

**Textúry**

# Motivácia

- Bez textúr by nám vo filmoch behali len „polygonálne“ modely
- Bez textúr by boli počítačové hry len grafika s holými stenami

# Definície pojmov

- Textúrny priestor (texture space/domain)
- Mapovanie textúr  
textúrny priestor → priestor objektu
- Multitextúry
  - použitie viacerých textúr naraz
- Hypertextúry
  - zmena optických vlastností nad povrchom objektu
- Shader
  - vstup:  $(x,y,z)$ ,  $(u,v)$  súradnice objektu, textúry
  - výstup: farba, ...

# Dimenzia textúr

- Príklady dimenzie textúr
  - textúra je diskretný zoskenovaný/vytvorený obraz
  - dimenzia je určená podľa toho čo určuje
  - podľa dimenzie priestoru v „doméne“:
    - 1D Parameter úsečky, výška, teplota, vzorkované čiary, ...
    - 2D Súradnice bodu plochy  $(u, v)$ , klasická 2D textúra
    - 3D Bod v priestore  $(x, y, z)$ , simulácia vyrezania objektov
    - 4D Animácia, hypertextúra
  - podľa dimenzie priestoru v obraze:
    - 1D hodnota odrazu  $(k_d)$ , intenzita/jas
    - 2D Intenzita a priesvitnosť
    - 3D Farba bodu  $(r, g, b)$ , normála, vektor odrazu
    - 4D Farba a priesvitnosť bodu  $(r, g, b, \alpha)$

# Textúry

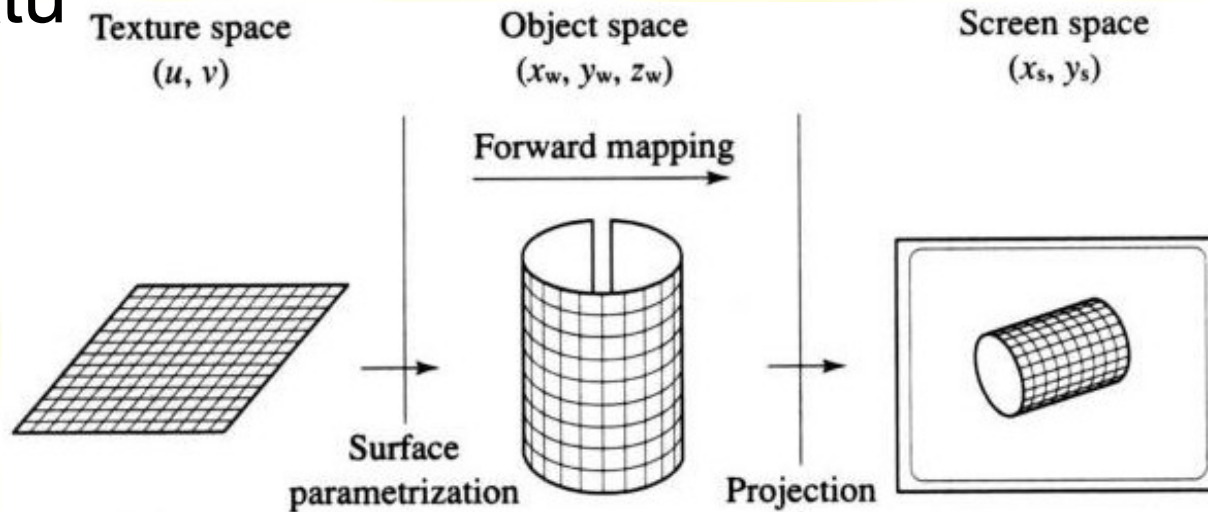
- Zmena vlastností / charakteru povrchu objektu:
  - farba objektu (texture map)
  - lesklosť objektu (diffuse + specular = light map)
  - normála / povrch objektu (bump map, relief map = self-occlusion, self-shadowing, parallax)
  - priehľadnosť (alpha map)
  - tvar objektu (displacement map)
- Vplyv okolia na objekt:
  - tiene (shadow map)
  - odraz okolia (environment map, reflection map)

