KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

Textury
KMI/3DG

Mgr. Markéta Trnečková, Ph.D.

Textury

- **Textura**– popis vlastností povrchu
- Je důležitá pro vnímání struktury objektu
- Dělení:
  - Pravidelná
  - Nepravidelná
- **Texel** – prvek textury
- Často je efektivnější použít jednoduchou geometrii a složitou texturu, než definovat složité geometrické útvary
- Objekty, které jsou v dálce, nebo zobrazeny krátce jsou od složitě definovaných k nerozeznání

Textury

- **Dělení** dle toho, jakou vlastnost povrchu popisují
  - **Barva povrchu** – je určena koeficientem difúzního odrazu, aplikace textury – mapování difúzní složky
  - **Odraz světla** – simuluje změnu zrcadlové složky materiálu, mění se s místem na povrchu
  - **Změna normálového vektoru** – opticky mění tvar povrchu, aniž by měnila jeho tvarů hrbolaté textury
  - **Průhlednost** – její aplikací docílíme dojmu změny geometrie povrchu
  - **Hypertextura** – Určuje optické vlastnosti nad povrchem objektu a hodí se na modelování vlasů, ohně, trávy

Textury

- **Dělení** dle dimenze
  - **Jednorozměrné** – definice opakujících se pravidelných vzorků (např. pruhy – přechod pro chodce), vzorek pro generování přerušovaných křivek a čar, animace deště a jiné
  - **Dvourozměrné** – jsou mapovány na povrch tělesa
  - **Trojrozměrné** – Definují hodnotu textury v prostoru – **objemové textury** (solid texture) – používají se pro simulaci objektů, které vypadají jako vyřezané z jednoho bloku materiálu
  - **Čtyřrozměrné** – animace trojrozměrných

Textury

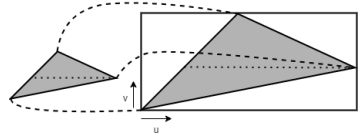
- **Dělení** dle reprezentace
  - **Uložené v tabulce** – 1, 2 nebo 3 rozměrné tabulky
  - **Obrazový soubor** – pole voxelů
  - **Definice procedurou** – procedurální textury

Nanášení textury

- **Nanášení textury** – mapování textury
- Nanesení textury na objekt = „nakreslení“ textury na její povrch
- Textura je to, co je nakresleno na papír, mapování, jak tento papír nalepíme na objekt
- Například dřevěná deska = nakreslíme na ní dřevěný vzor
- Použití takových textur je rychlé, ale výsledky nemusí být hezké
- Pro malé objekty, nebo objekty v pozadí je míra detailu dostatečná, na velkých plochách se budou projevovat chyby ve stínování
- Máme texturu – buď danou obrázkem, nebo vygenerovanou – uloženou ve dvojrozměrné mřížce
  $T_u \times T_v$
  $u \in \{0, 1, \ldots, T_u - 1\}$, $v \in \{0, 1, \ldots, T_v - 1\}$
- Hledáme mapování každého bodu objektu na texturu
- Hledáme transformaci mezi souřadným systémem objektu a souřadným systémem textury
- Body z objektu mapujeme na hodnoty $u \in \{0, \ldots, T_u - 1\}$, $v \in \{0, \ldots, T_v - 1\}$

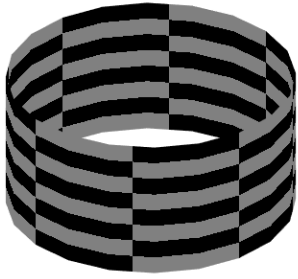
Nanášení textury

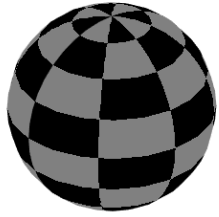
- Každému bodu dopočítáme bod v textuře
- Hodnotu mezi body interpolujeme
- Při vykreslování tělesa známe souřadnice bodu v systému souřadnic objektu $[x, y, z]$ a pro něj nalezneme z transformace souřadnice $[u, v]$ v prostoru textury
- Volba mapovací funkce musí odpovídat tvaru tělesa, na které je textura nanášena



