Bump-mapping

Jiří Dvořák, 2001

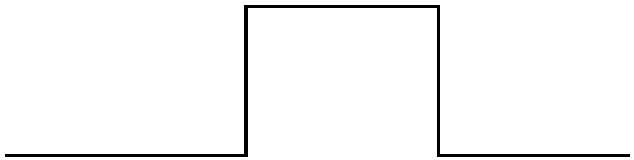
Připomenutí

- Je to metoda, jak vytvořit dojem složité povrchové struktury.

Emboss bump-mapping

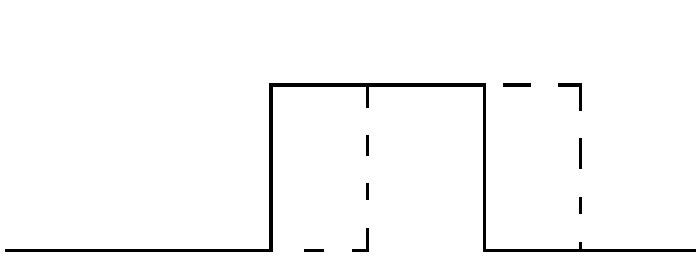
- Podporován velkou většinou karet
  - Pokud je v případě starších karet v reklamním letáku uvedeno, že podporují bump-mapping, tak je to většinou tento.
  - 3DLabs Permedia 3 má přímo speciální jednotku
- Pracuje s diffusní složkou
- Jedná se o trik

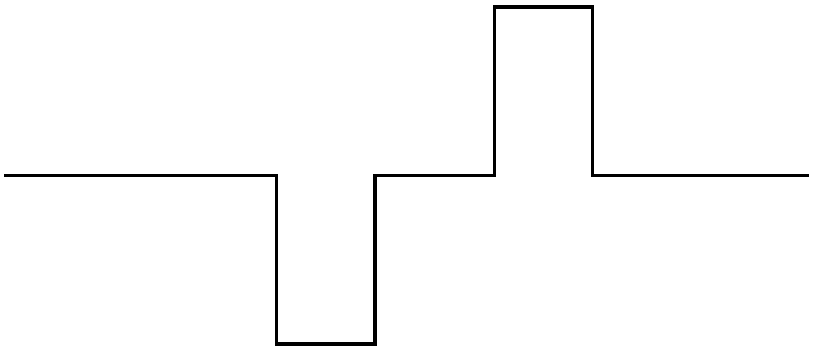
Základní myšlenka



Chceme dosáhnout toho, aby byl výstupek na straně přivrácené ke světlu ozářen a na opačné ve „stínu“

Vytvoříme jeho, vhodně posunutou, kopii a od ní „odečteme“ původní hrbolek



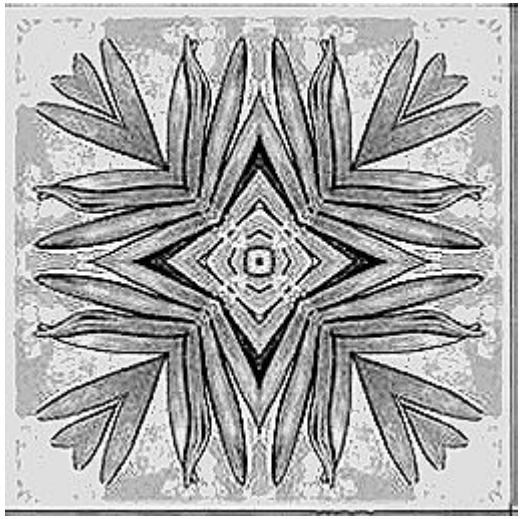


Na základě výsledné „funkce“ změníme intensitu barvy pixelu.