# Premietanie textúr

- 2D textúra → 3D objekt (najčastejšie používané, tzv. *uv* mapovanie)
- 3D textúra → 3D objekt
- Iné, 4D textúra
  
- Dopredné (forward) mapovanie
- Spätné (inverse) mapovanie

# Dopredné mapovanie

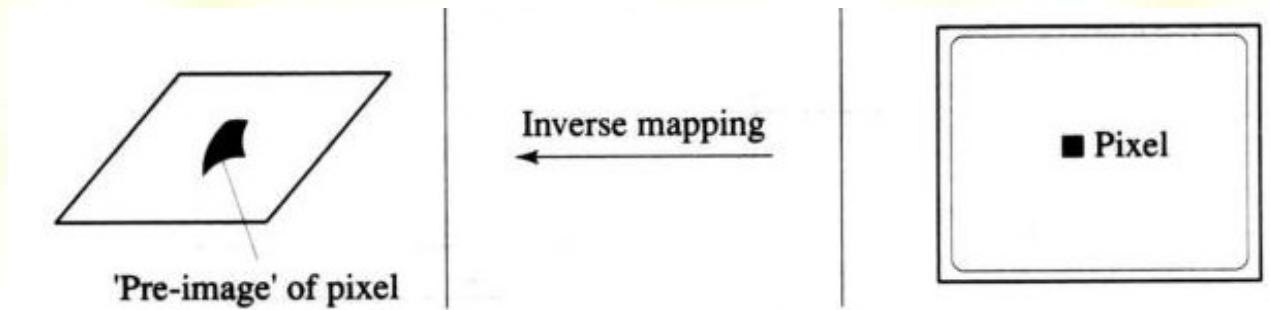
- Bod z textúry (texel)  $(u, v)$  sa premietaním zobrazí na  $(x, y, z)$  na povrchu/polygóne 3D objektu



- Uniformované vzorkovanie textúry nezabezpečí uniformované vzorkovanie v priestore obrazu (screen space)

# Spätné mapovanie

- Pre každý bod  $(x,y,z)$  na povrchu/polygóne 3D objekt sa pomocou inverzného zobrazenia nájde príslušný bod  $(u,v)$  textúry

