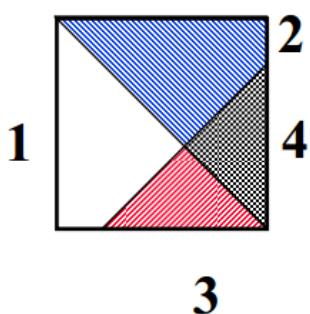
$$\alpha=0.5$$



$$\alpha=0.2$$



	oblast	plocha	vybarvení
1	nic	$(1-\alpha_A)(1-\alpha_B)$	0
2	A	$\alpha_A(1-\alpha_B)$	0, A
3	B	$\alpha_B(1-\alpha_A)$	0, B
4	A i B	$\alpha_A\alpha_B$	0, A, B



celkem 12 možností

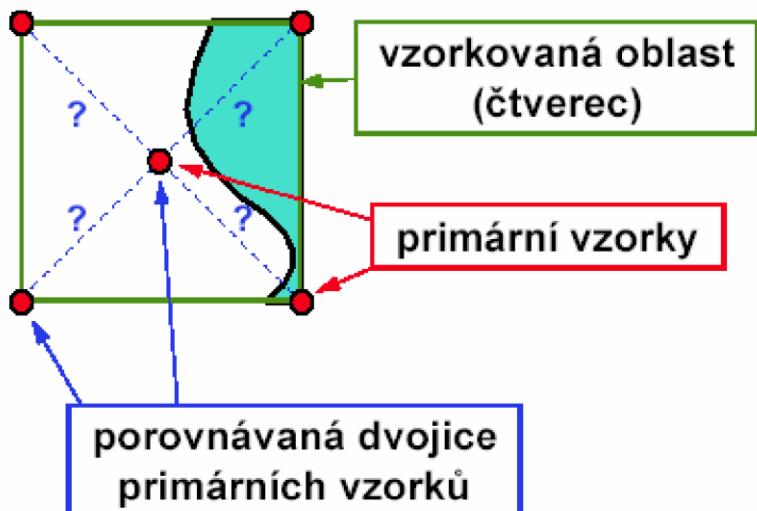
11. Vzorkování s adaptivním zjemňováním. Kritéria zjemňování, rekonstrukce výsledku.

Adaptivní zjemňování

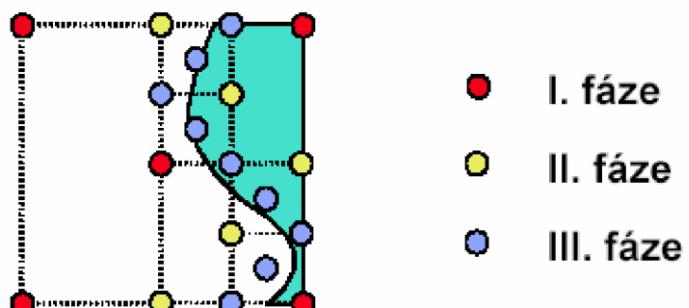
- vzorkování podle **lokální důležitosti** (vážené vzorkování) nebo **zajímavosti**
 - o některé oblasti pokryvaného intervalu mají větší váhu
 - o oblasti s větší variací vzorkované funkce je nutno pokrýt hustěji
- „důležitost“ nebo „zajímavost“ **nemusí znát dopředu**
 - o algoritmus se musí přizpůsobovat dosaženým výsledkům (adaptabilita)

Zjemňovací kritéria

- funkční hodnoty (rozdíl, rozptyl, gradient)
 - o rozdíl v barvě sousedních vzorků, ...
- čísla zobrazených těles
 - o větší priorita
 - o textury s opakujícími se vzory: signatury
- stromy výpočtu (rekurzivní sledování paprsku)
 - o topologické porovnávání celých stromů nebo jen několika horních pater
 - o identifikátor stromu – rekurzivní konstrukce pomocí našívací funkce

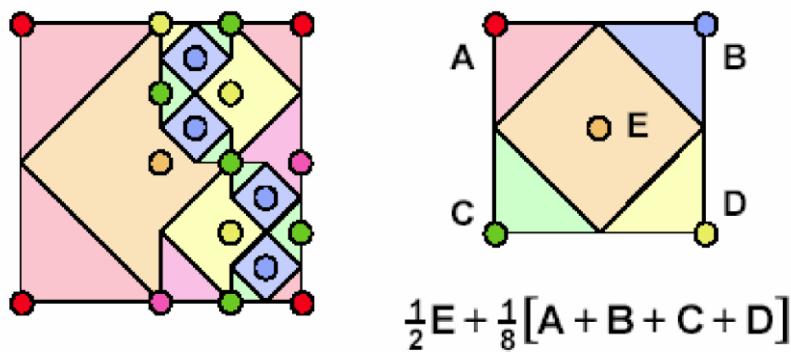


Rekurzivní zjemňování (Whitted)
Výsledný soubor vzorků



Celkem $5+5+9 = \mathbf{19}$ vzorků
(z celkového počtu **41**)

Rekonstrukce výsledku



V každém již dále neděleném čtverci se plocha rozdělí na dva protější vzorky

II. Textury

12. Co je textura? Uveďte jednotlivé typy používaných textur. Co ovlivňuje textura v lokálním světelném modelu a lokální geometrii tělesa? Uveďte možnosti aplikace textury ve světelném modelu.

Já: Textury

- 2D textury
- 3D textury
- Bump textury (hrbolaté textury)

Co mění textura

- Nejčastější:
 - Difúzní barva

$$I_{diff} = I_a \rho_a + I_{light} (\rho_d \cos \theta + \rho_s \cos^n \alpha)$$

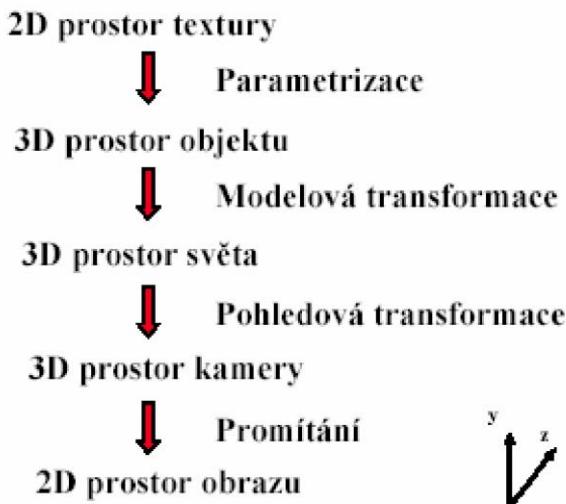
- A tak ... zrcadlové efekty se nemění

Ale ...

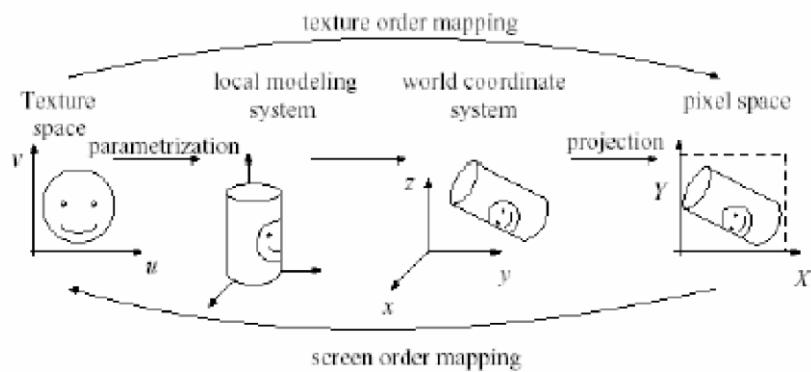
- Nejen difúzní barva, ale i další parametry se dají měnit jinými texturami
 - spekulární barva Phongova odrazu
 - směry normál (bump-mapping) nahrazení složité (mikro) geometrie
 - Příchozí světlo (mapy okolí, nebo reflexní mapy)
 - Náhodné textury (užití syntézy šumu)
 - Fraktální textury (deterministické i stochastické)

13. Uveďte sled transformací, které se uplatní při nanášení 2D textury na povrch tělesa. Vysvětlete přímé a inverzní nanášení textur a s tím spojené problémy.

Mapování textur \Leftrightarrow warp obrazu



Přímé na inverzní nanášení textur



Přístupy nanášení textur

1) prohlížení obrazového prostoru (inverzní mapování)

```
for all (x,y)
(u,v)=f-1(x,y)
SCREEN[x][y] = TEXTURE[u][v]
```

2) prohlížení prostoru textury (přímé mapování)

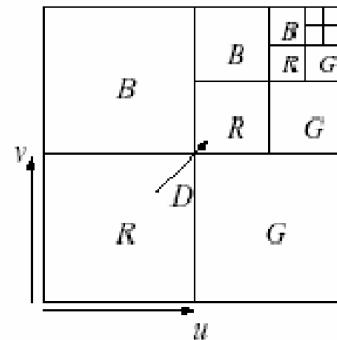
```
for all (u,v)
(x,y) = f(u,v)
SCREEN[x][y] = TEXTURE[u][v]
```

14. Filtrování textur pomocí MIP-MAP a součtové tabulky. Princip a použití při vykreslování.

“multum in parvo”

MIP[1..M,1..M]

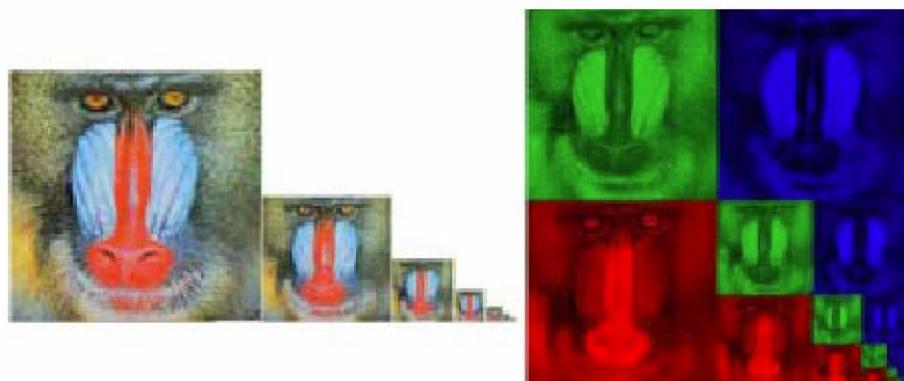
$$1 + 2^{-2} + 2^{-4} + \dots \approx 1.33$$



$$R(u, v, D) = MIP[(1 - 2^{-D}) \cdot M + u \cdot 2^{-D}, (1 - 2^{-D}) \cdot M + v \cdot 2^{-D}]$$

$$G(u, v, D) = MIP[(1 - 2^{-(D+1)}) \cdot M + u \cdot 2^{-D}, (1 - 2^{-D}) \cdot M + v \cdot 2^{-D}]$$

$$B(u, v, D) = MIP[(1 - 2^{-D}) \cdot M + u \cdot 2^{-D}, (1 - 2^{-(D+1)}) \cdot M + v \cdot 2^{-D}]$$



Aplikace MIP-MAP

Volba \mathbf{D} podle rozsahu d textury umístěné do plochy pixelu:

Aproximace d:

$$d = \max \left\{ \begin{array}{l} \| [u(x+1), v(x+1)] - [u(x), v(x)] \|, \\ \| [u(y+1), v(y+1)] - [u(y), v(y)] \| \end{array} \right\}$$

$$\approx \max \left\{ \sqrt{\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2}, \sqrt{\left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2} \right\}$$

Požadovaná úroveň \mathbf{D} v pyramidě

$$D = \log_2(\max\{d, 1\})$$

Aplikace MIP-MAP

Minimum 1 vzplývá ze situace, kdy pixel je pokryt pouze částí texelu. Pak není nutné filtrovat.

$$D = \log_2(\max\{d, 1\})$$

Jednoduché použití Trunc/Round pro výpočet indexu na základě \mathbf{D} by vedl k nespojitostem při zobrazení.
Použijeme interpolaci

$$R_{\text{filtered}} = R(u, v, \text{Trunc}(D)).(1 - \text{Fract}(D)) + R(u, v, \text{Trunc}(D) + 1).\text{Fract}(D)$$

$$G_{\text{filtered}} = \dots, \quad B_{\text{filtered}} = \dots$$

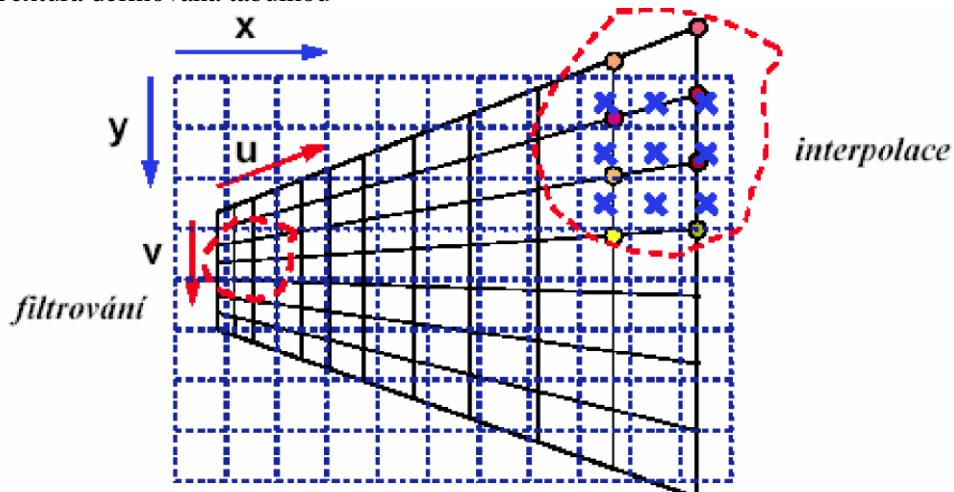
MPG: MIP – ultum in parvo = mnohé v malém

Reprezentace obrazu metodou mip-mapping se používá při mapování čtvercových textur. Jeho výhodou je, že v jedné reprezentaci je uchováváno více obrazů v různém rozlišení, což zvyšuje především rychlosť mapování takto reprezentované textury.

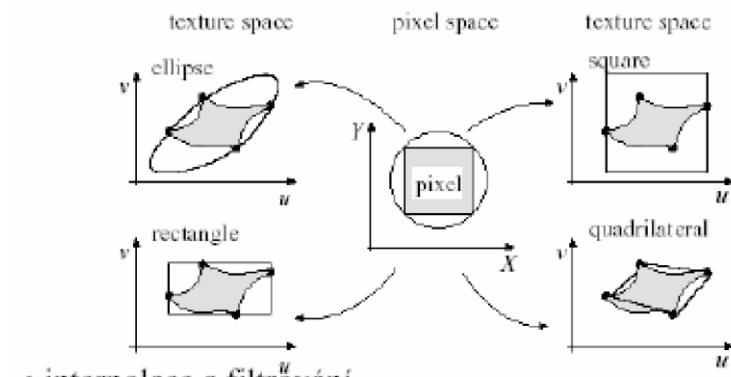
Já: Používá se pro různé úrovně kvality zobrazení textury (vzdálenost texturovaného tělesa od pozorovatele)

15. Vysvětlete postup při nanášení textury definované vzorky (tabulka, obraz) a související problémy. Bilineární a bikubická interpolace textury. Filtrování textur při kontrakci.