Realizace

Textura povrchu

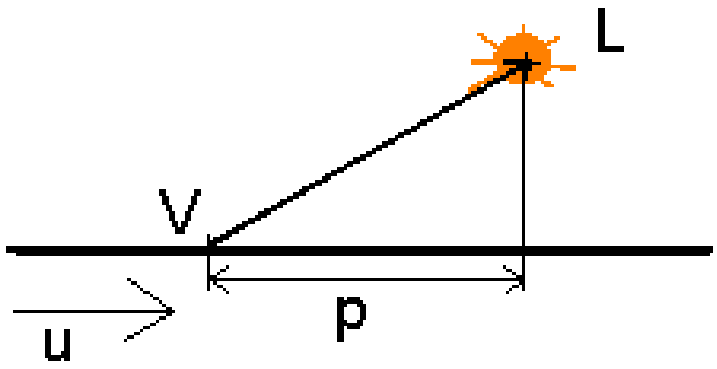
Textura výšek

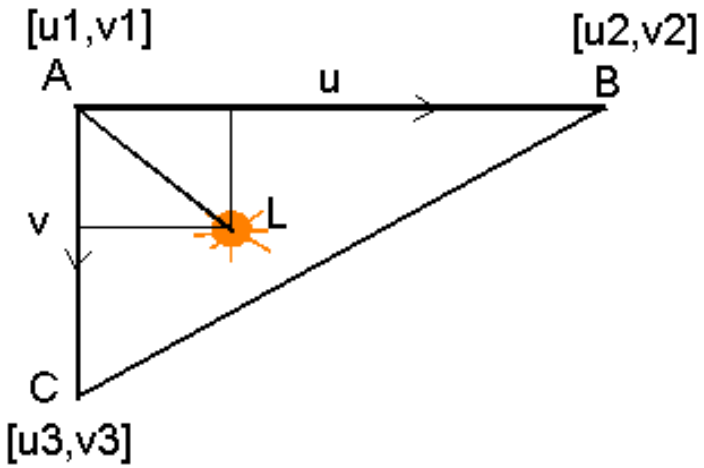


Invertovaná textura výšek

0.0 - největší díra
0.5 - rovina plochy
1.0 - nejvyšší výstupek

Realizace - pokračování





- Pro každý vrchol V spočítáme vektor L-V
- Tento vektor znormalizujeme
- Zjistíme velikost jeho složek du a dv
- Jako texturové souřadnice pro invertovanou texturu použijeme původní souřadnice [u1,v1] a pro neinvertovanou texturu [u1+du*Scale,v1+dv*Scale], kde Scale je <<1 a určuje velikost efektu.