Nanášení textury

- Pro plochy, které lze rozvinout do roviny, je možné takovou funkci najít
- U ploch, které se rozvinout nedají, dojde ke zkreslení textury

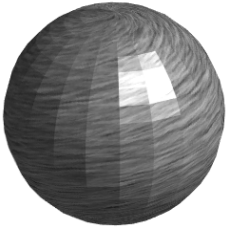




- Pokud je těleso ohraničeno více plochami, je těžké dosáhnout správného navazování textury

Nanášení textury

- Textura ovlivní osvětlovací proces $\rightarrow$ textura nahradí difúzní složku (konstantu) ve výpočtu
- Textura sice změní vzhled, ale ne odlesky (zrcadlovou složku)



Nanášení textury

- Mapování při rozvinutí tělesa do roviny
- Body rozvinutého tělesa $x \in \langle x_{min}, x_{max} \rangle$, $y \in \langle y_{min}, y_{max} \rangle$
- Body textury $u \in \langle u_{min}, u_{max} \rangle$, $v \in \langle v_{min}, v_{max} \rangle$

Příklad

Nalezněte transformaci pro mapování textury pro tělesa, která se dají rozvinout do roviny.

Nanášení textury

Příklad

Nalezněte transformaci pro mapování textury pro tělesa, která se dají rozvinout do roviny.

- $u = \frac{u_{max} - u_{min}}{x_{max} - x_{min}}(x - x_{min})$
  $v = \frac{v_{max} - v_{min}}{y_{max} - y_{min}}(y - y_{min})$
- u a v nemusí být celá čísla
- Při zaokrouhlení by vznikly artefakty
- Používá se bilineární interpolace
- Tato technika roztáhne texturu, pokud je objekt větší než textura
- Pokud je textura větší než objekt, zmenší ji

Nanášení textury

Příklad

Pro $u = 3.4$ a $v = 5.8$ nalezněte hodnotu textury, pokud známe $T(3,5)$, $T(3,6)$, $T(4,5)$ a $T(4,6)$.

Nanášení textury

Příklad

Pro $u = 3.4$ a $v = 5.8$ nalezněte hodnotu textury, pokud známe $T(3,5)$, $T(3,6)$, $T(4,5)$ a $T(4,6)$.

$T_1 = 0.6 \cdot T(3,5) + 0.4 \cdot T(4,5)$
$T_2 = 0.6 \cdot T(3,6) + 0.4 \cdot T(4,6)$
$T = 0.2 \cdot T_1 + 0.8 \cdot T_2$

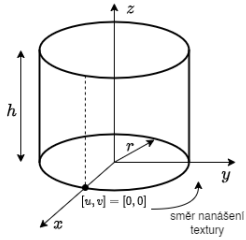
Inverzní mapování na válec

- **Válec** umístíme tak, aby jeho osa byla rovnoběžná s osou z a střed dolní podstavy ležel v počátku
- Mapovat budeme pouze na válcovou plochu tak, aby textura definovaná v oboru $\langle 0, 1\rangle \times \langle 0, 1\rangle$ pokrývala celou plochu
- Vyjdeme z popisu válcové plochy – r poloměr, h výška
- Cylindrické souřadnice $[x, y] = [r \cdot \cos\alpha, r \cdot \sin\alpha]$
- Mapovací funkce $M(x, y, z)$:

$$u = \begin{cases} \frac{1}{2\pi} acos\frac{x}{r} & y \leq 0 \\ 1 - \frac{1}{2\pi} acos\frac{x}{r} & y > 0 \end{cases}$$