Textura definovaná tabulkou



Předběžné filtrování textur



- interpolace a filtrování
- approximace pokryté plochy obdélníku a filtrování
- MIP-MAP
- Součtová tabulka
- EWA filtrování

Typy interpolace

- **bez interpolace** (zaokrouhlení)
 - o nejjednodušší a nejrychlejší metoda
 - o pokud se rozlišení obrázku blíží rozlišení textury vznikají výrazné a nepříjemné artefakty (Doom)
- **bilineární** interpolace
 - o zajišťuje **spojitost** obrazové funkce (C_0)
- **polynomální** interpolace (např. spline funkce)
 - o **spojitost** vyšších řádů (u bikubické až C_2)
 - o výpočetně náročná (kombinace 9-16 hodnot ve 2D)

Bilineární a bikubická interpolace

$$\begin{aligned}
 & \text{Bilineární interpolace:} \\
 & f(a, b) = a \cdot [b \cdot f_{11} + (1-b) \cdot f_{10}] + \\
 & \quad + (1-a) \cdot [b \cdot f_{01} + (1-b) \cdot f_{00}] \\
 & f(a, b) = \sum_{i,j=0}^3 C_i(a)C_j(b)f_{ij} \\
 & C_i(t) \dots \text{kubické polynomy}
 \end{aligned}$$

The diagrams illustrate the sampling of a bilinear grid and a bicubic grid.
 - The bilinear grid shows a 2x2 neighborhood of pixels with values f_{00} , f_{01} , f_{10} , and f_{11} . A point $f(a, b)$ is sampled at coordinates a and b using linear interpolation along the rows and columns.
 - The bicubic grid shows a 4x4 neighborhood of pixels with values f_{00} through f_{33} . A point $f(a, b)$ is sampled at coordinates a and b using cubic interpolation along the rows and columns.

$$f(a, b) = \sum_{i=0}^3 \sum_{j=0}^3 C_i(a) C_j(b) f_{ij}$$

B-spline
váhové funkce:

$$\sum_{i=0}^3 C_i(t) = 1$$

$C_0(t) = (1-t)^3$
$C_1(t) = 3t^3 - 6t^2 + 4$
$C_2(t) = -3t^3 + 3t^2 + 3t + 1$
$C_3(t) = t^3$

$$0 \leq C_i(t) \leq 1 \quad \text{pro} \quad 0 \leq t \leq 1$$

Kubická B-spline interpolace

16. Vysvětlete důvody a princip nanášení textur přes pomocné povrchy, varianty přenosu textury z pomocného povrchu na povrch tělesa. Reflexní mapy, varianty.

Já: Využívá se pro nanášení textury na tělesa se složitým povrchem.

Já: Textura se nanese na pomocný povrch, následně se přenese na povrch původního tělesa, k tomu jsou různé postupy (odražený paprsek (lesklý povrch), normála povrchu, centroid povrchu, normála pomocného povrchu)

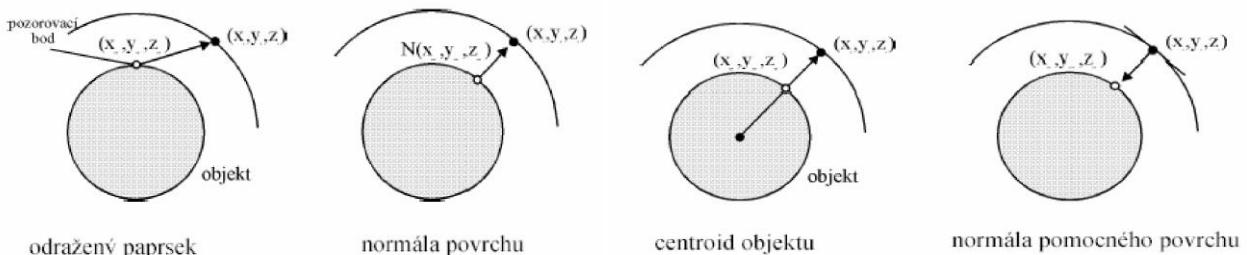
2D textury přes pomocné povrhy

1. 2D textury $T(u, v)$ nanesené na jednoduchý povrch T'
2. Z pomocného povrchu přeneseme texturu na těleso s povrchem O .

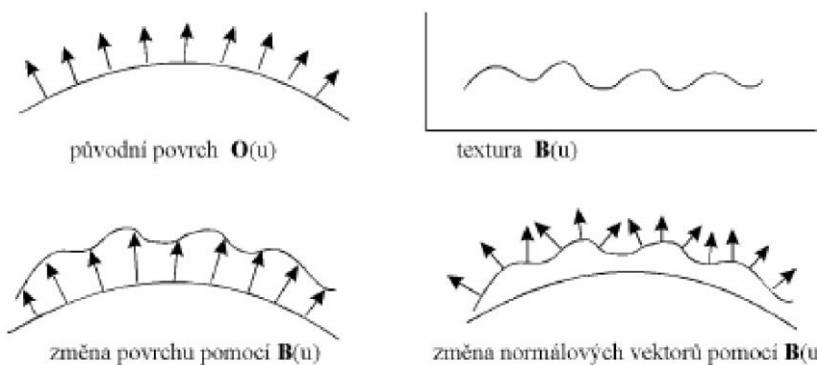
Jednoduché pomocné povrhy:

Rovnina, válcová plocha, krychle, koule

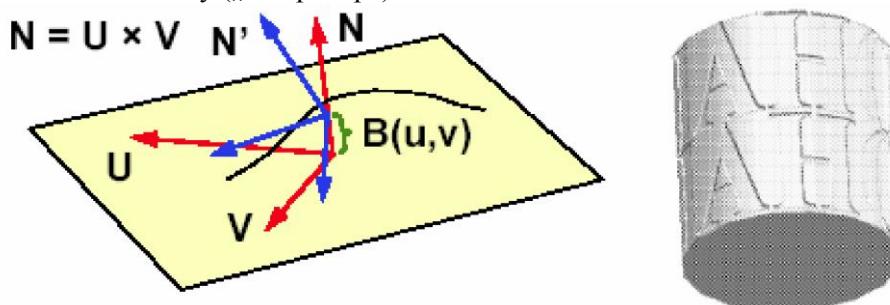
Reflexní mapování



17. Co jsou hrbovaté texture, jak jsou aplikovány? Určení modifikované normály pomocí diferencí.



Já: Místo zvrásnění povrchu změním jen normálové vektory tak aby odpovídaly zvrásněnému povrchu.
Modulace normály („bump map“)



$$P'(u,v) = P(u,v) + B(u,v) \cdot N / |N|$$

- napodobení **nerovností** na povrchu tělesa
- **B(u,v)** je funkce lokálního posunutí povrchu:
+ ven z tělesa, - dovnitř tělesa

Modulace normály

Původní normála: $N = U \times V$

$$\underline{P'(u,v)} = P(u,v) + \frac{B(u,v) \cdot N}{|N|}$$

Posunutý pod:

Aproximace **modifikované normály**:

$$N' = N + \frac{\frac{\partial B}{\partial u}(u,v) \cdot (N \times V) - \frac{\partial B}{\partial v}(u,v) \cdot (N \times U)}{|N|}$$

18. Syntéza textury z malé předlohy, postupy pro potlačení nežádoucích pravidelností.

Moznost generovat obrovské textury z malých vzorkov, pametova uspora.

Skokova mapa:

- nekonečný film realizovaný pomocí skoků mezi podobnými snímky
- kazky pixel si pameta k-skokov a nasledne sa z nich nahodne vybera
- moznost predpocitat podla podobnosti okolia pre kazdy pixel

Sešívané textury

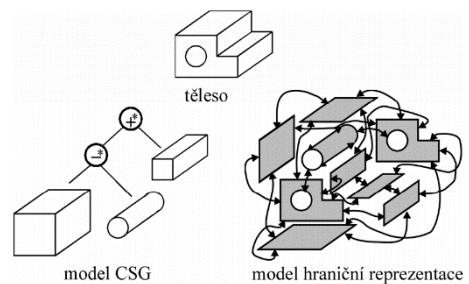
- Nahodny vyber blokov z povodnej textury
- Nahodny vyber blokov a ich nasledne plynule prekrytie
- Vyber vhodnych blokov a ich nasledne plynule prekrytie
- Vyber vhodnych blokov a ich spojenie pomocou rezu minimalizujuceho hranicnu chybu

III. Pokročilé modelování

19. Povrchové a objemové modely, datové struktury, použití.

Povrchove modely:

- Zlozene z trojuholnikov
 - Pole poloh jednotlivych bodov
 - Pole (po troch) indexov do pola bodov (trojuholníky)
 - Najcastejsie pouzivane
 - Velka vacsina ostatnich modelov sa prevadza pri vykresleni do tohto z dovodu HW podpory)
- Model hranicnej reprezentacie



Objemove modely

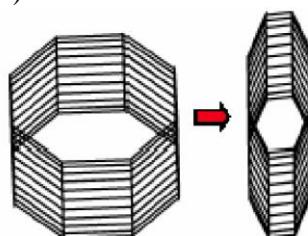
- CSG model (Constructive solid geometry) (AutoCad)
- Voxelovy model
 - K-dimenzionalna matica
 - Data v nej mozu byt bool, int, double,...
 - Pouzite pri zobrazeni 3d dat ziskanych z fyzickych merani ci simulacii,...

20. Co jsou Barrovy deformace, jaký je výsledek jejich aplikace?

Barrovy deformace $(X, Y, Z) = F(x, y, z)$

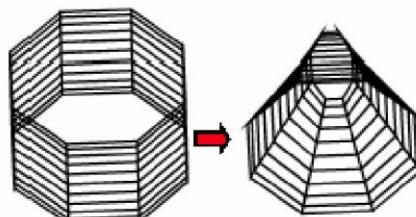
1. Změna měřítka

$$(X, Y, Z) = (s_x \cdot x, s_y \cdot y, s_z \cdot z)$$



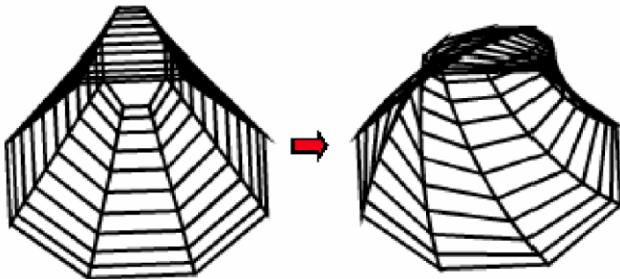
2. Zeslabování („tapering“)
ve směru osy z

$$(X, Y, Z) = (r_x \cdot x, r_y \cdot y, z)$$



3. Zkroucení („twisting“) - okolo osy z

$$(X, Y, Z) = (x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta, z)$$



21. Volné deformace těles (Sederberg, Parry). Vysvětlete princip a účinek deformace. Postup výpočtu. Jaké lokální souřadnice a parametry používá? Komentujte zajištění spojitosti v deformovaném tělese při použití více deformačních nástrojů.

Založené na deformaci Bezierova objemu

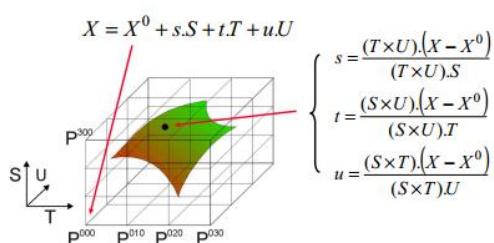
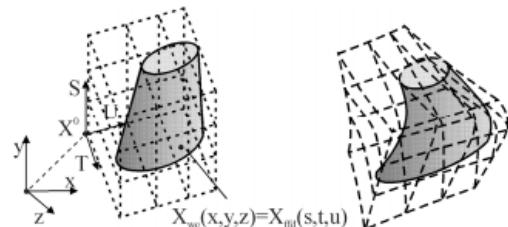
Aplikace: animace, modelování, zpracování obrazu

Postup:

1. Vytvoře deformační nástroje.
2. Vlož objekt do deformačního prostoru
3. Modifikuj deformační nástroje
4. Deformuj objekt podle změněného prostoru

Souřadnice:

Lokalne pre vramci kazdej FFD telesa.



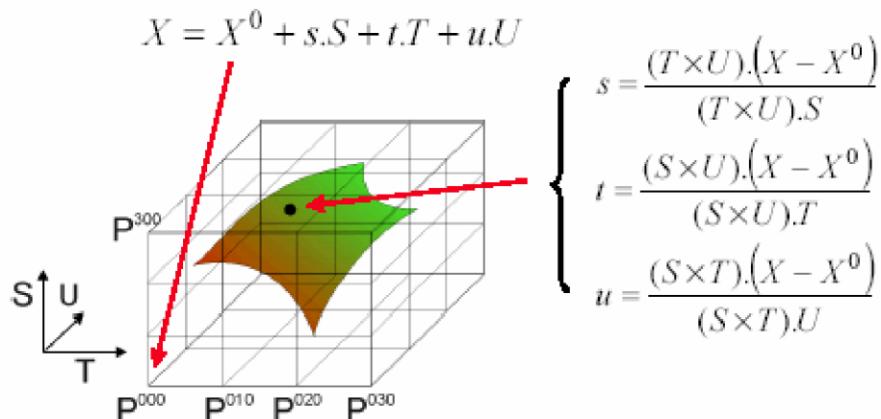
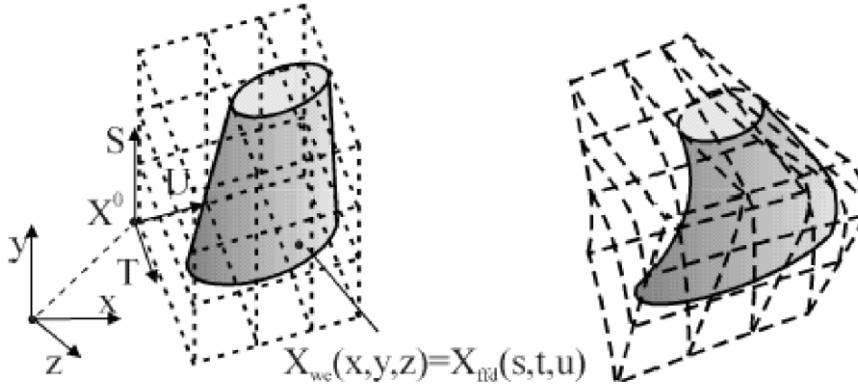
Spojitost

Derivacie ffd funkcií sa musia rovnati v mieste dotyku

22. Rozšíření FFD na obecné bodové svazy. Effd, princip výpočtu. Jaké lokální souřadnice a parametry používá?

Volné deformace (Sederberg&Parry)

SIGGRAPH: Dobrou fyzikální analogii pro FFD zvážení průzračného paralelního spojení, ohebná plastická hmota ve které je uložen objekt nebo množství objektů, které chceme deformovat. Objekty si představujeme také jako ohebné, takže se deformují podle plastického okolí.
„volné deformace“ – Free Form Deformation – FFD



body deformační sítě: $P^{ijk} = X_0 + \frac{i}{l}S + \frac{j}{m}T + \frac{k}{n}U$

$$X_{ffd} = \sum_{i=0}^l \binom{l}{i} (1-s)^{l-i} s^i \left[\sum_{j=0}^m \binom{m}{j} (1-t)^{m-j} t^j \left[\sum_{k=0}^n \binom{n}{k} (1-u)^{n-k} u^k P^{ijk} \right] \right]$$

Spojitost při skládání FFD

$$X_{ffd}(s, t, u) = X_{ffd}(s(v, w), t(v, w), u(v, w))$$

derivace složené funkce

$$\frac{\partial X_1(v, w)}{\partial v} = \frac{\partial X_1}{\partial s} \cdot \frac{\partial s}{\partial v} + \frac{\partial X_1}{\partial t} \cdot \frac{\partial t}{\partial v} + \frac{\partial X_1}{\partial u} \cdot \frac{\partial u}{\partial v}$$

$$\frac{\partial X_1(v, w)}{\partial w} = \frac{\partial X_1}{\partial s} \cdot \frac{\partial s}{\partial w} + \frac{\partial X_1}{\partial t} \cdot \frac{\partial t}{\partial w} + \frac{\partial X_1}{\partial u} \cdot \frac{\partial u}{\partial w}$$

nezávisí na deformaci

Postačující podmínky pro zachování C¹ spojitosti:
2 „sousední“ deformace X₁ a X₂ (pro s=0)

$$\begin{aligned} \frac{\partial X_1(0, t, u)}{\partial s} &= \frac{\partial X_2(0, t, u)}{\partial s} \\ \frac{\partial X_1(0, t, u)}{\partial t} &= \frac{\partial X_2(0, t, u)}{\partial t} \\ \frac{\partial X_1(0, t, u)}{\partial u} &= \frac{\partial X_2(0, t, u)}{\partial u} \end{aligned}$$