Vliv koeficientu scale

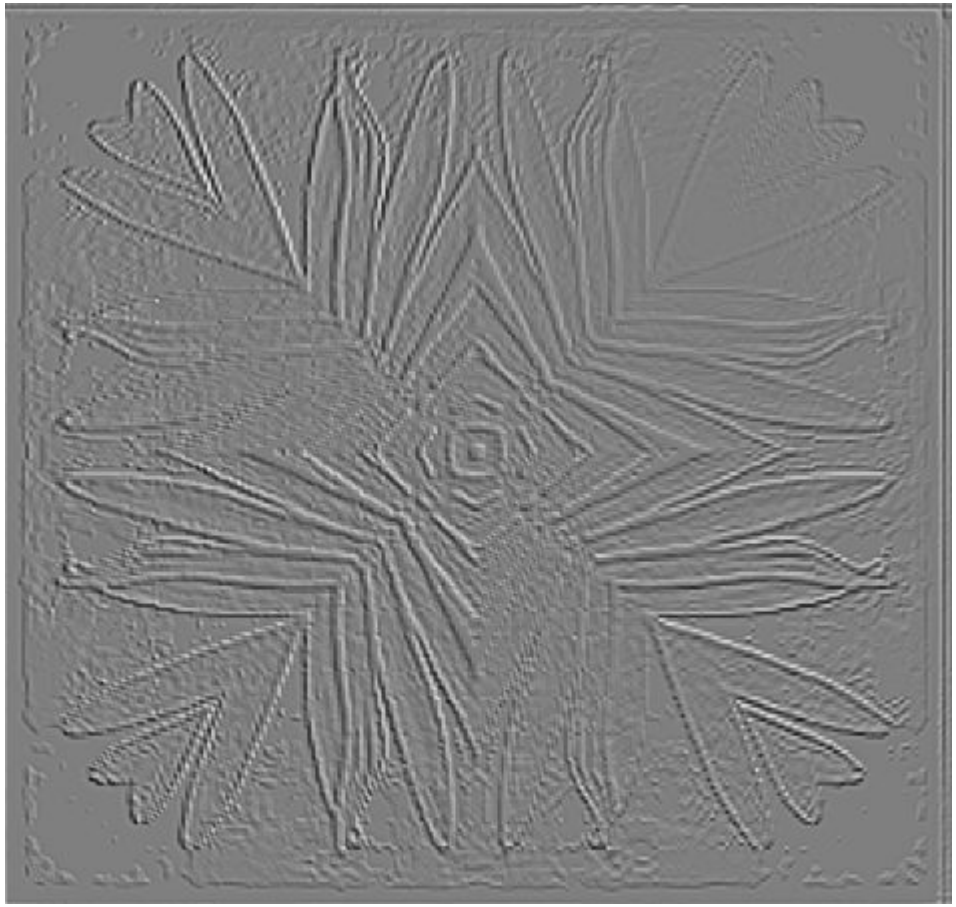
0.002

0.006

Tvorba bump efektu

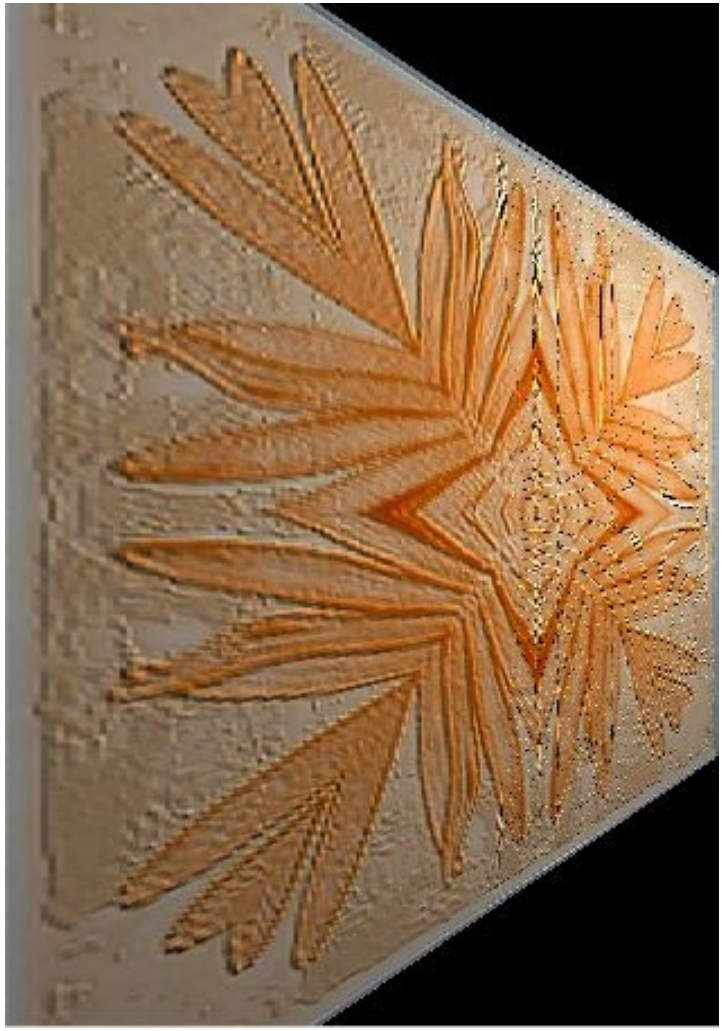
- Různé možnosti kombinace v závislosti na schopnostech HW.
  - Pomocí ADD
    - Lze použít i na HW bez multitexturingu
    - Je třeba použít dvě bump textury s polovičními hodnotami.
  - Pomocí ADDSIGNED
    - Vyžaduje multitexturing. Pokud HW podporuje invertováni barev, tak je třeba jen jedna textura.

Výsledná bumpmapa



Po aplikování textury povrchu

Nedostatky



- Citlivé na deformace
- Pro body ležící „pod světlem“ se efekt ztrácí
- Na křivých plochách je třeba jemné rozdělení.