$$v = \frac{z}{h}$$

- *acos* vrací hodnoty z intervalu $\langle 0, \pi\rangle$



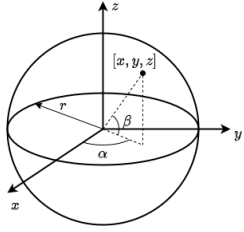
Inverzní mapování na kouli

- **Koule** – postupujeme analogicky
- Kouli o poloměru r umístíme tak, aby její střed byl v počátku souřadnic
- Sférické souřadnice bodů na ploše $[x, y, z] = [r \cdot \cos\alpha \cdot \cos\beta, r \cdot \sin\alpha \cdot \cos\beta, r \cdot \sin\beta]$
- $\alpha \in \langle 0, 2\pi \rangle$, $\beta \in \langle -\pi/2, \pi/2 \rangle$
- Mapovací funkce $M(x, y, z)$:

$$u = \begin{cases} \frac{1}{2\pi} acos \frac{x}{\sqrt{x^2+y^2}} & y \leq 0 \\ 1 - \frac{1}{2\pi} acos \frac{x}{\sqrt{x^2+y^2}} & y > 0 \end{cases}$$

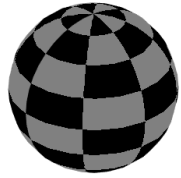
$$v = 0.5 + \frac{1}{\pi} asin \frac{z}{r}$$

- *acos* vrací hodnoty z intervalu $\langle 0, \pi \rangle$, *asin* vrací hodnoty z intervalu $\langle -\pi/2, \pi/2 \rangle$



Inverzní mapování na kouli

- Textura se směrem k pólům deformuje a v nich degraduje na jeden bod
- Tyto případy odpovídají dělení 0 ve vzorci



Mapování textury

- Při mapování může dojít k neuniformní změně měřítka textury (deformuje se)
- Texturu můžeme
  - Zvětšit a použít jen její část
  - Texturu opakovat – texture repeat, tiling
  - Texturu naneseme tak, jak je a pak už se dál nenanáší – texture clamping

Příklad

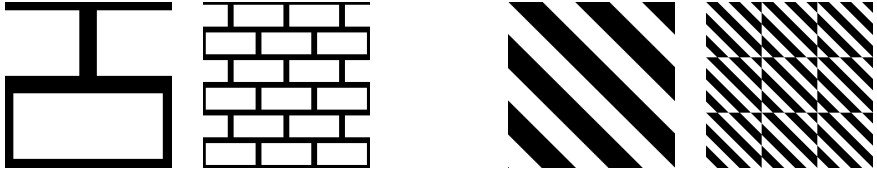
Jak bychom upravili předchozí postupy, abychom texturu nanášeli opakovaně?

Mapování textury

Příklad

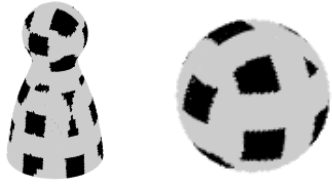
Jak bychom upravili předchozí postupy, abychom texturu nanášeli opakovaně?

- Pomocí funkce modulo
  $u = (x - x_{min})\%\, T_u$
  $v = (y - y_{min})\%\, T_v$
- Aby fungovalo opakované nanášení textury, musí být textura v obou směrech opakovatelná



3D textury

- Při „polepování“ tělesa texturou je těžké zachovat navazování textury i když je správně definovaná
- U těles, která není možné rozvinout do plochy je použití textur složité
- Alternativou k 2D texturám jsou 3D textury – prostoroví textury
- Objekt je vložen do textury $\rightarrow$ jako by byl z textury vyřezán
- Pokud je textura menší než objekt – opakování textury
- Mapování je přímočaré $x \rightarrow u$, $y \rightarrow v$, $z \rightarrow w$
- Velké paměťové nároky na texturu – textury se definují v menších mřížkách a hodnoty se interpolují, případně se používají funkční textury