Změna objemu při FFD

SIGGRAPH: Další důvod že FFD je tak dobře aplikovatelná na solid modeling je že nám poskytne kontrolu nad změnami objemu které těleso prodélává pod FFD. Změna objemu kterou FFD provede(imposes) nad odlišným elementem je dána Jakobiánem FFD. Zde je daná FFD a její Jakobián

$$FFD(x,y,z) = (F(x,y,z), G(x,y,z), H(x,y,z))$$

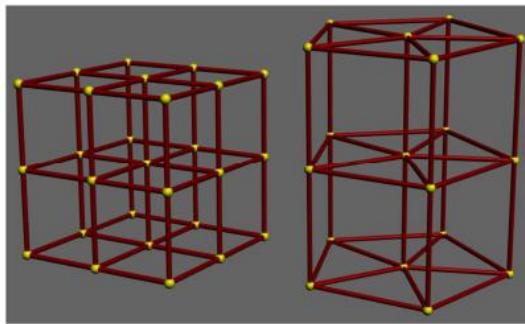
$$Jac(FFD) = \begin{vmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} & \frac{\partial F}{\partial z} \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \\ \frac{\partial H}{\partial x} & \frac{\partial H}{\partial y} & \frac{\partial H}{\partial z} \end{vmatrix}$$

Diferenciální objemový element

$$dx dy dz \quad \rightarrow \quad Jac(FFD(x,y,z)).dx dy dz$$

23. Deformace pomocí CC dělených objemů. Princip výpočtu.

Základem je bodový svaz – bod, hrana, stěna, jádro/buňka



CJD - postup

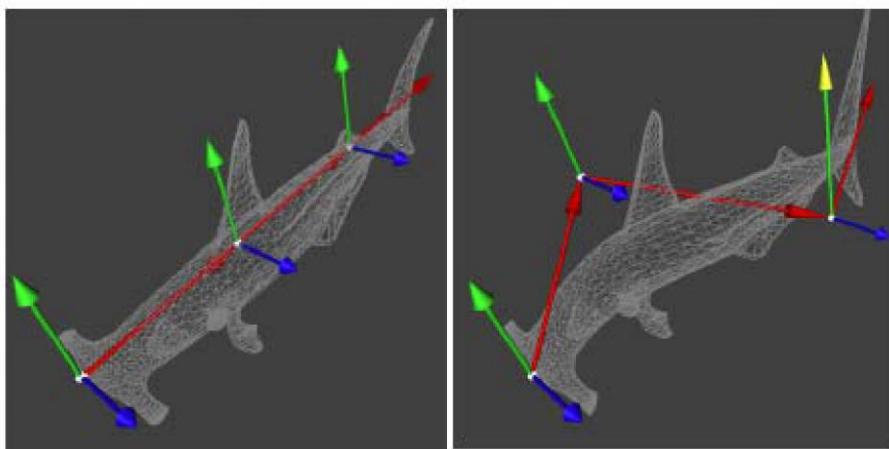
Vytvoř umísti bodový svaz.

Zjemni bodový svaz tak, aby vzniklo dostatečně „jemné“ dělení podprostoru na buňky

Pro všechny význačné body deformovaného povrchu

1. Nalezni buňku, ve které leží daný bod $P(x,y,z)$ a označbod indexem této buňky.
 2. S použitím numerické iterace urči lokální souřadnice u,v,w bodu $P(x,y,z)$ v dané buňce. Doporučený start iterace $(0.5,0.5,0.5)$.
 3. Výsledek uchovej u bodu P .
- S pomocí nalezených (u,v,w) vypočti polohu P_{cjd} v síti přemístěných řídících bodů.

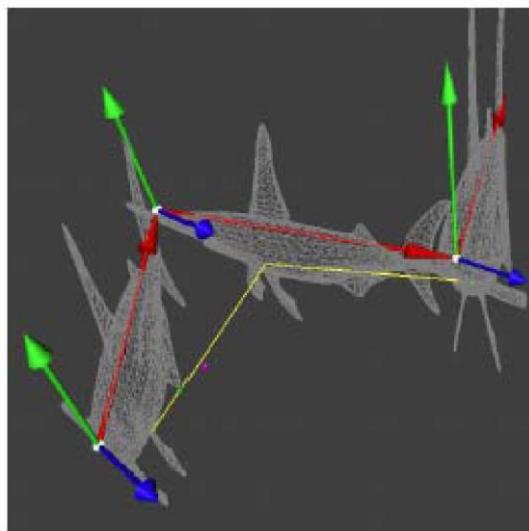
24.*Na jakém principu je založena deformace tělesa "deCasteljau"? Jaké lokální souřadnice a parametry používá? Jak se spočte poloha bodu z odpovídajících obrazů tohoto bodu v lokálních deformačních systémech?



$$\Phi[p, q] \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} q_x - p_x & s_x & t_x & p_x \\ q_y - p_y & s_y & t_y & p_y \\ q_z - p_z & s_z & t_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P_i^0(u) = P_i \quad 0 \leq i \leq n$$

$$P_i^j(u) = \Phi[P_i^{j-1}, P_{i+1}^{j-1}]u) \quad 1 \leq j \leq n, 0 \leq i \leq n-j$$



25. Co je deformace pomocí "drátů"? Jaké lokální souřadnice a parametry používá?

V skratce:

Je definovaný drát a jeho oblast vlivu, body na povrchu objektu ktoré sú v oblasti vlivu si definujeme v lokálnych súradničach drotu, potom drot zdeformujeme a nasledne body sú v oblasti vlivu zdeformujeme podľa krivky so silou určenou funkcou (vzdialenosť).

Transformacia aplikovana na bod je vypočítaná zo zmeny referenčnej kružnice na deformovaju.

Drát je definován ako 5-tice (W, R, s, r, f), kde

W – volnětvarovatelná parametrická křivka (deformační drát)

R – volnětvarovatelná parametrická křivka (referenční drát)

s – skalar, který je aplikován jako radiální měřítko kolem drátu

r – skalar, který definuje poloměr vlivu v okolí křivky

f – skalárna funkce hustoty, $f(0)=1$, $f(x)=0$ pro $x>=1$

Suradnice:

Parameter bodu na křivce, ktorý je najblízsí danemu bodu.

Sila vplyvu křivky na bod.

1. Uniformly scale P about point $R(p_R) \Rightarrow P_S$
$$P_S = P + (P - R(p_R)) \cdot (1 + (s-1) \cdot F(P, R))$$
2. Rotate P_S by modulated angle $\theta \cdot F(P, R)$
around axis $W'(p_R) \times R'(p_R)$, resulting point P_r
3. Add the translation $(W(p_R) - R(p_R)) \cdot F(P, R)$
$$P_{def} = P_r + (W(p_R) - R(p_R)) \cdot F(P, R)$$

26. Deformace pomocí tažených povrchů. Jaké lokální souřadnice a parametry používá?

- Modifikace tělesa použitím základních affinních transformací
- Přímé ovládání deformace uživatelem
- Užití metod v reálném čase
- Jednoduchá konstrukce

Výpočet

1. Generování taženého povrchu pokryvající množinu plošek (navázání vrcholů na povrch)
2. Zobrazení modelu a výběr neprázdné množiny plošek
3. Transformace libovolného přírezu
4. Aplikace transformací na vybrané plošky interpolované mezi přírezmy

Suradnice

- Relativne suradnice voci křivke tazeneho povrchu
- Vzdialenosť od křivky, parameter najblízsieho bodu na křivke, uhol otocenia.

27. Implicitní povrchy, distanční povrchy, příklady.

Implicitne povrhy

Reálná funkce $f(x,y,z)$

klasifikuje body v prostoru

- uvnitř $f > 0$
- vně $f < 0$
- na povrchu $f = 0$ zpracování a zobrazování objemových dat

implicitní povrhy vs. polygony

- hladší
- kompaktní reprezentace, méně primitiv
- obtížnější zobrazení v reálném čase

implicitní povrhy vs. parametrické pláty

- snazší spojování
- bez topologických problémů
- nižší stupeň
- obtížnější parametrizace
- snadnější sledování paprsku

Povrchové normály jsou obvykle gradientem funkce

Priklady:

Rovin, kvadriky, ...

Mozne definovat v deformovanom povrchu a potom ziskat spatnou transformaciou implicitny povrch v nedeformovanom priestore.

Distancne povrhy

Ako implicitna funkcia vzdialnosti od daneho povrchu

Priklady:

Bod->gula

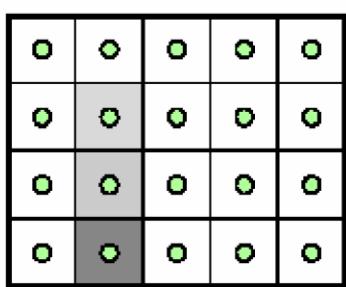
Priamka -> cylinder

Usecka -> sfylinder

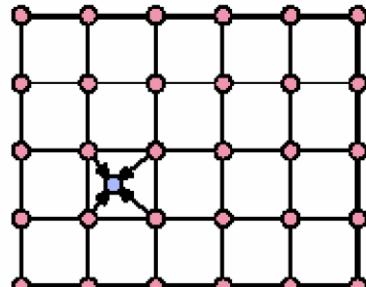
Obecna krivka -> sweep nurbs

IV. Zpracování a zobrazování objemových dat

28. Charakterizujte voxelová a buňková 3D data. Jakým způsobem zjistíte hodnotu v obecném prostorovém bodě objemových dat?



voxely
(naměřené hodnoty
jsou uprostřed)



buňky
(naměřené hodnoty
jsou ve vrcholech)

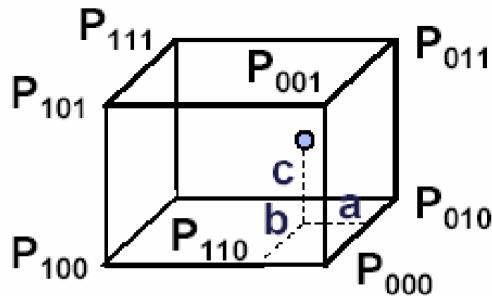
Interpolace v buňkách
- **Polynomiální interpolace a approximace**

- pro topologicky pravidelné mřížky
- **trilineární** interpolace
 - jednoduchý výpočet, není hladká
- **trikvadratická** nebo **trikubická** approximace

- hladké, ale vyžadují topologickou pravidelnost
- **radiální approximace**
- vhodná i pro topologicky nepravidelné mřížky

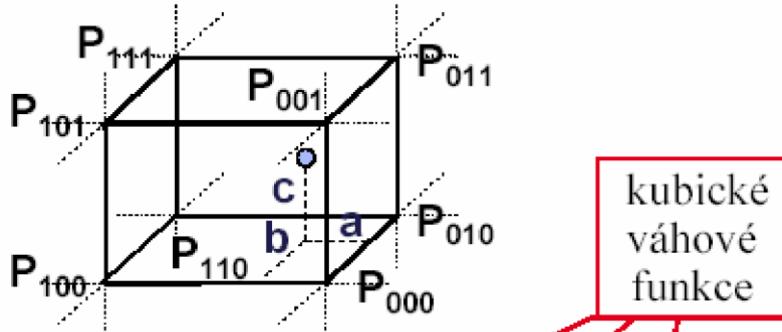
29. Vysvětlete a porovnejte trilineární, trikubickou a radiální approximaci. Pro které typy prostorových dat se používají?

Trilineární interpolace



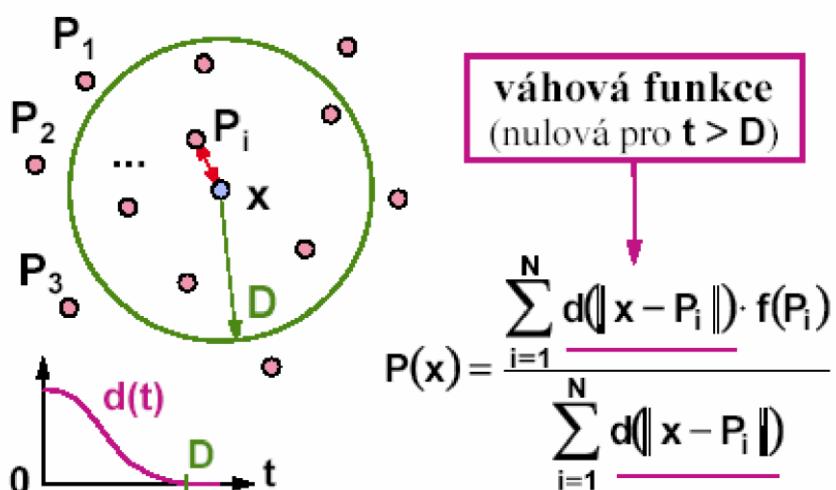
$$P(a, b, c) = (1-a)\left\{ \begin{array}{l} (1-b)[(1-c)P_{000} + cP_{001}] \\ + b[(1-c)P_{010} + cP_{011}] \end{array} \right\} + \\ + a\left\{ \begin{array}{l} (1-b)[(1-c)P_{100} + cP_{101}] \\ + b[(1-c)P_{110} + cP_{111}] \end{array} \right\}$$

Trikubická approximace



$$P(a, b, c) = \sum_{i,j,k=-1}^2 B_{i+1}(a)B_{j+1}(b)B_{k+1}(c) \cdot P_{ijk}$$

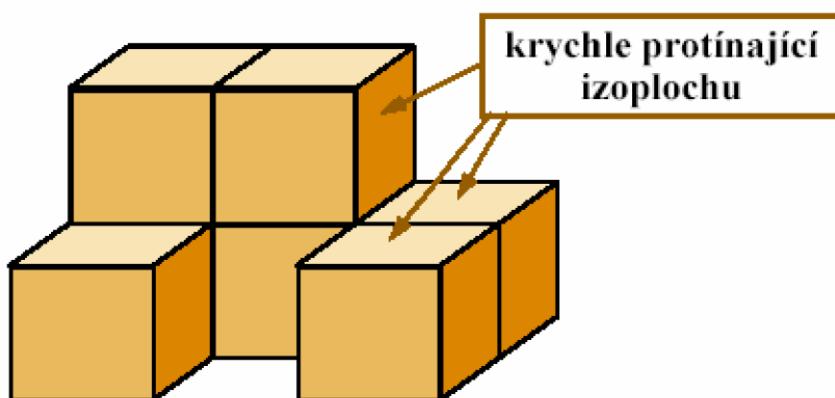
Radiální approximace



30. Vysvětlete zobrazování objemových dat pomocí tzv. neprůhledných, případně poloprůhledných kostek. Uveďte nedostatky a jejich možné odstranění.

Neprůhledné kostky

$$\min\{V(P_{ijk})\} \leq h < \max\{V(P_{ijk})\}$$



- zobrazení krychlí, které protínají **zadanou izoplochu**
 - o podle směru pohledu stačí kreslit pouze tři stěny krychle
- approximace je příliš **hrubá** (plocha je hranatá)
 - o lepší vzhled – **spojité stínování** gradientní metodou (hranaté okraje však zůstávají)
- současné zobrazení **několika izoploch**
 - o technika poloprůhledných ploch (kanál alfa)

31. Přímé vytvoření obrazu z objemových dat pomocí paprsku. Schémata průchodu paprsku scénou a stanovení zobrazované intenzity.