Environment **M**apped **B**ump **M**apping

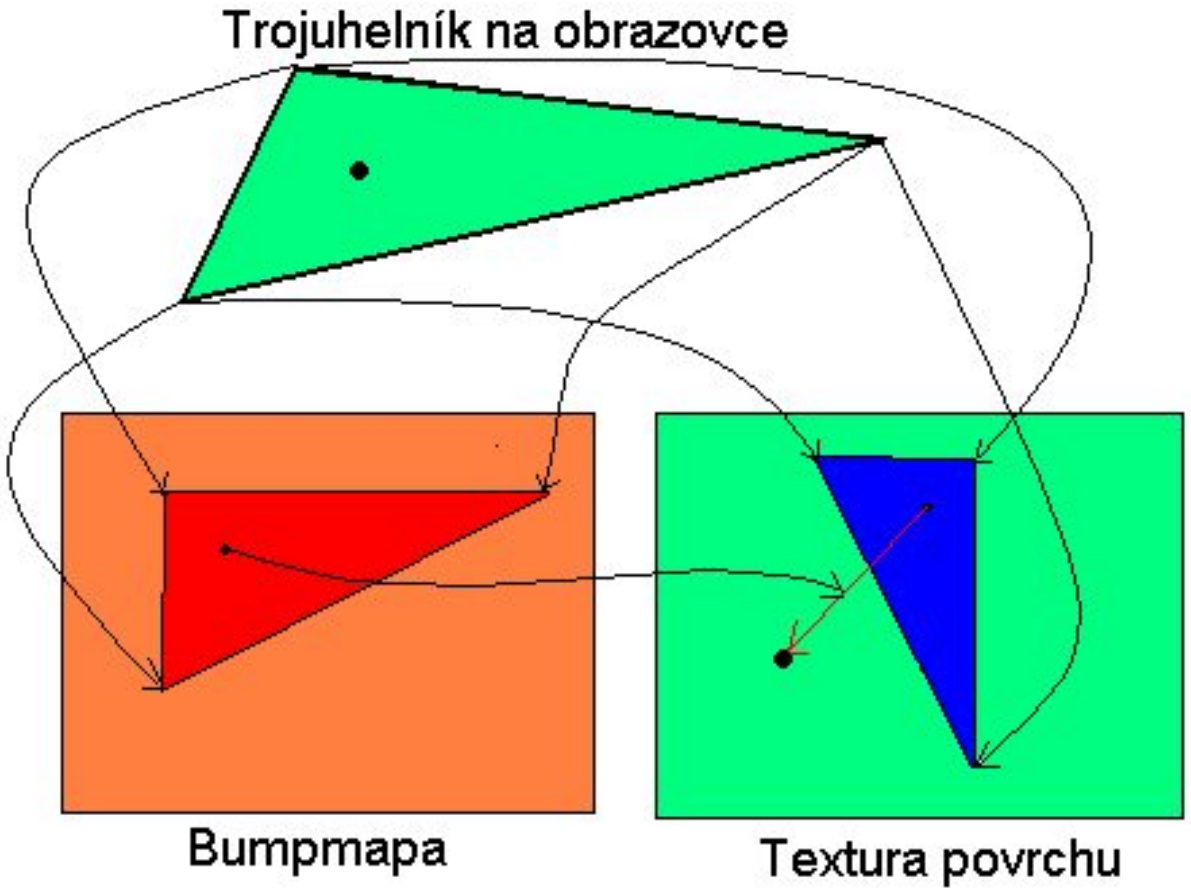
- Metoda pro simulaci nerovností na „zrcadlovém“ povrchu (voda, leštěné kovy,…), ale dá se použít i pro jiné efekty.
- Vyžaduje speciální HW podporu
  - ATI Radeon
  - Matrox G400
  - Permedia 3
  - GeForce 3

Příklady


0.91 fps (400x300x32)
REF (simulated pure hw vp)


0.95 fps (400x300x32)
REF (simulated hw vp)

Princip



Další vlastnosti

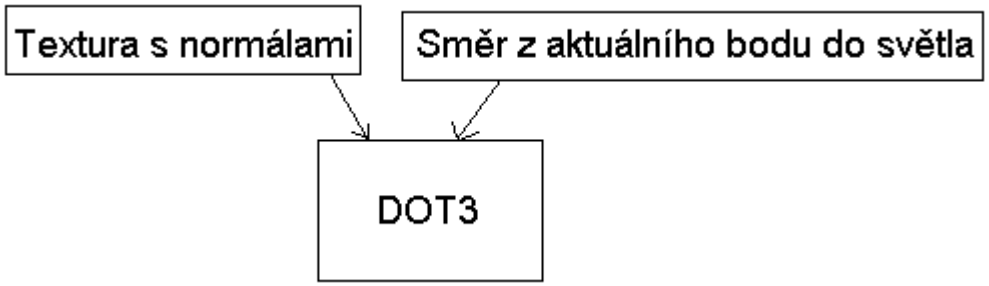
- Hodnotu přečtenou z bump textury lze, před použitím, transformovat jako vektor pomocí matice 2x2.
  - Lze využít pro zvětšení posunu či pro animaci
- Bump textura může navíc obsahovat informaci o jasu, kterou se vynásobí hodnota přečtená z textury povrchu.