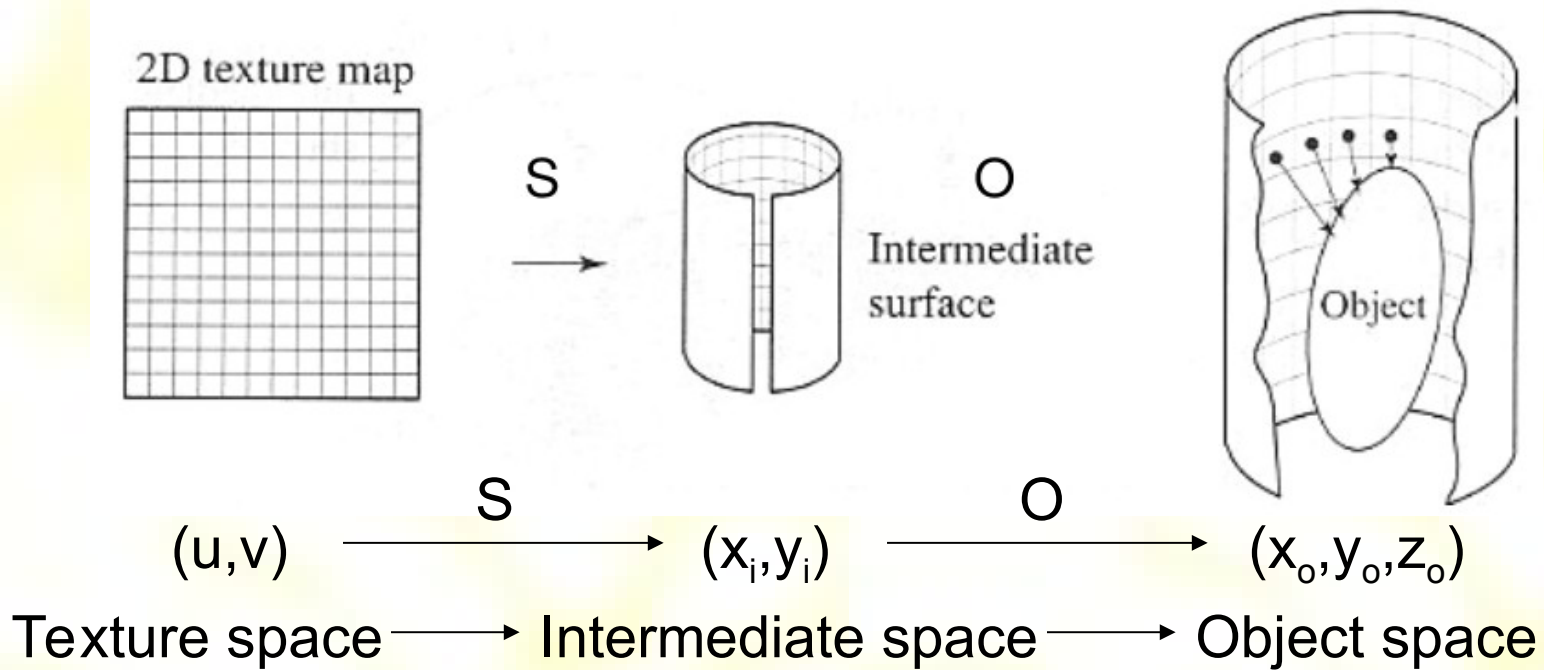


- Pre každý polygón 3D objektu sú v jeho vrcholoch zapamätané texturové súradnice



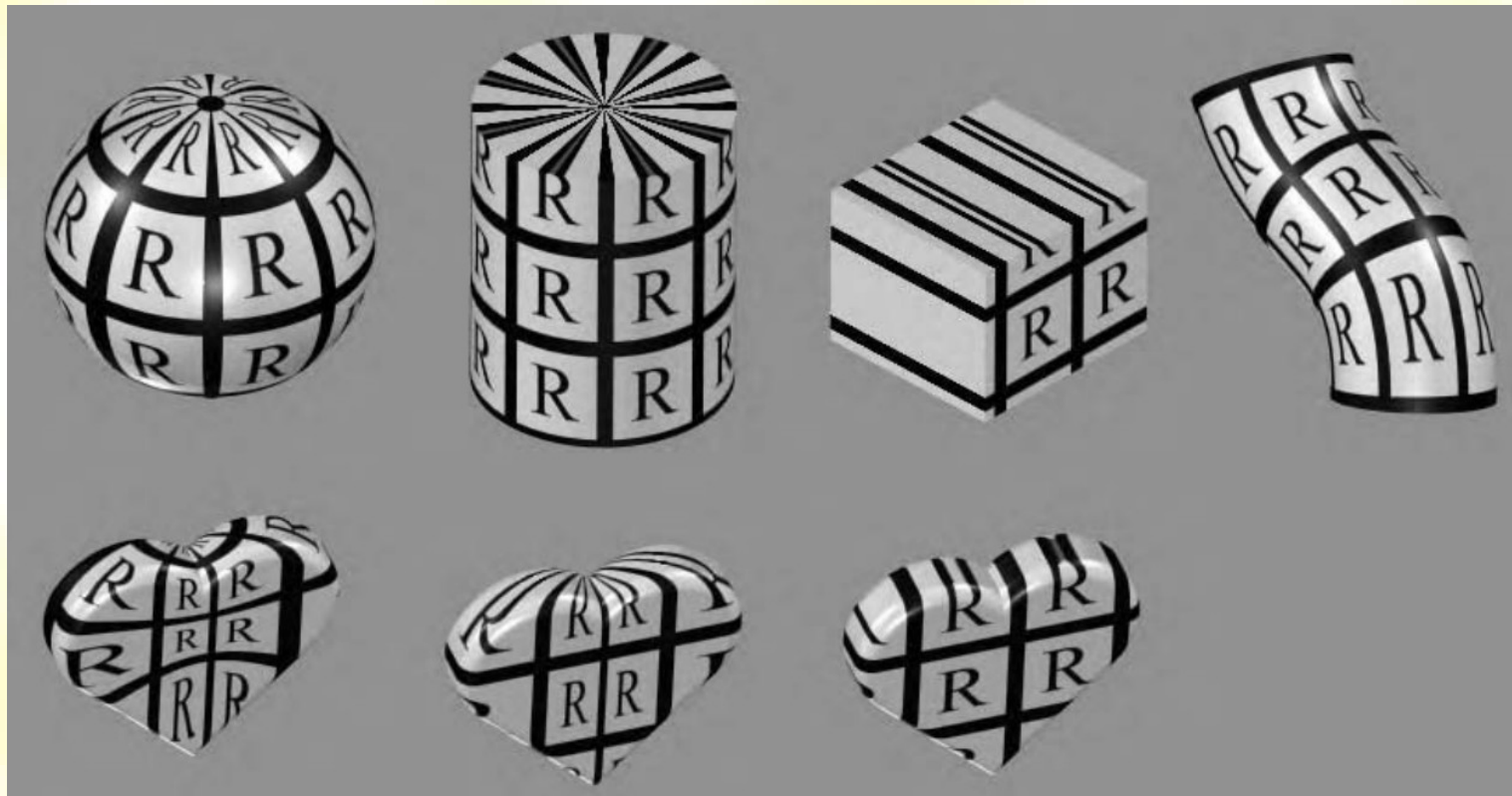
# Dvojfázová projekcia (zobrazenie)