– simulace rozptylu světla v poloprůhledném (pasivním nebo aktivním) prostředí

– integrace podél dráhy paprsku

Schemata:

Prvy – zobrazi izoplochy, voxely ktore maju vacsiu hodnotu ako zadana

Priemer – zobrazi priemer po drahe paprsku

Max – pri MRA, zobrazi maximalnu hodnotu po ceste paprsku

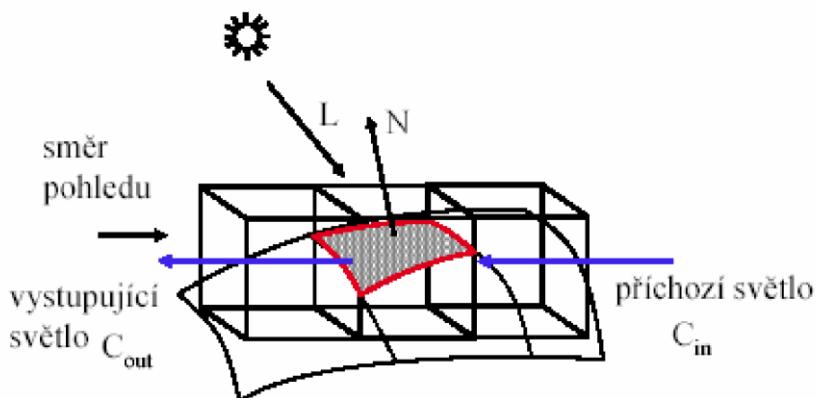
Akumulace – paprsek akumuluje farbu cestovanim po transparentnych voxeloch

Vypočet:

Integrováním po trase paprsku – súčet hodnot po drahe

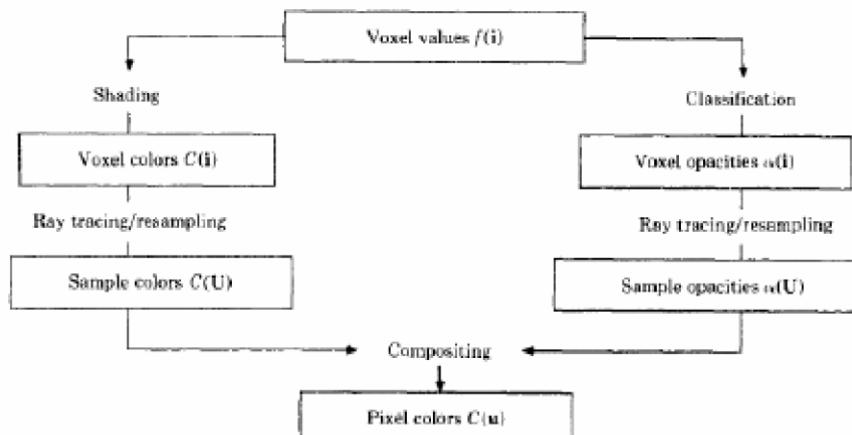
32. Vysvětlete metodu přímého zobrazování objemových dat (Levoy). Co je vstupem a výstupem algoritmu? Přípravná a sestavovací fáze.

Levoy – direkt volume visual.



Předzpracování vytvoří 2 pomocné objemy:

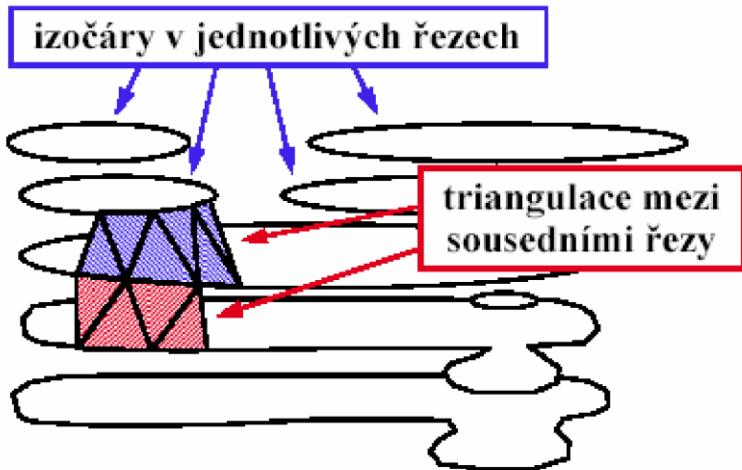
- pomocné pole barev $C(X)$
pro každý voxel vypočte barvu Phongovým modelem
odhad normály (gradientu) symetrickou diferencí
- pomocné pole neprůhlednosti $\alpha(X)$
klasifikace závislá na aplikaci



MPG: Algoritmus má dvě části v první (předspracování) se nastaví pole barev $C(X)$ a pole neprůhledností $\alpha(X)$.

Ve druhém kroku se vytváří výsledný dvojrozměrný obraz na průměrně kombinací obou pomocných objemů $C(X)$ a $\alpha(X)$ rovnoběžným promítáním. Pro každý pixel je do objemu vyslán jeden paprsek. Ten prochází pole $C(X)$ a $\alpha(X)$ a akumuluje hodnoty intenzity (jasu). Paprsek vstupuje do objemu voxelu s intenzitou (jasem) C_{in} je částečně utlumen průchodem jeho objemem o neprůhlednost α a získá část jeho barvy.

33. Jaké jsou hlavní kroky algoritmu pro konstrukci izoplochy pomocí izočar? (Ekoule et.al). Co je vstupem a výstupem algoritmu? Řešení konkávních napojení. Větvení 1:M.



Napojování izočar

- **výpočet izočar** v jednotlivých řezech

- izočára se reprezentuje jako lomená čára
- jeden řez může obsahovat několik uzavřených smyček izočar

- **triangulace izoplochy** mezi dvěma řezy

- trojúhelníky by neměly být příliš protáhlé
- **topologicky obtížné situace:** několikanásobné větvení (1:N, M:N), nejednoznačné přiřazení napojovaných izočar

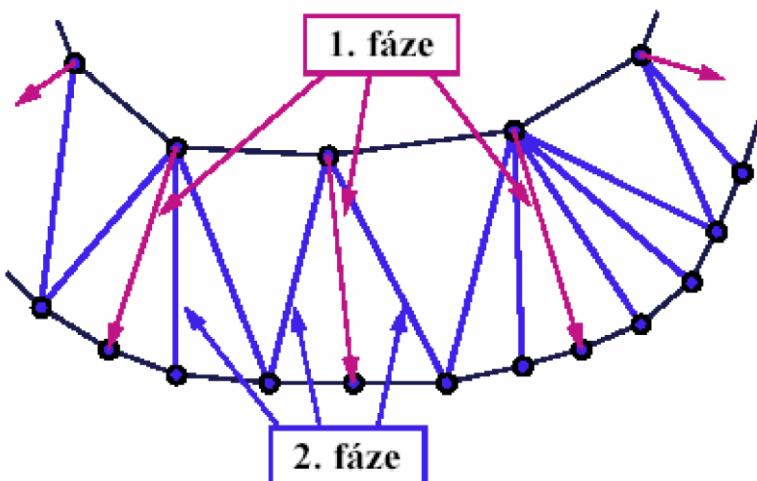
Triangulace

- rozdělení na několik geometricky/topologicky odlišných **případů**:

- konvexní napojení 1:1
- nekonvexní napojení 1:1
- větvení 1:N a M:N

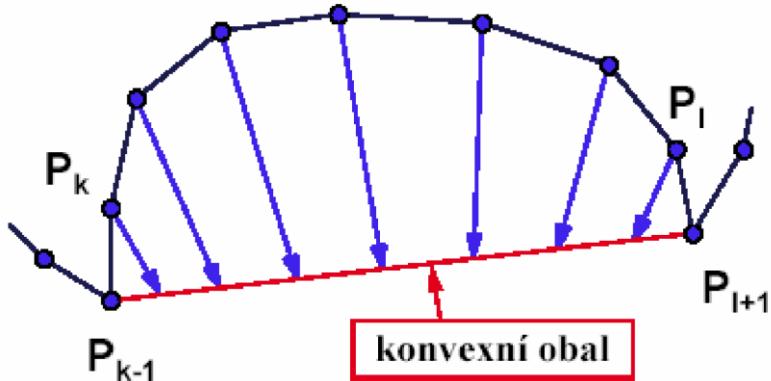
Konvexní napojení 1:1

- pro izočáru s **menším počtem vrcholů**: každý vrchol se spojí s nejbližším bodem naproti
 - efektivní algoritmus pracuje inkrementálně – hledá nejbližšího souseda v okolí naposledy přiřazeného vrcholu
- zbývající vrcholy z **druhé izočáry** se spojí s nejbližšími protějšky
 - některé vrcholy se musejí spojit se dvěma protějšími body (aby vznikla triangulace)

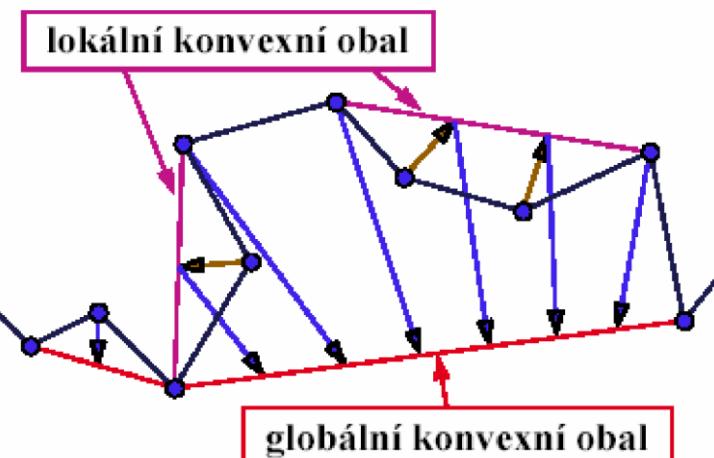


Transformace nekonvexních izočar

- přenášení vrcholů z nekonvexních úseků na **konvexní obal izočáry**
 - hypotéza: konvexní obaly dvou sousedních izočár si budou více podobné
 - přenášením se zachovává pořadí i relativní vzdálenosti vrcholů
- **triangulace** se provádí na konvexních obalech
 - hranami se potom spojí původní vrcholy izočar

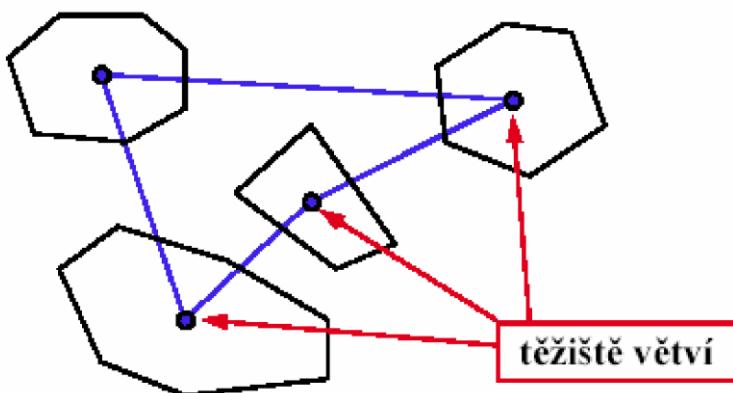


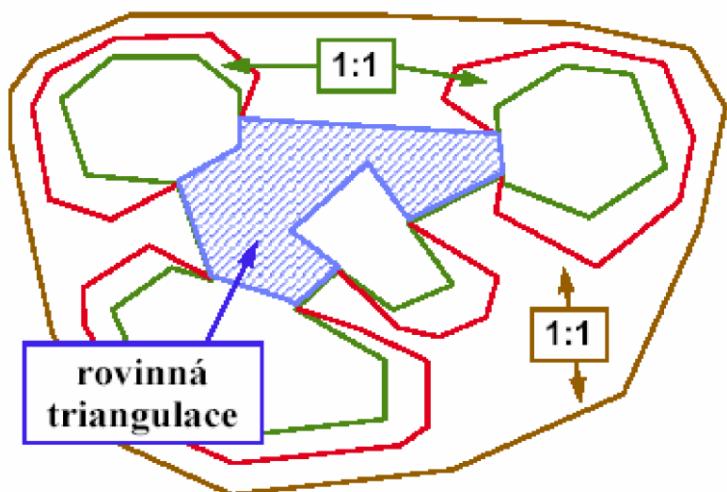
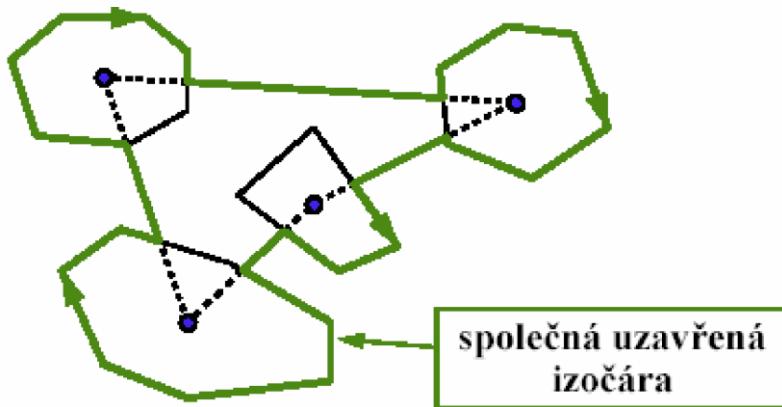
Složitější nekonvexní izočára



Větvení 1:N

- spočítám **pomocnou izočáru** v polovině mezi oběma řezy
 - konstrukce společné uzavřené izočáry pokrývající všechny spojované větve
 - interpoluje se pomocí prvního kroku algoritmu napojování 1:1
- N-krát aplikuji napojení 1:1 mezi každou větví a pomocnou izočárou
 - + doplním příp. rovinou triangulaci





34. Na jakém principu pracuje algoritmus „pochodující kostky“ (Lorensen)? Co je vstupem a výstupem algoritmu? Jak se spočítají souřadnice vrcholů trojúhelníků a normálové vektory? Nedostatky.