Mapování prostředí

- **Pohledově závislé mapování**, environment mapping
- řešíme případy, kdy renderujeme předměty s povrchem, který odráží prostředí
- Ostatní objekty ve scéně ovlivní vzhled renderovaného předmětu
- To, co ovlivní vzhled objektu v určitém místě můžeme najít sledováním odrazové složky z povrchu objektu
  $R = 2 \cdot N(N \cdot V) - V$
- Metoda sledování paprsku – časově náročné
- Bude-li se pohybovat objekt, pozorovatel nebo se bude měnit okolí objektu, bude se měnit i textura na povrchu

Mapování prostředí

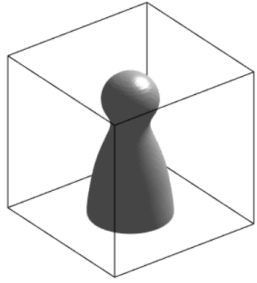
Zjednodušení

- **Myšlenka** – na kouli, kterou obklopíme vykreslovaný objekt promítneme okolí (scénu)
- Koule má střed ve stejném bodě jako je střed objektu
- Projekce (i renderování) na kouli používá střed koule jako střed projekce a polohu pozorovatele $\rightarrow$ **sférický pohled na scénu** (reflection map)
- Tak, jak se scéna mapuje na kouli, tak se pak mapuje na povrch objektu
- Kulová sférická reflection map může být uložena ve 2D mřížce $\rightarrow$ ve sférických souřadnicích
  $u = \frac{1}{2}(1 + \frac{1}{\pi} atan(\frac{R_x}{R_y})$
  $v = \frac{R_z+1}{2}$
- *atan* vrací hodnoty z intervalu $\langle -\pi, \pi \rangle$
- Problémy nastávají v pólech – malé změny v odrazovém vektoru vedou k velkým změnám v u a v

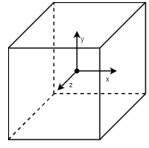
Mapování prostředí

Zjednodušení

- **Alternativa** – místo koule použijeme krychli
- Střed krychle leží ve středu objektu
- Střed promítání a pozorovatele jsou v tomto středu
- Máme 6 rovin, na které promítáme



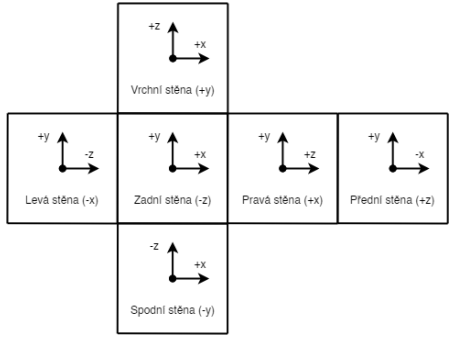
- Krychli orientujeme tak, aby byly hrany rovnoběžné s osami souřadného systému



Mapování prostředí

Zjednodušení

- Když krychli rozložíme, každý čtverec má vlastní u, v souřadnice (u horizontální – doprava, v vertikální – nahoru)
- Střed každého čtverce je v místě, kde jedna ze souřadných os (v kladném, nebo záporném směru) opouští krychli
- To, kterou ze stěn prochází paprsek, je snadné určit – prozkoumáme normalizovaný vektor odrazu a největší absolutní souřadnice určuje, kterou stěnou prochází



Příklad

Které stěny protínají následující vektory?

- $R = (0.4, 0.7, 0.59)$
- $R = (-0.75, 0.25, 0.43)$

Mapování prostředí

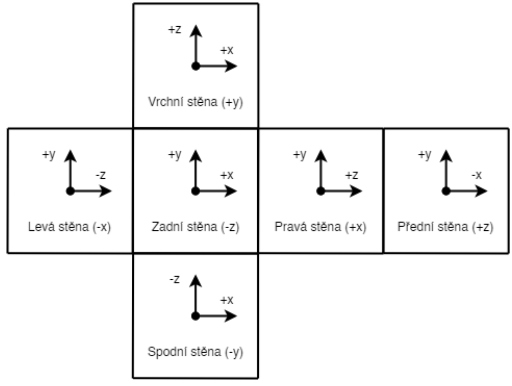
Zjednodušení

Příklad

Které stěny protínají následující vektory?