- Projekcia = S mapovanie, O mapovanie



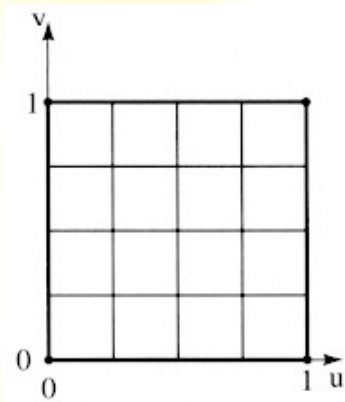
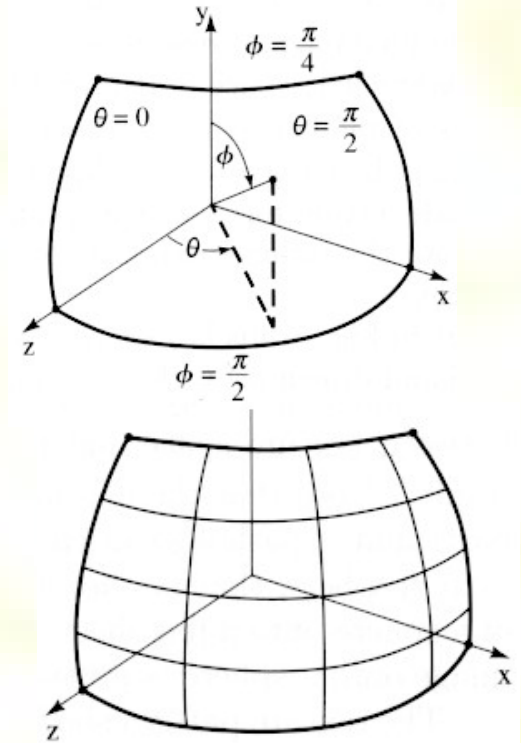
- Jednoduché projekcie:
  - Planárna
  - Kubická, spojitá kubická
  - Sféricá
  - Cylindrická

# Typy projekcí - příklad



# Sférická projekcia – príklad

$$\begin{aligned} x(\theta, \phi) &= \sin \theta \sin \phi \\ y(\theta, \phi) &= \cos \phi & 0 \leq \theta \leq \pi/2 \\ z(\theta, \phi) &= \cos \theta \sin \phi & \pi/4 \leq \phi \leq \pi/2 \end{aligned}$$



$$\begin{aligned} u=0, v=0 &\Leftrightarrow \theta=0, \quad \phi=\pi/2 \\ u=1, v=0 &\Leftrightarrow \theta=\pi/2, \quad \phi=\pi/2 \\ u=0, v=1 &\Leftrightarrow \theta=0, \quad \phi=\pi/4 \\ u=1, v=1 &\Leftrightarrow \theta=\pi/2, \quad \phi=\pi/4 \end{aligned}$$