„Pochodující kostky“ (Lorensen)

- jeden z nejpoužívanějších algoritmů

- + jednoduchá implementace (i HW)
 - + generované trojúhelníky nemají dlouhé hrany
 - velké množství trojúhelníků (i menších než 1 pixel)
 - mohou se objevit drobné chyby v napojení
- generuje síť trojúhelníků v každé buňce

- topologie – podle konfigurace vrcholů buňky
- vrcholy sítě leží na hranách buňky

- snadno se implementuje gradientní stínování

Základní algoritmus

- 4 sousední řezy se načtou do paměti

- pro každou buňku mezi dvěma prostředními řezami se počítá šířka:

- podle konfigurace hodnot ve vrcholech buňky se z tabulky přečte topologie sítě
 - informace o vrcholech na hranách trojúhelníků

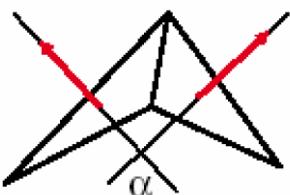
- vrcholy sítě se spočítají lineární interpolací podle hodnot ve vrcholech buňky

- **normálové vektory** ve vrcholech buňky se spočítají pomocí diferencí a lineárně se pak interpolují do vrcholů sítě
- **sít' trojúhelníku** se stínováním se zapíše **na výstup**
 - předpokládá se HW podpora viditelnosti a Gouraudova stínování

Implementace, vylepšení

- spočítané **vrcholy sítě** a **normálové vektory** se ukládají do „cache“ paměti
 - v sousedních buňkách se již nemusí počítat
- náhrada souvislé sítě trojúhelníků v buňce **jedním mnohoúhelníkem** (nerovinným)
 - průměr buňky je často malý (řádově několik pixelů)
 - větší efektivita při kreslení
- **množinové operace**, řezy rovinou
 - paralelně se provádí výpočet několika izoploch

35. Zjednodušení trojúhelníkové sítě. Princip algoritmu s metrikou QEM.



Zjednodušení sítě

Významné hrany – test prostorového úhlu podle prahové hodnoty

Vrchol s právě 2 významnými hranami – **vrchol vnitřní hrany**

Kritérium vzdálenosti od hrany

Vrchol s jednou nebo více než 2 významnými hranami – **roh**

Ponecháváme beze změny

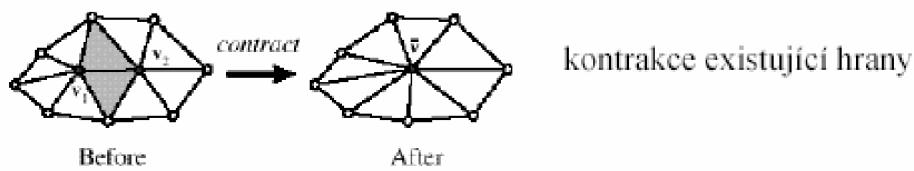
Zjednodušení s metrikou QEM

Surface Simplification Using Quadric Error Metrics

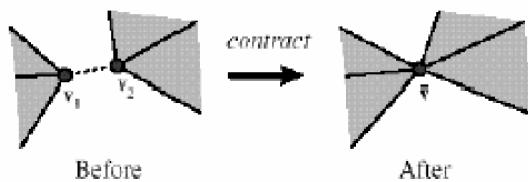
M.Garland, P.Heckbert, CMU, SIGGRAPH 97

$$(v_1, v_2) \rightarrow \bar{v}$$

párová kontrakce (dvojice vrcholů)



kontrakce existující hrany



kontrakce 2 vrcholů nepropojených hranou

Výběr párů pro kontrakce

Páry vrcholů pro budoucí kontrakce jsou vybrány předem v inicializační fázi
(v1,v2) je **platný pár** pro kontrakci, pokud

1. (v1,v2) je hrana, nebo
2. $\|v_1, v_2\| < t$, t je zvolený práh

Algoritmus zjednodušování povrchu s QEM

1. Vypočti matice Q pro všechny iniciální vrcholy
2. Vyber všechny platné páry.

3. "vypočti optimální kontrakční cíl v pro každý platný pár (v_1, v_2) . Chyba $\mathbf{v}^T(\mathbf{Q}_1 + \mathbf{Q}_2)\mathbf{v}$ je cennou kontrakce tohoto páru.
4. Všechny páry dej na haldu uspořádaně podle ceny, pár s nejmenší cenou kontrakce dej na vrchol.
5. Opakováně odeber pár s nejmenší cenou (v_1, v_2) z hladky, kontrahuji tento pár a oprav všechny ceny u páru obsahujících vrchol v_1 .

rovina $ax + by + cz + d = 0 ; a^2 + b^2 + c^2 = 1$

$$\Delta(v) = ([v_x v_y v_z]^T)^T = \sum_{p \in \text{roviny}(v)} (p^T v)^2 =$$

$$= \sum_{p \in \text{roviny}(v)} v^T (p p^T) v = v^T \left(\sum_{p \in \text{roviny}(v)} K_p \right) v$$

matice Q;
výchozí chybová metrika

$$K_p = p p^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

Ve výchozím modelu má každý vrchol sdružené roviny a odpovídající matice Q; matice metrik se přenášejí po kontrakci jako součet $Q = Q_1 + Q_2$

36. Progresivní síť. Princip a využití pro geomorfy.

Progresivne siete:

Povodnu suiet mozne zjednodusovat pomocu zrusenia hrany, a zozlozitovat pridavanim hran.

Problém optimalizace sítě spočívá v nalezení sítě $M=(K,V)$,

která co nejpřesněji aproxiimuje síťbodu x_i a má malý počet vrcholů.

Minimalizujeme energetickou funkci

$$E(M) = E_{\text{dist}}(M) + E_{\text{rep}}(M) + E_{\text{spring}}(M)$$

energie vzdáleností (původní body od sítě)

energie reprezentace (penalizuje počet bodův sítě m)

energie napružení (každá hrana je pružinou s klidovou délkom 0 a tenzí k)

Geomorfy:

Geomorf medzi dvoma reprezentaciami Progresivnej siete (pred a po deleni) ako siet, ktorá ma vrcholy nasledujucej reprezentacie a ich polohy interpolujeme medzi hodnotami pred a po deleni.

$$M^G(\alpha) = (K^{i+1}, V^G(\alpha))$$

$$v_j^G(\alpha) = \begin{cases} (\alpha)v_j^{i+1} + (1-\alpha)v_{s_i}^i & \text{delené vrcholy} \\ v_j^{i+1} = v_j^i & \text{nedelené vrcholy} \end{cases}$$

V. Zobrazování terénu

37. Aplikace LOD technik při zobrazování terénu. Odlišnosti v porovnání s obecnými povrchovými modely. Terén v pravidelné síti. Kritéria pro zjednodušení zobrazované sítě (Lindstrom, ROAM).

Cíl: udržovat dynamické, pohledovězávislé trojúhelníkové sítě a texturové mapy, které produkují dobré obrazy při požadované snímkové rychlosti

Pohledově závislé zjemňování sítě

- shora - dolů
- zjemňování až do splnění kritéria, dělení trojúhelníků
- zdola - nahoru
- zjednodušování, slučování trojúhelníků

Sítě

- nepravidelné, TIN (triangulated irregular networks)
- pravidelné, semipravidelné

Data

- výškové pole v pravidelné mřížce
- triangulované nepravidelné sítě

Terénní LOD vs. tradiční LOD

snazší

- omezená geometrie (obecně)
- specializovanější a jednodušší algoritmy

obtížnější

- spojité a velké modely
- současně velmi blízké a velmi vzdálené
- vyžaduje pohledovězávislé LOD
- mimo paměť

Pravidelné mřížky

- uniformní pole výškových hodnot
- jednoduché uložení a manipulace
- kódované v rastrových formátech
- snadná interpolace pro stanovení elevace
- menší paměť (ukládá pouze hodnoty z)
- snadné pohledové ořezávání a detekce kolize
- použito ve většině implementací: CLOD, ROAM

Lindstrom

- zavedli uživatelem ovládaný chybový práh v prostoru obrazovky
- každý vrchol má přiřazenu "chybovou výšku"
- při promítnutí na obrazovku je odhadnuta velikost této výšky
- při překročení prahu je vrchol „povolen“
- vrcholy jsou seskupeny do bloků seřazených podle chybové hodnoty.
- s použitím mezisnímkové koherence je omezen počet vrcholů testovaných v každém snímku

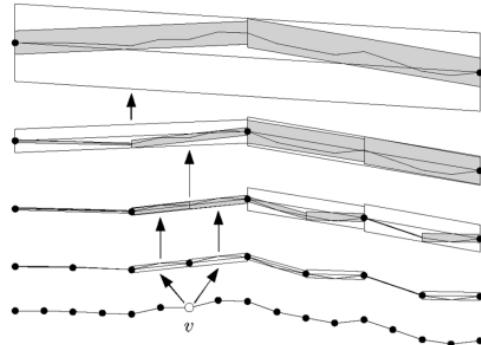
ROAM

- binární strom trojúhelníků

◆ kosoúhelník – hraniční objem

- Není nutné se starat o praskliny
- lze zvolit požadovaný počet trojúhelníků
- rozdělení a sloučení
- dvě prioritní fronty
- jedna pro rozdělování a druhá pro slučování
- umožňuje meziobrazovou koherenci
- chybové metriky pro rozdělení a slučování
- geomorfy – jsou možné, ale nejsou téměř potřeba
- zavádí inkrementální pásek trojúhelníků

- pokrývá (x,y) rozsah trojúhelníku a výškový rozsah od $z-e_T$ do $z+e_T$



38. Zobrazování terénu pomocí metody Geometry Clipmaps: datové struktury, princip algoritmu.

- Store data in uniform 2D grids
- Level-of-Detail from nesting of grids
- Refine based on distance
- Main Advantages
 - Simplicity
 - Compression
 - Synthesis
- Store data in uniform 2D grids
- Level-of-Detail from nesting of grids
- Refine based on distance
- Main Advantages
 - Simplicity
 - Compression
 - Synthesision

Data:

Terrain as mipmap pyramid

Individual Clipmap Levels

- Uniform 2D grid
- Indexed triangle strip

Alg:

Vytvorenie terenu cim dalej tym podla nízzej Mipmap urovne.

- Inter-Level Transitions
 - Vertex shader -> blend geometry
 - Pixel shader -> blend textures
- Clipmap Update(For each level)
 - Calculate new clipmap region
 - Fill new L-shaped region
 - Use toroidal arrays for efficiency

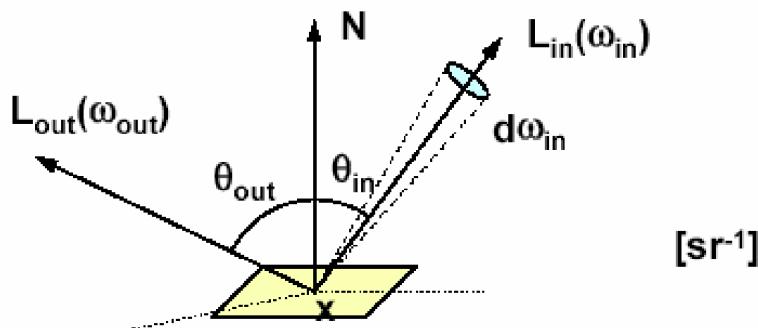
- Možnosť syntézy plochy pridaný vysokofrekvenčného sumu
- Možnosť kompresie levelov

VI. Globálne osvetlovanie, zobrazovacie rovnice

39. Obousmerná odrazová distribučná funkcia, vysvetlenie pomocou obrázku. Proč sa nazýva "obousmerná"? Příklady BRDF pro ideální difusní a zrcadlové povrchy.

Dvousmerná odrazová distribučná funkcia – BRDF

(Bidirectional Reflectance Distribution Function)



$$f_r(x, \vec{\omega}_{in} \rightarrow \vec{\omega}_{out}) = \frac{dL_{out}(x, \vec{\omega}_{out})}{dE(x, \vec{\omega}_{in})} = \frac{dL_{out}(x, \vec{\omega}_{out})}{L_{in}(x, \vec{\omega}_{in}) \cos \theta_{in} d\omega_{in}}$$

Vlastnosti BRDF

- Helmholtzov zákon – pro reálné povrchy těles platí:

$$f_r(x, \vec{\omega}_{in} \rightarrow \vec{\omega}_{out}) = f_r(x, \vec{\omega}_{out} \rightarrow \vec{\omega}_{in})$$

- BRDF nemusí být obecně isotropní (invariantní k otočení kolem simetrické osy), např. kovové povrchy leštěné v jednom směru

$$f_r(x, \theta_{in}, \varphi_{in}, \theta_{out}, \varphi_{out}) \neq f_r(x, \theta_{in}, \varphi_{in} + \alpha, \theta_{out}, \varphi_{out} + \alpha)$$

- BRDF je **normalizovaná** – množstvo odraženej energie bude menšie než (alebo rovno) množstvo dopadajúcej energie
- BRDF je **lineárny** vzhľadom k zářeniu – energia prichádzajúca z rôznych smerov prispívá nezávisle na výslednej odraženej energii
- Jednotkou je $[sr^{-1}]$

BRDF príklady

- pro **ideálne difusné** povrchy je BRDF konstantný.

$$f_{r,d} = const$$

- Pro **ideálne zrcadlové** povrchy je BRDF rovno:

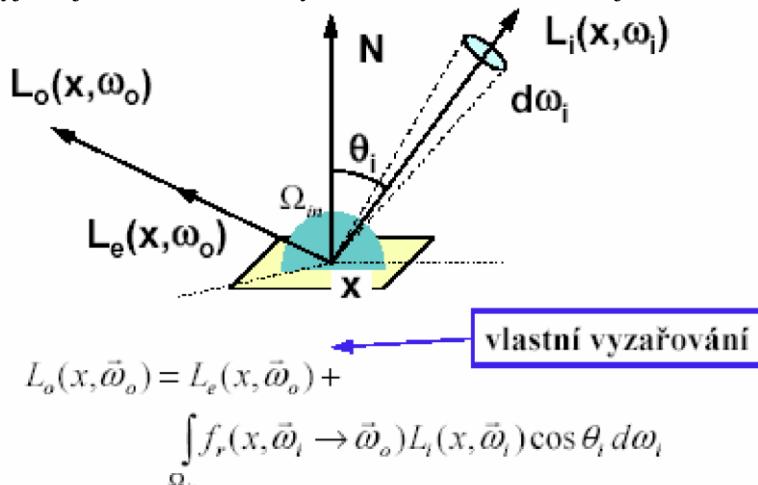
$$f_{r,s}((\theta_i, \varphi_i) \rightarrow (\theta_o, \varphi_o)) = \frac{\delta(\cos \theta_i - \cos \theta_o)}{\cos \theta_o} \cdot \delta(\varphi_i - (\varphi_o \pm \pi))$$

- ve skutečnosti zastoupeny obje složky, tzv. **lesklé** povrchy.

40. Co vyjadřuje rovnice VTIGRE (vacuum time invariant gray radiance equation)? Jaký je význam jednotlivých členů (materiál, prostorové uspořádání aj.).

Zobrazovací rovnice (VTIGRE)

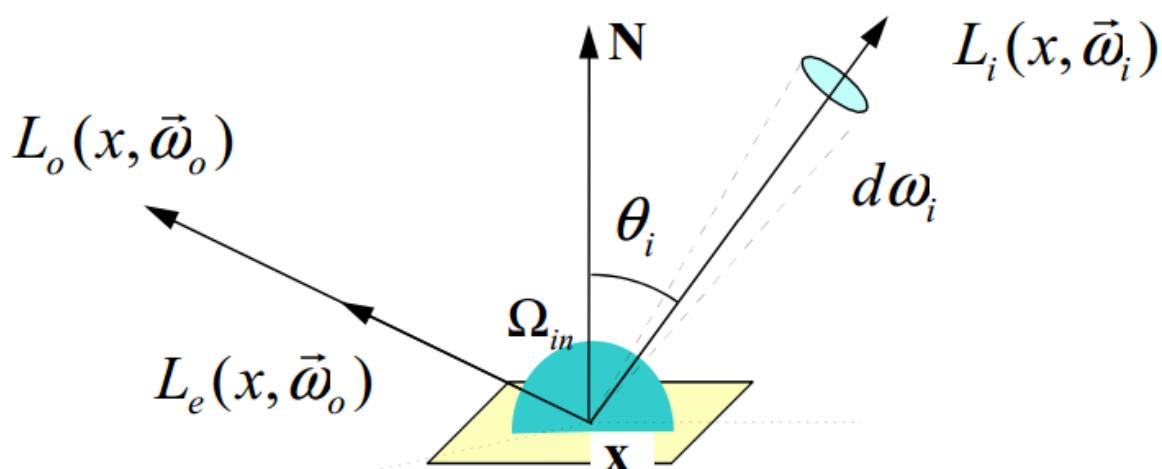
Vyjadruje kolko svetla sa vyziari a odrazi z daneho objektu.



To co sa vyziari= vlastne vyzazovanie objektu + integral cez vsetky uhly (odrazivost v tom uhle * svetlo prichadzajuce z toho uhlu*cos uhlu (cim kolmejsie tym silnejsie))

41. Zobrazovací rovnice Kajiya (OVTIGRE). Porovnání s VTIGRE. Jaký je význam jednotlivých členů (materiál, prostorové uspořádání aj.).