DOT3

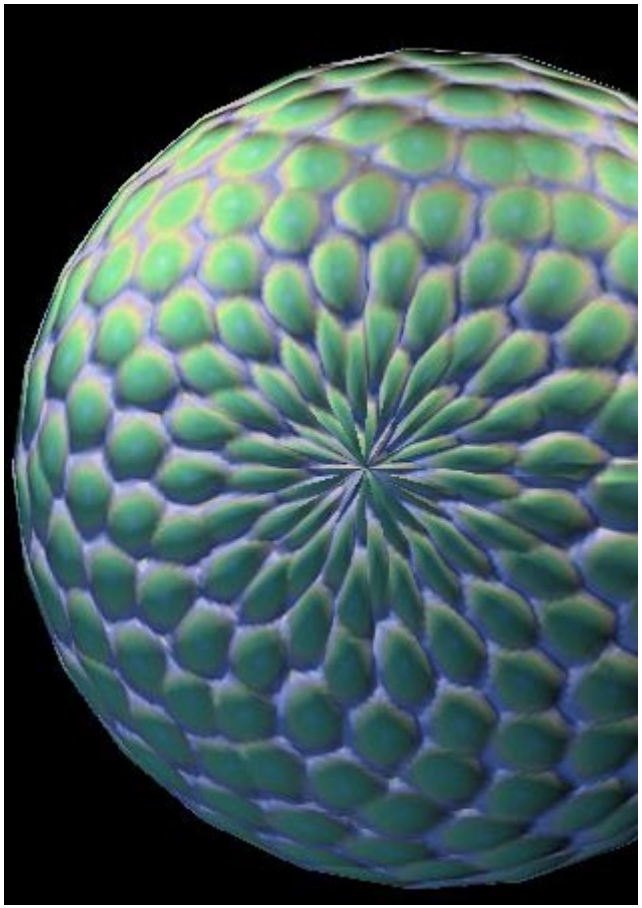
- Umožňuje provádět per pixel stínováni
- Vyžaduje speciální HW podporu
  - NVIDIA GeForce 1,2,3
  - ATI Radeon
  - VideoLogic PowerVR-250
  - Permedia 3