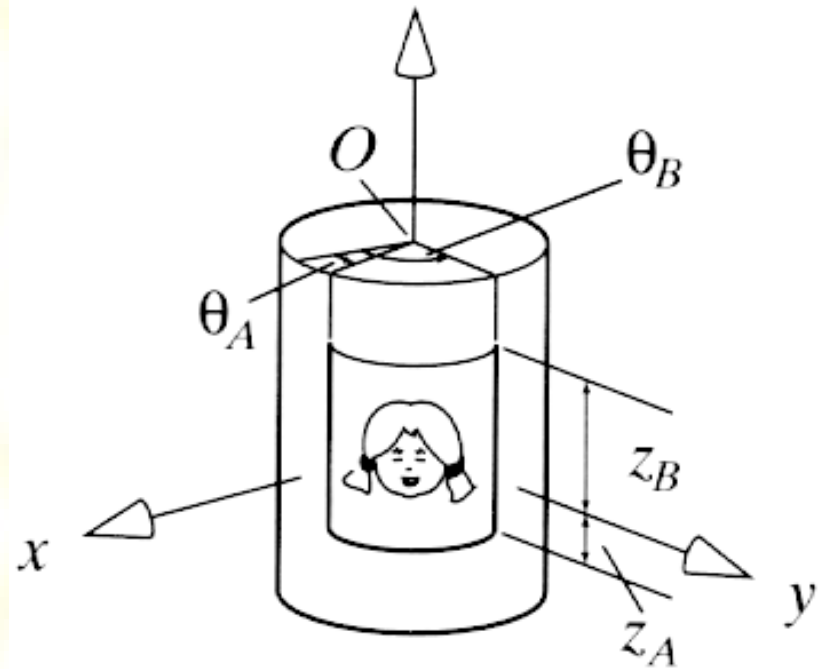
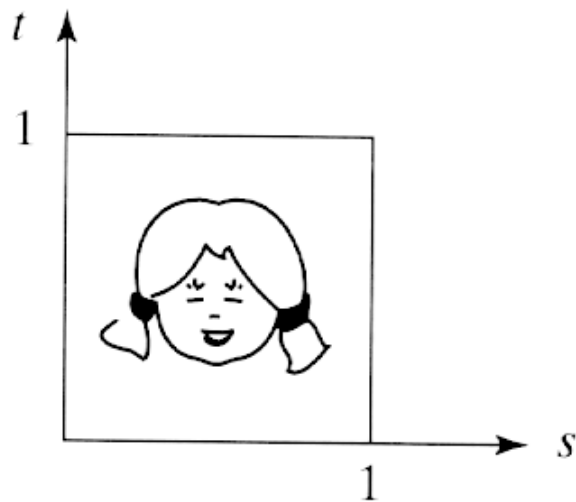
$$\theta = \arctan\left(\frac{y}{x}\right)$$