- $R = (0.4, 0.7, 0.59)$
- $R = (-0.75, 0.25, 0.43)$

- Vrchní stěnu $(+y)$
- Levou stěnu $(-x)$



Mapování prostředí

Zjednodušení

- Jakmile máme určenou stěnu, určíme u a v ($\in \langle 0, 1 \rangle$)
  $u = \frac{a+c}{2c}$
  $v = \frac{b+c}{2c}$
  a . . . velikost souřadnice v horizontálním směru
  b . . . velikost souřadnice ve vertikálním směru
  c . . . velikost souřadnice vybrané stěny

Příklad

Pro vektor $R = (0.4, 0.7, 0.59)$ spočítejte souřadnice u a v ve stěně, kterou prochází.

Mapování prostředí

Zjednodušení

Příklad

Pro vektory $R = (0.4, 0.7, 0.59)$ a $R = (-0.75, 0.25, 0.43)$ spočítejte souřadnice u a v ve stěně, kterou prochází.

$R = (0.4, 0.7, 0.59)$

- Vrchní stěna
- $a - x$ souřadnice
  $b - z$ souřadnice
  $c - y$ souřadnice
  $u = \frac{0.4+0.7}{1.4}$
  $v = \frac{0.59+0.7}{1.4}$

$R = (-0.75, 0.25, 0.43)$

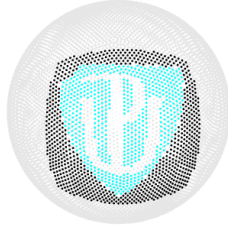
- Levá stěna
- $a - -z$ souřadnice
  $b - y$ souřadnice
  $c - -x$ souřadnice
  $u = \frac{-0.43+0.75}{1.54}$
  $v = \frac{0.25+0.75}{1.5}$

Mapování prostředí

Zjednodušení

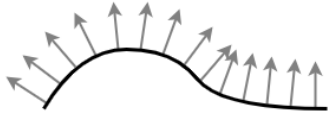
- Díky použití mapy není potřeba odraz počítat při každém pohybu pozorovatele/kamery
- Pouze při pohybu objektů ve scéně
- Při otáčení renderovaného objektu ve scéně není potřeba přepočítávat mapu, jen texturu

- To, co se odráží není úplně to, co by se ve skutečnosti od objektu odráželo
- Promítáme scénu z 1 bodu (střed objektu) ne z povrchových bodů
- Čím je povrchový bod dál od středu, tím bude odraz méně reálný
- Při mapování se mohou objevit nespojitosti v hranách krychle – i přes to je nejčastěji používaná



Hrbolaté textury

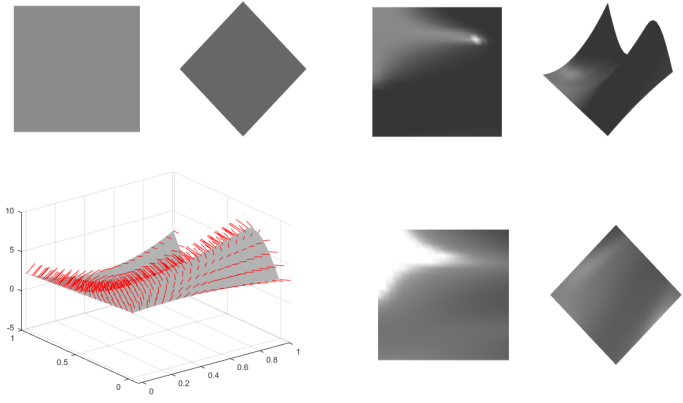
- **Hrbolaté textury** – bump mapping
- Textury používáme k modifikaci zrcadlové (specular) složky osvětlení
- **Myšlenka** – povrchy s hrbolky se od těch rovinných liší tím, kde se více odráží světlo, kterým směrem se odráží světlo
- Na povrchu s hrbolky jdou normálové vektory různými směry