Vyjadruje kolko svetla sa vyziari a odrazi z daneho objektu.



$$L_o(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \int_S f_r(x, x' \rightarrow x, \vec{\omega}_r) L_i(x' \rightarrow x) G(x, x') V(x, x') dA \quad (3.5)$$

$L_e(x, \vec{\omega})$	Dvoubodová intenzita vlastního emitovaného světla
$L_r(x, \vec{\omega}_r)$	Odražená radiance
$V(x, x')$	Člen určující viditelnost
$G(x, x')$	Geometrický člen
A	Ploška, na kterou přichází radiance
A'	Ploška, z které přichází radiance
x	Bod na A (vyšetřovaný bod)
x'	Bod na A'

Tabulka 2: Tabulka termínů – zobrazovací rovnice.

Člen určující viditelnost může obsahovat jednu ze dvou hodnot. Pokud jsou body x a x' navzájem viditelné, pak má hodnotu jedna, v ostatních případech nabývá hodnoty nula. Geometrický člen (obrázek 3.2) v sobě zahrnuje koeficienty vyjadřující promítnutí radiance po vyzáření z bodu x' a následném dopadu do bodu x .

$$G(x, x') = G(x', x) = \frac{\cos \theta_i \cos \theta_o}{|x' - x|} \quad (3.6)$$

Rovnice tedy vlastně vypočítá radianci odcházející v nějakém směru z bodu x , který se nachází na nějaké ploše. Celková odcházející radiance se skládá z vlastní emise a odařené radiance, kterou tvoří ze součtu radiancí přicházejících ze všech viditelných ploch, vynásobený geometrickým členem a BRDF materiálu.

Porovnanie:

VTIGRE pracuje so svetlom z daneho uhlu, OVTIGRE pracuje s prispevkami jednotlivych ploch v scene.

42. Pomocí obrázků a operátorů šíření světla (E,L,D,S) charakterizujte jednotlivé metody výpočtu osvětlení: RayTracing, radiozita, obousměrné sledování cest, kombinace radiozity a RT (dvoukroková metoda).

Zobrazovací metody (RT)

- **Rekurzivní sledování paprsku**

(eye ray-tracing, forward ray-tracing, visibility tracing)

- **BRDF** udává váhu sekundárního paprsku

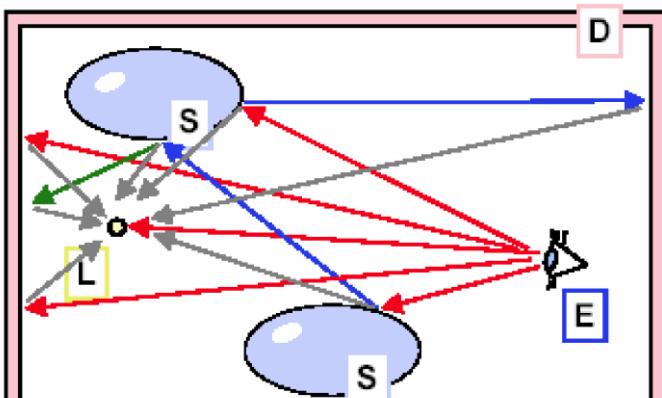
- Lesklé plochy: zrcadlový paprsek má obvykle největší váhu, a proto ho sleduje dále
- Difusní plochy: vše má stejný příspěvek – co dál???

- Nepokryté cesty nahradím **amgentní složkou**

- První lesklý odraz se počítá přesně, ostatní se nahrazují ideálním zrcadlovým odrazem

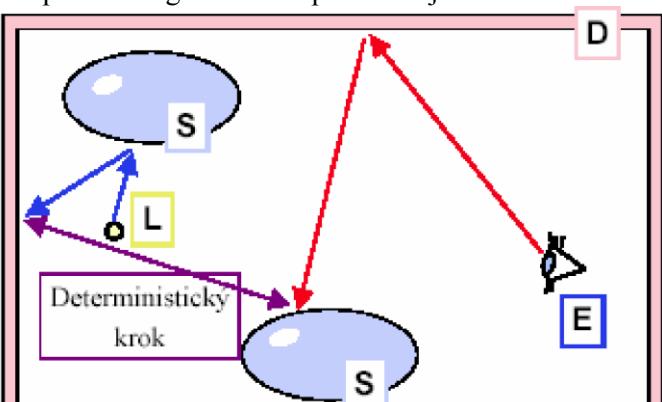
- **L[D|S]Sm*E**

Sledování paprsku (ray-tracing)



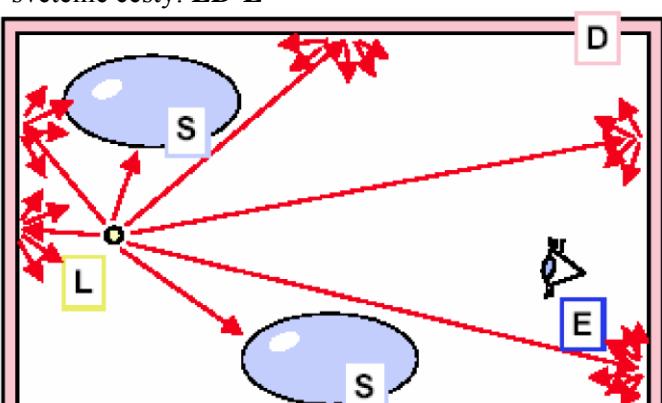
Obousměrné sledování cest

- **Obousměrné sledování cest** kombinuje sledování cest a sledování světla.
- Po několika náhodných krocích ze zdroje a od pozorovatele se provede **deterministický krok**, který spojí konce (nebo vnitřní body) obou cest.
- Lepší konvergence než u předcházejících metod.



Radiozitní metoda

- vícenásobné difusní odrazy
- nerozlišuje primární a sekundární záříče
- řešení metodou konečných prvků
- závisí na kvalitě sítě ploch a elementů
- pohledově nezávislé řešení
- světelné cesty: **LD*E**

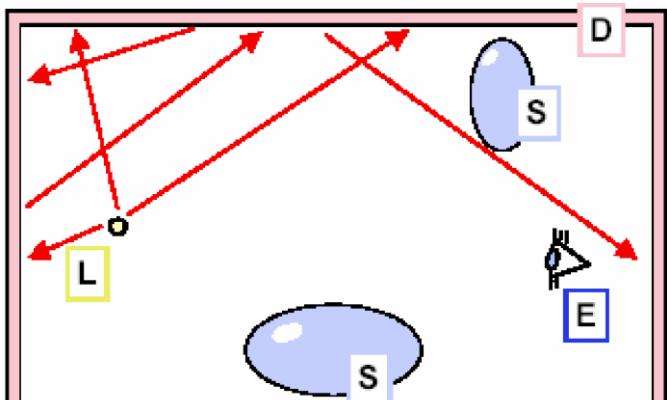


Jednoduchá dvoukroková metoda

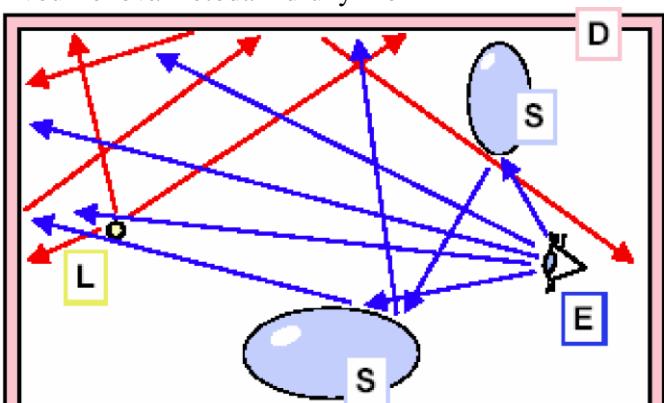
- klasická radiozitní metoda: **LD***
 - o výpočet osvětlení na všech ploškách scény
 - o tento krok není závislý na pozici pozorovatele

- zobrazení sledováním paprsku: **Sm*E**
 - o nepočítají se stínovací paprsky => velká rychlos
 - o místo světelného modelu se používá **předem spočítaná radiozita**
 - o tento krok závisí na úhlu pohledu

Dvoukroková metoda – první krok



Dvoukroková metoda – druhý krok



43. Z jakých omezujejících podmínek vychází řešení klasické radiozitní úlohy?

Zjednodušený fyzikální model Radiozitní rovnice

- přepokládáme **ideálně difusní povrchy**:

 - o **BRDF** není závislá na vstupním a výstupním úhlu
 - o Výstupní zář $L(x,\omega)$ nezávisí na směru ω

$$L_o(x) = L_e(x) + \int_S f_r(x) L_o(y) G(x,y) V(x,y) dA_y$$

$$L_o(x) = B(x)/\pi \quad L_e(x) = B_e(x)/\pi \quad f_r(x) = \rho(x)/\pi$$

$$B(x) = B_e(x) + \frac{\rho(x)}{\pi} \int_S B(y) G(x,y) V(x,y) dA_y$$

Radiozitní metoda

- vícenásobné difusní odrazy
- nerozlišuje primární a sekundární zářiče
- řešení metodou konečných prvků

- závisí na kvalitě sítě ploch a elementů
- pohledově nezávislé řešení
- světelné cesty: **LD*E**

VII. Rastrové obrazy

[44. Otočení obrazu vícekrokovou metodou. Princip výpočtu, varianty vícekrokových metod.](#)

V dvoch krokoch bez korekcie mierky:

Dve skosenia, najskor v ose x potom v ose y. Modifikuje sa obrazok len riadkovo ci stlpcovo. Jeden pixel sa zobrazí na max 2.

V dvoch krokoch s korekciou mierky:

$$R = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \tan \alpha & \frac{1}{\cos \alpha} \end{pmatrix} \begin{pmatrix} \cos \alpha & -\sin \alpha \\ 0 & 1 \end{pmatrix}$$

V styroch krokoch:

Dve skosenia a nasledne dva scaly (po osach) . Modifikacia len jedneho riadku vzdy.

V troch krokoch:

Tri skosenia, bez nasobenia mierkou. Jeden pixel sa zobrazí na max 2.

$$R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\tan(\theta/2) & 1 \end{bmatrix} \begin{bmatrix} 1 & \sin \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\tan(\theta/2) & 1 \end{bmatrix}$$

CORDIC algoritmus

Rotacia zlozena z dielcich rotacii.

Obecne:

$-45 \leq \alpha \leq 45$: dvoukroková metoda

$-45 \leq \alpha \leq 45$: tříkroková metoda

Pri vacsich uhloch pomocné otočení o násobek 90° .

[45. Deformace obrazu pomocí trojúhelníkové sítě. Princip výpočtu.](#)

Barycentrické souřadnice určují polohu bodu v trojúhelníku

$$p = \lambda_0 p_0 + \lambda_1 p_1 + \lambda_2 p_2$$

$$W(p) = \lambda_0 w(p_0) + \lambda_1 w(p_1) + \lambda_2 w(p_2)$$

Postup:

Vytvorí sa vhodna trojuholnikova siet.

Vypocitaju sa barycentricke suradnice jednotlivych bodov v povodnom obrazku.

Tato siet sa zdeformuje.

Body povodneho obrazka sa zobrazia do noveho s pomocou svojich barycentrickych suradnic.

Obecne sietovy warp

je obtížné vložit „vhodnou“ síťdo obrázku

globální vliv sítěpri zméně řídících bodů

nedostatečné ovládání v místech, kde potrebujeme ovlivnit podrobnosti

[46. Deformace obrazu pomocí splajnové sítě. Princip výpočtu.](#)

Přesun bodův obraze určuje splajnová osnova.

Splajny vytvoří hladké vektorové pole.

2-průchodový síťový warp

- Splajny určené řídícími body specifikují sítkřivek na obrazu
- Vstup
 - Zdrojový (& cílový) obraz
 - 2D pole řídících bodůve zdroji
 - 2D pole řídících bodův cíli

Postup

1. Prolož křivky kontrolními body (např. Catmull-Rom splajny)

2. (1.průchod): Spáruj úseky ve směru x a interpoluj

3. (2.průchod): Spáruj úseky ve směru y v bezprostředním obraze (výsledek kroku 2) a interpoluj

4. Umísti obraz (nebo obrazy) do nové mřížky

Obecne sietovy warp

je obtížné vložit „vhodnou“ síťdo obrázku

globální vliv sítěpri zméně řídících bodů

nedostatečné ovládání v místech, kde potrebujeme ovlivnit podrobnosti

[47. Deformace obrazu pomocí párových rysů. Princip výpočtu.](#)

Namísto křivek použij páry úseček pro určení korespondence

1 průchod, který vypočte vážené příspěvky od každého páru ke každému pixelu

Vstup:

- zdrojový obraz
- (cílový obraz)
- páry úseček PQ_S a PQ_D

Vypocet

Pro každý bod X cílového obrazu stanovíme

souřadnice vůči rysu a ve zdrojovém obraze

vypočteme X'_i pro každý pár rysů

Váženým součtem X'_i určíme cílovou polohu X'

Barvu z místa X' přeneseme do X

Vaha:

$$weight = \left(\frac{length^p}{a + dist} \right)^b$$

a – hladkost pokřivení b – úbytek síly se vzdáleností p – ocenění delších úseček

výhody

- expresivní
- lze snadno přidávat řídící body

nevýhody

- při zpracování pixelůmusí být vzaty v úvahu všechny úsečky
- úsečky mají globální vliv

48. Míchání obrazů při metamorfóze, použití warfování před interpolací.

Metamorfoza:

Interpolacia z jedneho obrazka(objektu) do druhého.

CrossDissolve:

Linearna interpolacia dvoch obrazov, nevhodne ak obrazy nezarovnane.

Morphing:

Naist priemer dvoch objektov (nie obrazkov). Neda sa automaticky, nutny umelecky zasah.

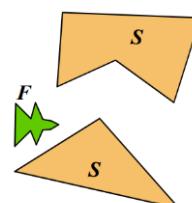
Feature matching – najdenie podobnych rysov na obrazkach a lokalnym zmackanim previes rysy z jedneho na druhy

Morfuje sa tvar z prveho na druhy, z druhého na prvý a nasledne sa linearne prechadza z jedneho do druhého cez CrossDissolve.

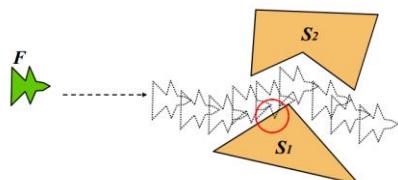
VIII. Detekce kolizí těles

49. Formulujte základní přístupy k řešení kolizí (statická, pseudodynamická, dynamická).

Mějme scénu S a pohybující se objekt F. Naším úkolem je zjistit, jestli objekt F přišel v nějakém okamžiku do kontaktu se scénou S.

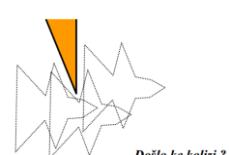


- Statická detekce kolizí : Provádíme jeden test v jedné konkrétní pozici a orientaci objektu F.
- Pseudo-dynamická detekce kolizí: Provádíme testy na diskrétní množině pozic a orientací objektu F odpovídající



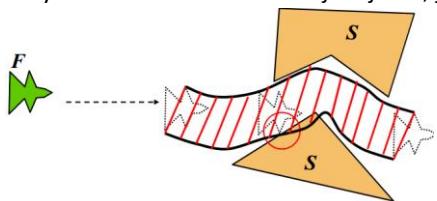
pohybu objektu F.

Mozny problem pri rychlom pohybe.



Došlo ke kolizi?

- Dynamická detekce kolizí: Zjišťujeme, jestli prostor vzniklý tažením objektu F protíná scénu S.



50. Využití hierarchických reprezentací BVH a SP pro detekci kolize, příklady datových struktur.

Rozklad prostoru (čtyřstěny, voxely, BSP) a algoritmy kolize v děleném prostoru, obecný princip.

- Hierarchie obalových objemů(Bounding Volume Hierarchy, **BVH**)

– stromy koulí, krychlí, osově-zarovaných kvádrů...