$$u = \frac{\theta}{\pi/2}$$

$$\phi = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right)$$

$$v = 2 - \frac{\phi}{\pi/4}$$

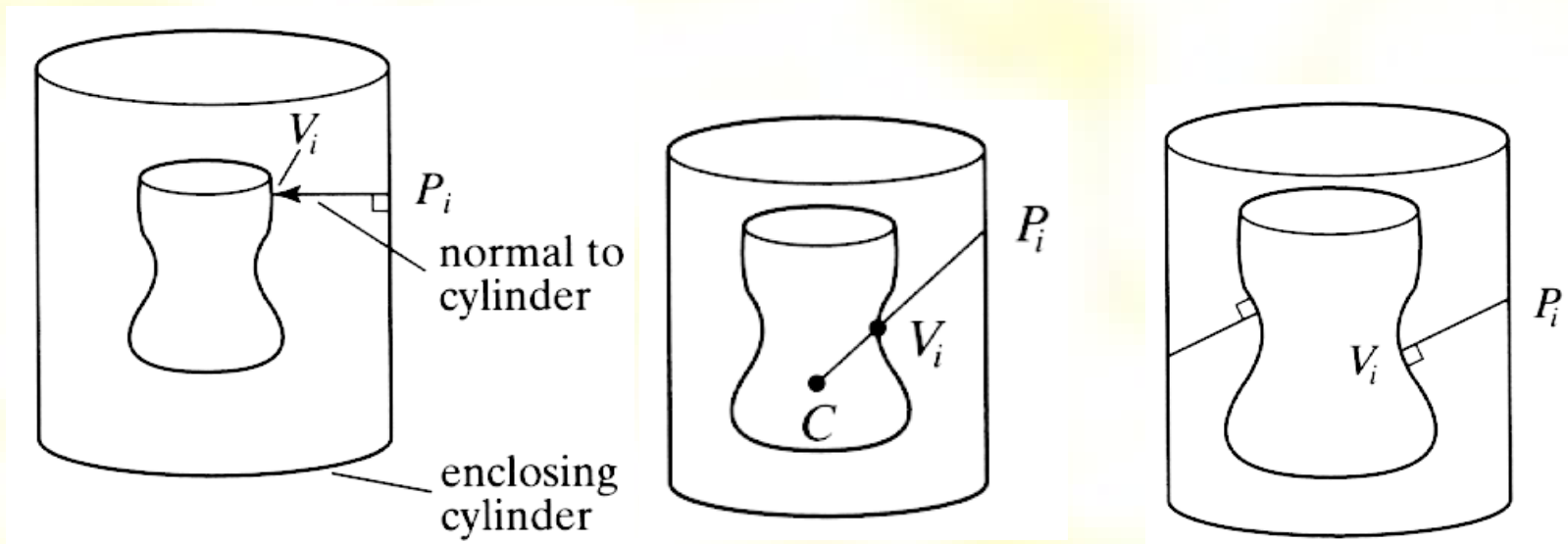
# Cylindrická projekcia – S mapovanie



$$s = \frac{\theta - \theta_A}{\theta_B - \theta_A} \quad t = \frac{z - z_A}{z_B - z_A}$$

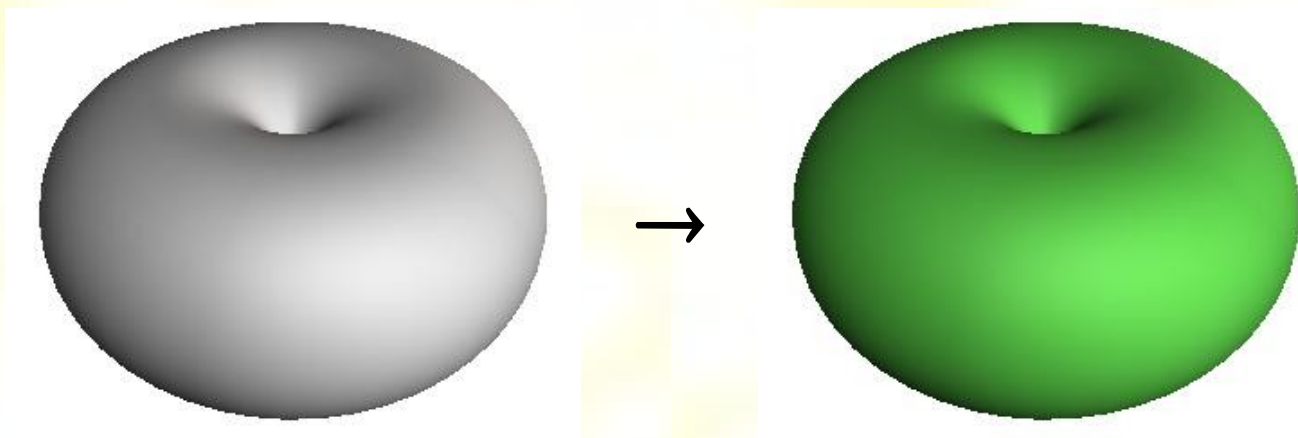
# Cylindrická projekcia – O mapovanie

- Niekoľko spôsobov:

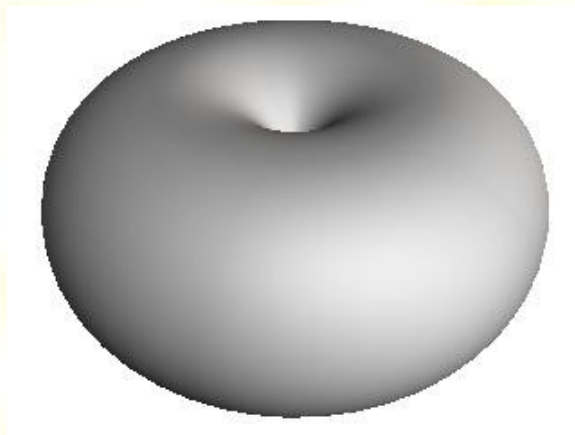
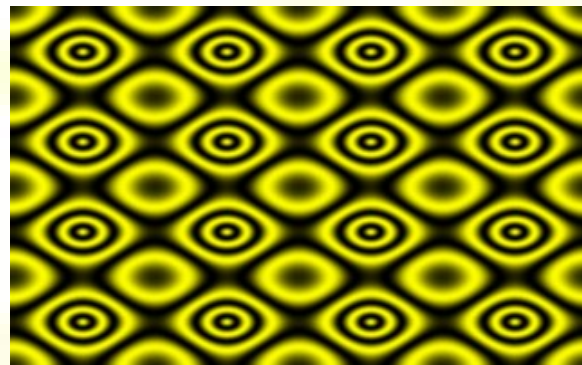


- najbežnejšie je kolmé premietanie

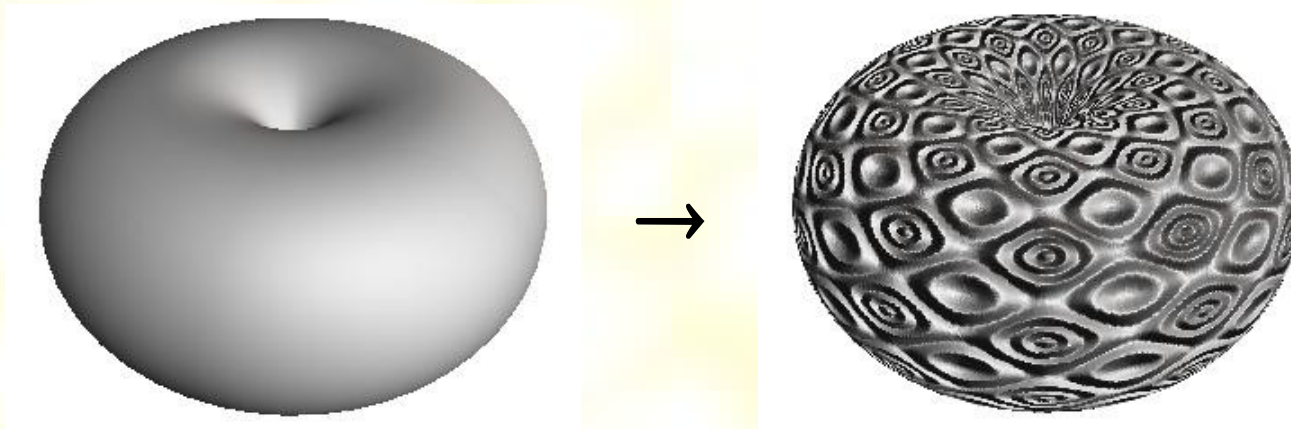
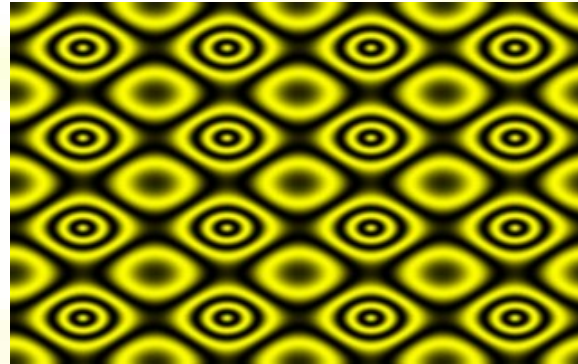
# Farebné textúry – príklad 1