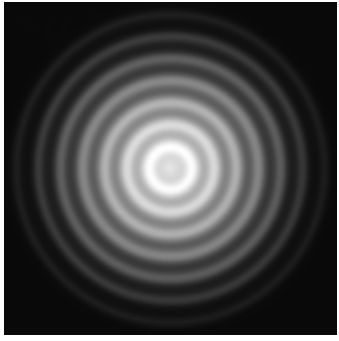
- Abychom toto chování simulovali, změníme normálové vektory

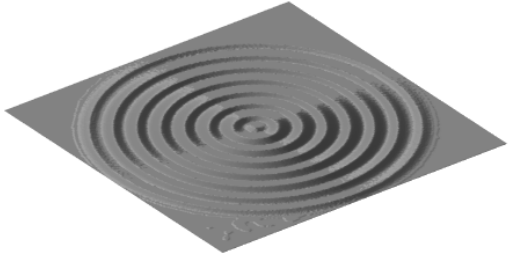
Hrbolaté textury

- Když stejné směrové vektory, které bychom získali z „hrbolatého“ povrchu použijeme pro plochu, bude plocha vypadat hrbolatě i když není



Hrbolaté textury





Hrbolaté textury

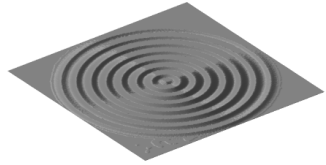
- Při použití kreslených textur se modifikuje difúzní složka při výpočtu osvětlení
- Hodnota hrbolaté textury mění směr normálového vektoru $\rightarrow$ změní se jak difúzní, tak zrcadlová (specular) složka
- Hrbolatá textura – **bump map** – N_B
  $R_B = 2 \cdot N_B(N_B \cdot V) - V$
  $C_{\{r,g,b\}} = k_{a\{r,g,b\}} + I_L \cdot [k_{d\{r,g,b\}} L \cdot N_B + k_s(R_B \cdot V)^n]$
- Bump funkce je závislá na gradientu bump mapy v místě $[u, v]$
- Změna v tomto bodě popisuje změnu normály povrchu
  $BM(u+1, v) - BM(u-1, v)$ ... ve směru u
  $BM(u, v+1) - BM(u, v-1)$ ... ve směru v

Hrbolaté textury

- Nalezneme dva tečné vektory na pozici $[u, v]$ povrchu S_u a S_v
- Ty jsou na sebe kolmé – vektorovým součinem získáme normálový vektor v tomto bodě
- Souřadnice $[x, y]$ převedeme na $[u, v]$ (souřadnice textury)
- Spočítáme texturové gradienty B_u a B_v a ty použijeme spolu s S_u a S_v k výpočtu modifikované normály
  $N' = N + \frac{B_u(N \times S_v) - B_v(N \times S_u)}{||N||}$
  $N \times S_u$ a $N \times S_v$ slouží k výpočtu tečných vektorů, které jsou na sebe kolmé. To ale splňují i samotné S_u a S_v
  $N' = N + \frac{B_u \cdot S_v}{||S_v||} - \frac{B_u \cdot S_u}{||S_u||}$

Hrbolaté textury

- Nedokážeme touto technikou simulovat to, že silně zvlněné povrchy by stínili sami sebe



Procedurální textury

- Doposud byly textury uloženy v nějaké další struktuře – paměťová náročnost
- **Procedurální textury** – programy, které vrací hodnoty pro jednotlivé prostorové body
- Výpočet hodnoty je výpočetně náročnější, než přístup do pole
- **Implementace textury do osvětlení** – přímočaré
- Funkce `textura(x,y,z)` – vrátí hodnotu texelu pro bod
  $k_d =$`textura(x,y,z)`
  $C_{\{r,g,b\}} = k_{a\{r,g,b\}} + I_L \cdot [k_{d\{r,g,b\}} L \cdot N + k_s (R \cdot V)^n]$
- Můžeme je ale použít i jako bump funkce