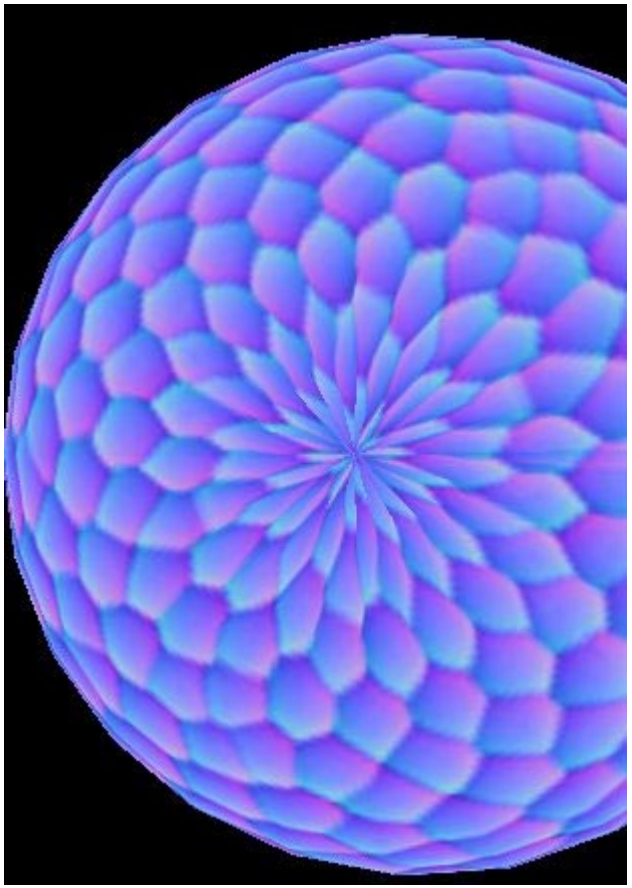
Základní myšlenka

- Barvu chápeme jako třísložkový vektor.
  - V[0]=(R-127.5)/127.5
  - V[1]=(G-127.5)/127.5
  - V[2]=(B-127.5)/127.5
- Existuje speciální operace (DOTPRODUCT3), která umí spočítat skalární součin dvou vektorů reprezentovaných pomocí barev.
- Na objekt budeme nanášet speciální texturu, do které zakódujeme směr normály v jednotlivých bodech povrchu a pomocí této operace ji skombinujeme s vektorem směřujícím z daného bodu povrchu do světla (v případě, že počítáme diffusní složku osvětlení)



Příklad textury normál

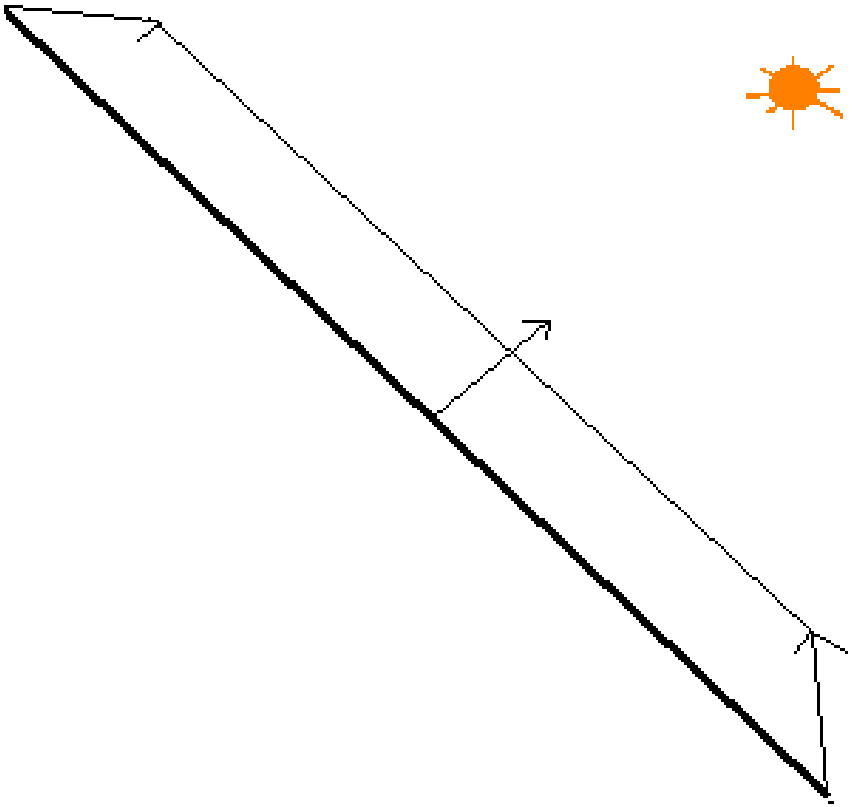




Kde lze uložit směr do světla

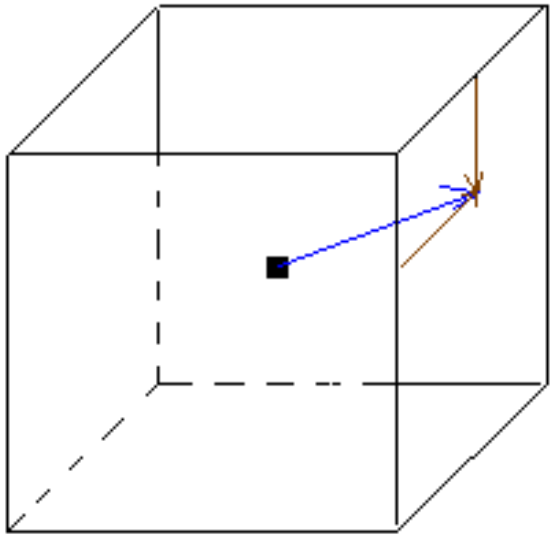
- V konstantním registru
  - Stejný všude na povrchu objektu
  - Lze použít pro diffusní osvětlení od směrových či hodně vzdálených bodových zdrojů.
- V barvě bodu
  - Lze použít i pro bodové zdroje
  - Nepotřebujeme další texturu
- V texturové souřadnici
  - Vyžaduje další texturu
  - Umožňuje HW normalizaci normály