- Objektocentrismus

- Prostorová redundance (jeden priestor vo viacerých BV)

– každý uzel obsahuje jeden objem - obal, který omezuje množinu trojúhelníků

– potomci obsahují obaly, které ohraňují odlišné části (skupiny) trojúhelníků daného rodiče

– listy hierarchie obvykle obsahují individuální trojúhelníky (často však seznamy trojúhelníků)

– Nejčastěji se používá binární hierarchie obalů

- Prostorová dekompozice (Space Partitioning, **SP**)

– rozdelenie priestoru na podpriestory, pri ktorých lachko určime kolizi s nejakym BB.

- nasledne skumanie kolizie len s objektami v podpriestore.

– BSP, K-d stromy, oktalové stromy, ...

- Prostorocentrismus

- Objektová redundance (jeden object vo viacerých castiach priestoru)

- PR: Uniformní dělení, Quadtree/Octree,kd-strom, BSP-strom

- Klady a záporu hierarchií

– Pracují dobře při testech "odmítnutí", kdy jsou objekty dále od sebe

– Výkon může klesnout, pokud jsou dva objekty blízko sebe a mají více kontaktů

Priestor je možné rozdeliť do pravidelných buniek (v 2D stvorsteny, v 3D voxely) do ktorých vložíme odkazy na objekty, ktoré ich pretinajú. Nasledne kolizie detektujeme len v bunkach, ktoré majú odkaz na dva objekty. Veľke nároky na pamäť. Uspora cez OctTree alebo QuadTree, kde delime priestor na 4(2d) alebo 8 (3d) pravidelných castí podľa potreby (ak viac objektov v jednej casti tak ju rozdelime pre jemnejší popis objektu).

Alg kolize pre novy objekt v spocitanom deleni na voxely: vypocita BB skumaneho objektu, najdi vsetky bunky ktore tento BB pretina, preved test kolizie medzi skumanym objektom a objektami v bunkach.

BSP (binary space partitioning, binárni rozdelení prostoru) je způsob rozdelení prostoru pomocí binárního stromu. Výsledný strom ve svém kořenu obsahuje rovinu, která všechny objekty v prostoru dělí na 2 skupiny (ležící před a za rovinou). Potomci kořene pak reprezentují vzniklé skupiny, jež jsou opět rekursivně děleny nově zvolenou rovinou na 2

části. Listy stromů pak obsahují vhodné skupiny objektů (nejčastěji se jedná o množinu polygonů, které tvoří konvexní celek, tj. žádná rovina určena polygonem neprotne jiný polygon ze skupiny).

Alg kolize pre novy objekt v spocitanom deleni BSP: vypocitaj BB objektu. Zacni v koreni stromu. Ak BB pretina priestor laveho podstrmu tak rekurzivne spusti vypocte vlavo. Ak BB pretina priestor praveho podstrmu tak rekurzivne spusti vypocte praveho. Ak si v liste tak spocitaj kolizie objektu s objektem v uzle.

51. Konvexní obaly a hierarchie obalů (koule, AABB, k-DOP, OBB). Algoritmy kolize s hierarchií obalů, obecný princip.

Nákladová funkce použití hierarchie obalů pro detekci kolize.

Koule

- nejjednodušší obalové těleso
- rychlý test protnutí mezi dvěmasférami (stačí porovnat vzdálenost středů porovnávaných sfér se součtem poloměrů)
- nejméně citlivé na tvar tělesa, největší procento obsaženého volného prostoru

AABB – axis aligned bounding box

- jednoduchá konstrukce
- v testu překrytí stačí porovnat jednotlivé strany krabice, což je pro osově orientované krabic
- velmi jednoduché
- málo citlivé na tvar tělesa

Obal k-DOP

- konvexní mnohostěn, jehož plošky jsou určeny poloprostory, jejichž vnější normály patří do malé, pevně dané množiny k směru
- v testu se porovnávají vždy pouze rovnoběžné roviny stran sousedících těles

Obal OBB

- OBB – oriented bounding box
- osy určeny vlastními vektory kovarianční matici
- dobrý obal OBB kopíruje tvar tělesa ... ale je obtížnější pro vytvoření
- test kolize 2 obalů je rychlý, stačí provést 15 jednoduchých testů.

BVH – bounding volume hierarchy:

- každý uzel obsahuje jeden objem - obal, který omezuje množinu trojúhelníků
- potomci obsahují obaly, které ohraňují odlišné části (skupiny) trojúhelníků daného rodiče
- listy hierarchie obvykle obsahují individuální trojúhelníky (často však seznamy trojúhelníků)
- Nejčastěji se používá binární hierarchie obalů

- Obalové těleso by mělo obklopovat původní model tak těsně, jak jen to je možné (minimalizace N a P).
- Testování dvou obalových těles by mělo maximálně rychlé (minimalizace B).
- Konstrukce stromu by měla být maximálně rychlá
- Výsledný strom by měl poskytovat co jak nejlepší výsledky

Pristupy:

- Top-down / Bottom-up
- Vytváření stromu postupným dělením scény shora dolů, nebo sjednocováním menších částí zdola nahoru

• Stupeň stromu

Binární vs. jiné stromy

• Statický strom / Dynamický strom

Bude se často provádět update vstupních dat? Je vhodnější vybudovat vždy stromovou strukturu znova od začátku, nebo se jí pokusit upravit?

- Výběr dělících pravidel
- Vyber osu s nejdelší stranou tělesa
- Vyber osu s největší variancí těžiště trojúhelníků
- Vyber osu, která minimalizuje součet objemů vzniklých těles
- Vyber osu, která minimalizuje největší z objemů vzniklých těles

Volba bodu na dělicí ose, kde povede dělící nadrovina

– Střední hodnota těžiště trojúhelníků

– Střed těžiště trojúhelníků

Detekcia kolizie:

Testuj kolizi mezi dvěma rodičovskými uzly (od kořenů daných stromů)

If není překrytí mezi dvěma rodiči,

then stop a výsledek "bez kolize";

/* else */

Sestup: Všichni potomci jednoho rodičovského uzlu jsou testováni proti všem potomkům druhého rodiče

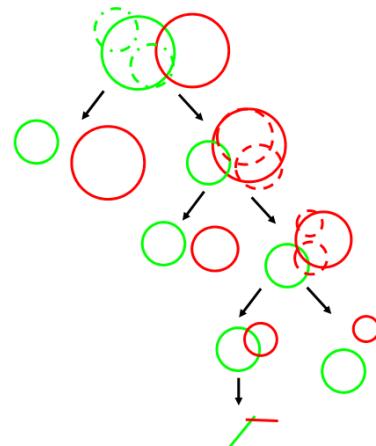
If je kolize mezi potomky

then If jsou to listové uzly

then nahlaš "kolizi"

else jdi na **Sestup**

else stop a výsledek "bez kolize"



Cenová funkce:

$$F = N_u \times C_u + N_{bv} \times C_{bv} + N_p \times C_p$$

F : celková cenová funkce pro detekci kolize

N_u : počet upravených BV

C_u : cena úpravy BV,

N_{bv} : počet BV páru testovaných na překrytí

C_{bv} : cena testu překrytí mezi párem BV

N_p : počet páru primitiv testovaných na kolizi

C_p : cena testu kolize mezi párem primitiv

IX. Okluze

52. Druhy viditelnosti (bod, oblast), množina objektů PVS.

viditelnost z bodu

– seznam objektů úplně nebo částečně viditelných z bodu

viditelnost z oblasti

– seznam objektů úplně nebo částečně viditelných alespoň jednoho místa oblasti

Potenciálně viditelná množina (PVS)

– konzervativní nadhodnocený odhad viditelných polygonů

– obvykle řešeno shlukováním polygonů a rozhodnutí o potenciální viditelnosti shluku rychlým testem

– obdoba hierarchických hraničních objemů nebo dělení prostoru pro ray-tracing

53. Techniky odstřelování. Přehled, použití, porovnání.

bez předzpracování, vhodné pro scény s dynamickými změnami, s interakcí uživatele

redukce dat - odstřelovací techniky

dynamické objekty mohou plnit funkci okluzních objektů

využití časové koherence

u některých metod částečné předzpracování

– virtuální okluzní objekty, hierarchické okluzní mapy

Culling

“Odstřelit” znamená “vybrat ze skupiny”

V grafice: nezpracovávej data, která nepřispějí nebo „nepřispějí“ k výslednému obrazu

“Skupina” je celá scéna, výběrem „na odstřel“ je podmnožina ze scény, jejíž příspěvek nepovažujeme za významný

Prehľad

- Odstřelování zadních ploch (back-face culling)

-jednoduše a rychle, nutne mat konzistentne orientovane plochy (podla poradia bodov).

-Implementované v API

- (Hierarchické) odstřelování podle tělesa záběru (view frustum culling)

- Ohraničení každé “přirozené” skupiny primitiv jednoduchým objemem (např. koule, kvádr)

- Pokud je hraniční objem (BV) mimo těleso záběru, pak je také obsah BV mimo (neviditeľný)

-Vyhni se zpracování těchto BV a jejich obsažené geometrie

- Portálové odstřelování (portal culling)

- procházení architektonickými modely(budovy, města, katakomby...)

-Tyto modely se dělí přirozeně buňky (pokoje, výklenky, chodby...)

- Průhledné portály propojují buňky (dveře, vchody, okna...)

-Klíčový poznatek: buňky se vzájemně vidí pouze skrze portály!

- Odstřelování detailů (detail culling)

-Idea: objekty, jejichž promítnutý BV zabírá méně než N pixelů, jsou odstřeleny

-Je to approximativní algoritmus, protože odstřelené objekty mohou ve skutečnosti přispívat k výslednému obrazu

-Výhoda: kompromis kvalita/rychlosť

-Případné LOD : použití různých úrovní detailu na různých vzdálenostech od pozorovatele

- Odstřelování zakrytých částí (occlusion culling)

-Hlavní idea: objekty, které leží zcela “za” jinou množinou objektů, mohou být odstřeleny

-Obtížný problém pro efektivní řešení

-kresli zpredu dozadu, a pameta si horizont zakrytu, ak BB objektu zakryty tak sa nevykresluje

54. Portálový odstřel, nadhodnocené portály, princip a charakteristika metody.

Info

- procházení architektonickými modely(budovy, města, katakomby...)

-Tyto modely se dělí přirozeně buňky (pokoje, výklenky, chodby...)

- Průhledné portály propojují buňky (dveře, vchody, okna...)

-Klíčový poznatek: buňky se vzájemně vidí pouze skrze portály!

Idea:

Vytvoř graf sousedství buněk

Počínaje buňkou obsahující pozorovatele procházej graf a vykresluj viditelné buňky

Buňka je viditelná jedině tehdy, když ji lze spatřit skrze sekvenci portálů

- Potřebujeme přímku pohledu
- viditelnost buňky se zredukuje na testování portálových sekvencí ...

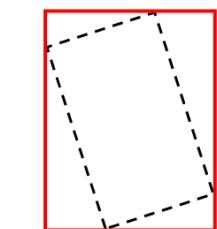
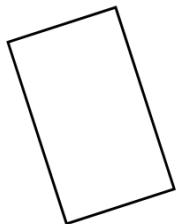
Alg:

Rozdělení na buňky s portály (vytvoření grafu)

Pro každý snímek:

- lokalizuj buňku pozorovatele a inicializuj 2D AABB pro celou obrazovku
- * vykresli současnou buňku - aplikuj VF odstřel omezený na AABB
- přejdi k nejbližším buňkám (přes portály)
- Průnik aktuálního AABB & AABB procházeného portálu
- Jdi na *

◆ Pro zjednodušení:



skutečný portál

nadhodnocený portál

kvoli rychlosti vypočtu.

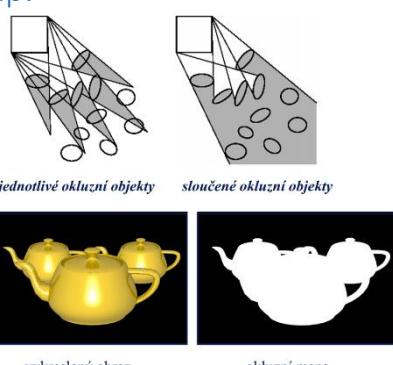
55. Slučování okluzních objektů, odstřel pomocí okluzních map.

- objekty, které leží zcela "za" jinou množinou objektů, mohou být odstřeleny

Pro každý objekt g dělej:

```
if (not Occluded( $OR$ ,  $g$ ))  
    vykresli ( $g$ );  
    aktualizuj ( $OR$ ,  $g$ );  
end;  
end;
```

OR-okluzna reprezentacia



Slučování

Několik malých okluzních objektů sloučíme do většího „virtuálního“ objektu

- kumulované zakrytí má podstatně větší sílu

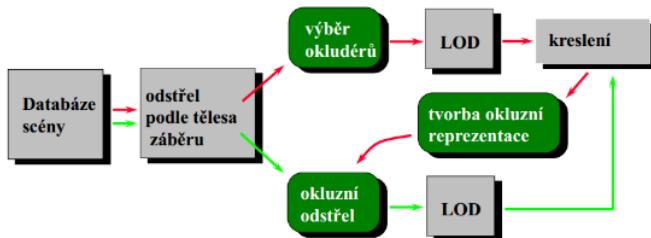
- Potřebujeme reprezentaci okluze (OR)

- rychle vypočítat
- rychle používat

- možné testy prekrycia a hlbky

alg s Potencionalnymi Okludermi:

- Vyber okludery, až je PO dostatečně velká
- Uprav (vytvoř) okluzní reprezentaci
- Okluzní odstřel & konečné vykreslení



Okluzní mapa

- odpovídá dělení obrazovky
- zaznamenává průměrnou neprůhlednost pro každé dělení
- Může být vytvořena vykreslením okludérů (záznam neprůhlednosti pixelů(pokrytí pixelů))
- Průnik průmětů okludérů
- Reprezentace okluze v prostoru obrazu

Testy prekrycia:

Vykresli BB na obrazovku, skontroluj ci je cely BB zakryti okluznou mapou, ak ano tak ho nevykresluj.

X. Výpočetní geometrie

56. Bentley-Ottmanův algoritmus hledání průsečíků úseček, princip, složitost.

Princip:

Použijeme „sweep line“

Setřídíme množinu koncových bodů podle osy x.

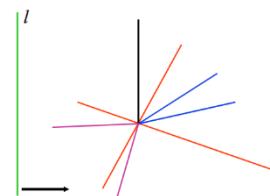
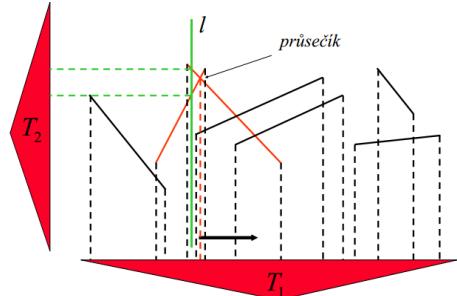
Budeme postupovat zleva doprava a budeme pracovat pouze s úsečkami, které protínají pomyslnou vertikální přímku.

Problemy

Vyskytuju sa speciálne prípady ktoré je nutné extra riešiť

Složitost:

- Setřídění bodů v ose x: $O(n \log n)$
- Operace nad stromem T_1 :
(opakujeme $(4n + I)$ krát) $O(\log n)$
- Operace nad stromem T_2 :
(opakujeme $(4n + I)$ krát)
Pozn.: I – počet průsečíků
- Celkem:
 $O(n \log n) + (8n + 2I)O(\log n) = O((n + I)\log n)$



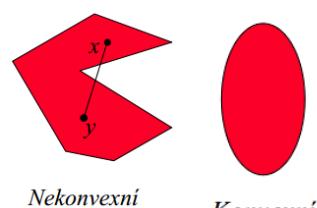
Optimalné pre problem:

$$O(I + n \log n)$$

57. Konvexní obaly. Definice, algoritmy konstrukce, složitost.

Konvexní množina:

Množina S se je konvexní právě tehdy, jestliže pro každou dvojici bodů x, y patřících do S platí, že úsečka xy je zcela obsažena v množině S .