Problém s uložením v barvě bodu



Protože se mezi hodnotami ve vrcholech lineárně interpoluje, může dojít k tomu, že vektor, zakódovaný do barvy, přestane mít jednotkovou velikost.

Řešení pomocí texturových souřadnic a „Cube texture“

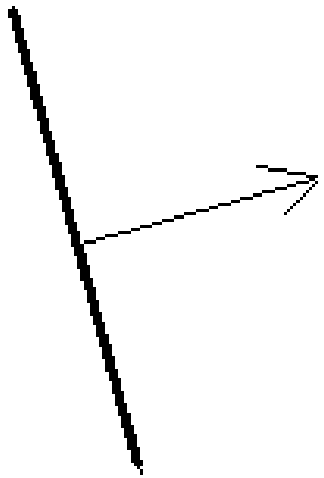


- Cube texture (dále jen CT) je speciální druh textury, která vytváří kolem texturovaného objektu pomyslnou krychli.
- Adresuje se pomocí tří souřadnic, které reprezentují vektor směřující ze středu krychle. Jako barva se použije barva toho bodu na povrchu krychle, do kterého se tento vektor trefil.

Využití CT pro normalizaci

- Vytvoříme jednu CT a naplníme ji tak, aby bod na adrese [x,y,z] měl barvu, která odpovídá zakódovanému vektoru, který je normalizací [x,y,z]
- V každém bodě spočítáme vektor ke světlu a uložíme ho do jeho texturových souřadnic.
- Tyto souřadnice použijeme pro adresaci v CT a získanou hodnotu využijeme v operaci DOTPRODUCT3

Problém s rotací objektu

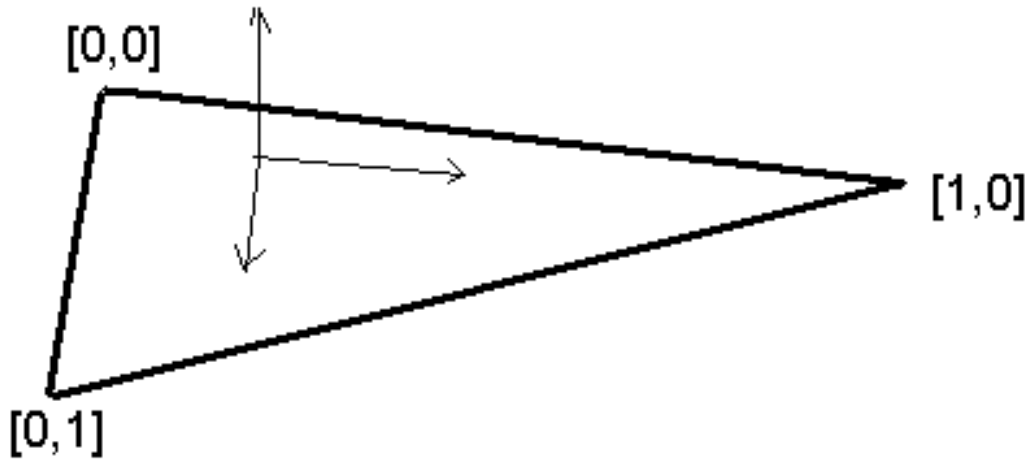


Před rotací

Po rotaci

„Tangent space“

- Budeme do něj transformovat pozici světla
- Z-složka směřuje ve směru normály plochy=> nevychýlený vektor v textuře normál bude mít souřadnice (0,0,1)
- X a Y-složky ve směru rostoucích texturových souřadnic



Poslední novinka

Literatura

- Jeff Molofee's OpenGL Windows Tutorial
  - http://nehe.gamedev.net/tutorials/lesson22.asp
- DX SDK
- Nvidia
  - http://developer.nvidia.com/view.asp?PAGE=nvsdk
- http://www.realtimerendering.com/