Konvexní obal – Convex Hull, CH:

- Konvexní obal $CH(S)$ množiny S je nejmenší konvexní množina, která obsahuje S .

- Průnik všech konvexních množin obsahujících S.
- Průnik všech poloprostorů obsahujících S
- nejmenší konvexní polygon obsahující S
- sjednocení všech trojúhelníků určených množinou S

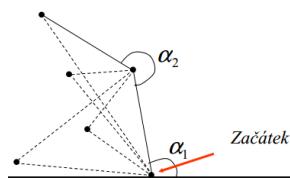
Naivní algoritmus

1. Projdi všechny trojúhelníky a označ všechny body, které leží uvnitř nějakého trojúhelníku jakožto ne-extrémní.
2. Vypiš všechny neoznačené body jakožto extrémní.

Zložitost $O(n^4)$

GiftWrapping

Zacina z extremného bodu a postupne obaluje množinu hranami ktoré majú najväčší uhol.



```

1. Nalezni nejnižší bod;
2. Nechť  $i_0$  je jeho index,  $i = i_0$  ;
3. repeat
   for každé  $j \neq i$  do
      vypočítej úhel  $\alpha$  mezi úsečkou  $(j,i)$  a úsečkou
      z predchôzihho konvexného obalu;
      Nechť  $k$  je index bodu s najmenším úhlem  $\alpha$ ;
      Dej na výstup hranu  $(p_i, p_k)$  akožto hranu konvexného
      obalu;
       $k = i$ ;
   until  $i = i_0$ ;

```

Zložitost $O(n*k)$ kde k je počet bodov na CH, tým ale može byť až n teda celková zložitost je $O(n^2)$

Graham Scan

Samostatne vytvara hornu a dolnu cast CH.

Postupne pridava body zoradene podla X a zistuje ci nova množina je convextna (podla uhlu s predchadzajucou hranou)

Ak nie tak predposledny bod odstrani a pyta sa rekurzivne dalej.

Zložitost $O(n \log n) + 5n = O(n \log n)$ zoradenie v $n \log n$ a nasledne raz bod pridame a pripadne raz odoberieme pre hornu a dolnu cast.

Quick Hull

Zacina s dvoma bodmi a,b , najde c ktorý má najväčšiu vzdialenosť od priamky (a,b). Rekurzívne pokracuje na polpriestoroch od priamky (a,c) a (b,c) .

Zložitost $O(n^2)$ ale v praxi lepsu

Optimalna zložitost problemu je $O(n \cdot \log(k))$

58.Triangulace. Naivní algoritmus. Metoda monotónních polygonů.

Naivný alg: postupne pridavaj vsetky mozne hrany a testuj konzistenciu, ak porusena tak ju nepridaj. $O(n^3)$

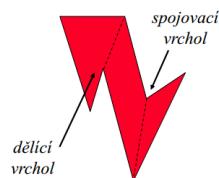
Jednoduchý polygon se nazývá **monotónní vzhľadem k priamce l**, jestliže pro každou priamku l kolomou na l je průnik polygonu s priamkou l spojity.

Rozbitie polygonu na monotonne:

- Pro každý dělící vrchol přidáme diagonálu jdoucí z tohoto vrcholu směrem nahoru.
- Pro každý spojovací vrchol přidáme diagonálu jdoucí z tohoto vrcholu dolů.

Alg:

Použijeme opět metodu „sweep“. Budeme postupovat shora dolů, události nastávají ve vrcholech polygonu.



Události jsou uloženy ve frontě událostí, která je prioritní frontou . Priorita vrcholu je jeho y-souřadnice.

Vrcholy se stejnou y-souřadnicí jsou porovnávány podle x-souřadnice (lexikografické třídění).

Během „sweepu“ nebudou vznikat nové události.

Pomocník je nejnižší vrchol mezi hranou a nejbližší hranou k nej vpravo . Pokud takový vrchol neexistuje, tak je to horní vrchol hrany .

Během průchodu polygonem si pamatujeme ke každé hraně, kterou jsme potkali, jejího pomocníka.

V okamžiku, kdy dosáhneme vrcholu , který přepíše vrchol jakožto pomocníka hrany , můžeme použít tento vrchol pro diagonální řez.

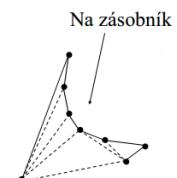
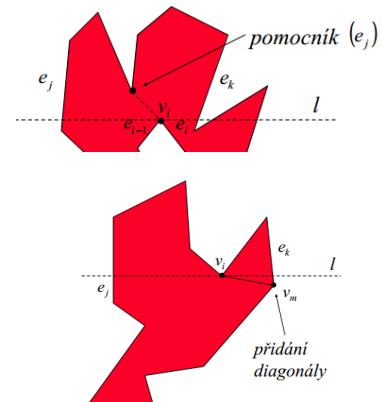
V případě, že ke změně pomocníka nedojde, použijeme dolní vrchol hrany

Pre rychlosť sú hrany pretinajúce SweepLine uložene v bin vyhľadavacím strome.

Zložitost $O(n \cdot \log(n))$

Triangulacia monotonného trojuholníka

Budeme postupovať z nejvyššího vrcholu monotonného polygonu směrem dolů obou smerech současně. Při práci využijeme zásobník.



59. BSP stromy. Typy, konstrukce, využití v grafice.

BSP stromy jsou binární stromy sloužící k prostorovému třídění scény obecně v rozměrném prostoru

Celá scéna je rekuzivně dělena nadrovinou. Listy BSP stromu odpovídají konvexnímu rozkladu scény.

Prostor určený jedním listem obsahuje maximálně jeden objekt (případě zlomek objektu)

Konstrukce:

Rekuzivne del mnozinu polygonov.

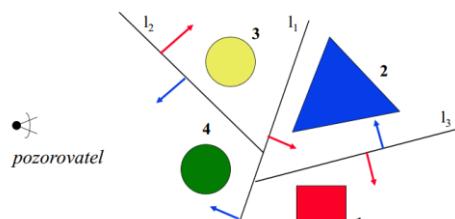
Ak zostal viac ako jeden polygon tak ich rozdel podla deliacej roviny na dve skupiny. Z deliacej roviny urob uzol stromu. Rekuzivne zavolaj alg na obidvoch skupinach polygonov a vytvor z nich potomkov uzlu.

Ak zostal len jeden tak z neho sprav list.

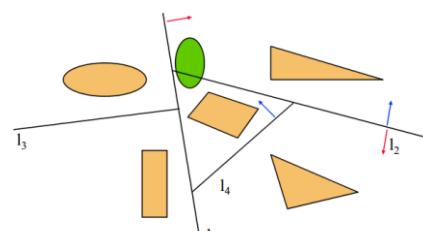
Varianty tvorby BSP stromů

Dělící nadroviny jsou orientovány: v libovolném směru / pouze v ortogonálních smerech

Pouzitie:



V maliarovom alg.



Detekce kolizí těles

Speciálne prípady:

KD stromy - striedavo delia pristor osovo orientovanymi priamkami

QuadTree, OctTree – delia priestor v ortogonalnych smerov na polovice v kazdej dimenzi.

60. Rozsahové vyhľadávání. Formulace úlohy, využití K-D stromů.

Úloha

Mějme k-rozměrný stavový prostor \mathbf{R}^k , množinu \mathbf{U} n-bodův tomto prostoru a libovolné spojité těleso (rozsah) T z tohoto prostoru.

Obecnou úlohou rozsahového vyhľadávania je určiť maximální podmnožinu V množiny U takovou, že všechny body z V leží v T .

V případě, že těleso T je osově orientovaný k-rozměrný kvádr, jedná se o tzv. Ortogonální rozsahové vyhľadávání.

K-D tree (K-D strom)

-vyvážený binárni vyhľadávací strom provádějící dělení střídavě podle x-ové a y-ové souřadnice.

Region(v) odpovídající uzlu v je obdélník, který může být neomezený v jednom nebo více směrech.

Region kořenu KD-stromu je celá rovina

Bod je uložen v podstromu uzlu v právě tehdy, jestliže leží v region(v).

Podstrom uzlu v prohledáváme pouze tehdy, jestliže region(v) má neprázdný průnik s dotazovaným obdélníkovým rozsahem.

Jestliže je region(v) zcela obsažen v dotazovaném rozsahu, můžeme nahlásit všechny body celého podstromu v.

61. Rozsahové stromy, princip, použití.

Range tree (Rozsahový strom)

-vyvážený binárni vyhľadávací strom setříděný podle jedné ze souřadnic. Uzel obsahuje buď hledaný bod, nebo podstrom, který je rozsahovým stromem pro nalezený rozsah a následující souřadnici.

A range tree on a set of 1-dimensional points is a balanced binary search tree on those points. The points stored in the tree are stored in the leaves of the tree; each internal node stores the largest value contained in its left subtree. A range tree on a set of points in d-dimensions is a recursively defined multi-level binary search tree. Each level of the data structure is a binary search tree on one of the d-dimensions. The first level is a binary search tree on the first of the d-coordinates. Each vertex v of this tree contains an associated structure that is a $(d-1)$ -dimensional range tree on the last $(d-1)$ -coordinates of the points stored in the subtree of v .

Využitie:

Rychlejsie vyhľadávanie ako KDTree ale zabera viac pameta (kvôli tomu že kazdy uzol ma vnorený podstrom)

Vyhľadávanie dotazov na rozsah vramci nejakeho priestoru, osovo orientovany rozsah.

XI. Kompozice digitálních obrazů

62. Princip klíčování podle intenzity (luma), podle barvy (modré pozadí), rozdílové klíčování. Co je vstupem a výstupem jednotlivých metod.

Klíčování

oddelení popředí od pozadí, vytvoření masky popředí

jiný název vytažení, extrakce masky (pro popředí), nebo (vy)klíčování (t.j. zprůhlednění) pozadí

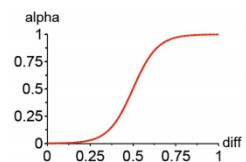
Dobrá maska má $0 < \alpha < 1$ ve stínu, podél hranic objektu a podél hranic stínu.

Postup

-vypočti rozdíl mezi popředím a pozadím

Luma klíčování

Vstup: luma, pripadne funkcia prevodu lumy na alfu => alfa pre matovanie



klíčování pozadí na základě luminance.

Ize použít, pokud má pozadí uniformní luminanci, která se výrazněji od luminance popředí

Modre pozadie:

Vstup: farba popredia, farba pozadia => alfa pre matovanie

Dana farba popredia C_f a pozadia C_k taka ze $C_f = \alpha_o C_o + (1 - \alpha_o) C_k$. Chceme ziskat C_o povodnu farbu popredia podla α_o .

$$R_f = \alpha_o R_o$$

$$G_f = \alpha_o G_o$$

Po upraveni mame rovnice $B_f = \alpha_o B_o + (1 - \alpha_o) B_k$

Pripad ak popredie bez modrej : $\alpha_o = 1 - B_f / B_k$

Pri sedom ci inej farby popredy sa vyber α_o zozlozituje

Problémy obrazu s modrým pozadím

Modre zafarbenie obrazu, polopriehladne materialy.

63. Klíčování obrazu s obecným (přírodním) pozadím. Základní postup, vstupní data, co je výsledkem zpracování?

Pro **obecné řešení** matovacího problému s definovaným barevným pozadím:

– Pořídíme stejný snímek se dvěma různými barvami pozadí. Povodna farba popredia vieme vypočitat z serie rovnic. Naprikad metodou najmensich stvorcov.

Niekedy vhodnejšie **rozdielove maskovanie**.

-spravime snimok pozadia s popredim a bez popredia.

-alfa bude velkosť rozdielu medzi obrazkom s popredim a obrazkom len s pozadim.

Nezname pozadie

vstup: původní kompozice + trimapa (biela-popredie,cierna-pozadie,seda-nevieme)

výstup: alfa maska

Algoritmus doplni v trimape sede oblasti.

Napr: **Knockout-**

Ruzon,Tomasi- klustrovanim farieb

Bayesian- Maximum-a-Posteriori (MAP) Estimation (odhad postupne z okrajov)

XII. Silová zpětná vazba, haptika

64. Uveďte příklady systémů pro aplikaci silové zpětné vazby na jednotlivé části lidského těla.

Haptika-the science of applying touch (tactile) sensation and control to interaction with computer applications.

Tactile Perception

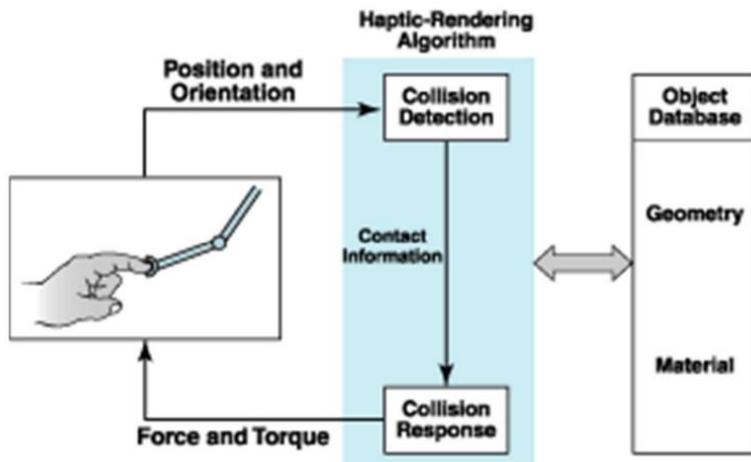
- Provides information about our environment – e.g. hot, cold, smooth, rough,...
- Provides feedback – e.g. when trying to lift an object, press buttons,...

Devices

- tactile devices - finger tips, skin, body
 - haptic gloves –finger phalangea (segments),
sense of grasp
 - force feedback – finger & fingers,
sense of touch
 - force feedback - arm & body movement
- Pins or other mechanical vibrating elements
– Air jets blow to produce a disturbance
– Cushions of air can be inflated or deflated to vary pressure
on skin
– Electrical stimulation - low levels of current provide a
localized tingling sensation
– The Pen Based Force Display (PHANToM)

Magnetic Levitation Haptic Device

65. Popište systém se silovou zpětnou vazbou, vysvětlete, jakou funkci plní jednotlivé části a jaké nároky na výpočetní výkon kladou příslušné výpočetní komponenty.



Uzivatelsky vstup sa zisti kolizia podla modelu, podla modelu a materialu sa vytvorí odozva a ta sa aplikuje späť na uzívateľa.

1KHz Performance Requirement

Basic approach

- Check if point penetrates an object.
- Find closest point on the surface.
- Apply penalty-based force.

The user becomes part of the simulation loop.

- 1KHz is necessary so that the whole system doesn't suffer from disturbing oscillations.

Think of the analogy with numerical integration of a system with spring, mass and damper, where the frequency of the haptic loop sets the integration step.

- The haptic devices run their control loop at 1KHz.
- Consequence: **we are very limited on the amount of computation that we can do.**

Práca v 3d prostredí nie v 2d projekcii na obrazovke. Možnosť oklamat užívateľa vizuálnym vstupom, vyhľadávajúcim haptikov.

Nutnosť korektného zistovania kolizii a nasledného vektoru pre aplikovanie protisily. Používa sa GodObject(najblízší bod na povrchu objektu tak užívateľ vstupí dnu, podľa neho sa vypočítajú protisily), proxy objekty(namiesto bodu vypočítajú súčelovú sílu, force shading (interpolácia smeru protisily po hránach objektov, vyhľadávajúce ho))