# Farebné textúry – príklad 2

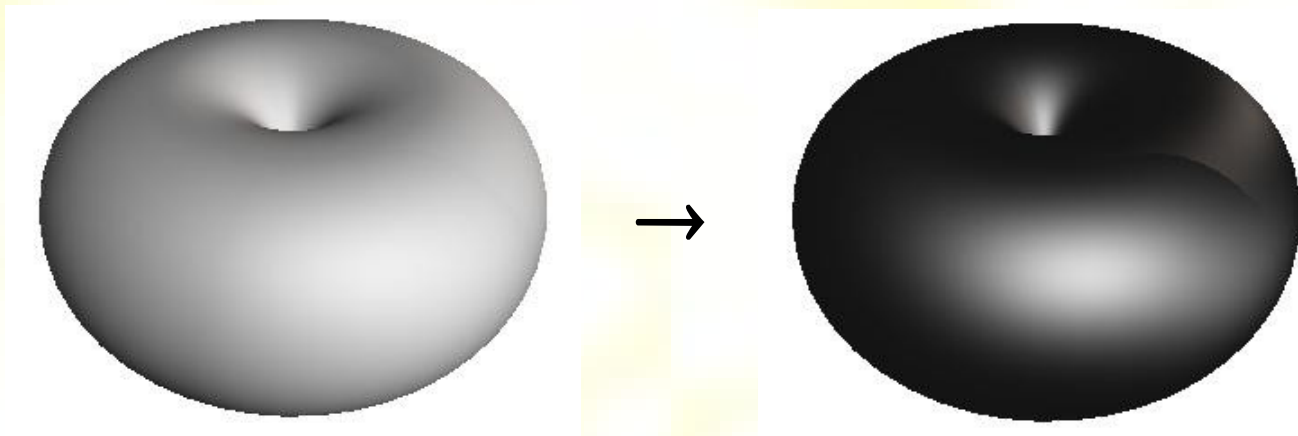


# Hrbol'até textúry – príklad



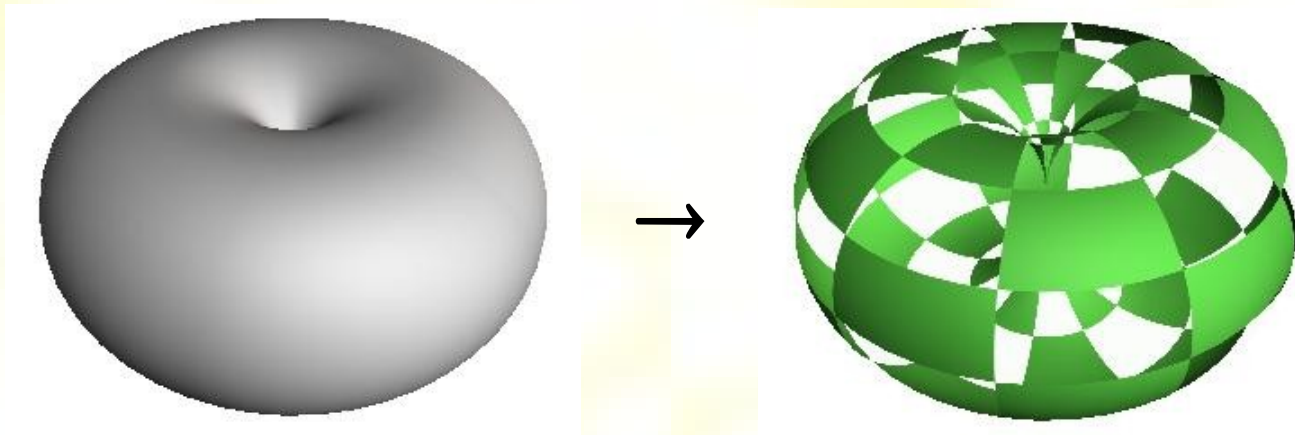


# Textúry lesklosti – příklad

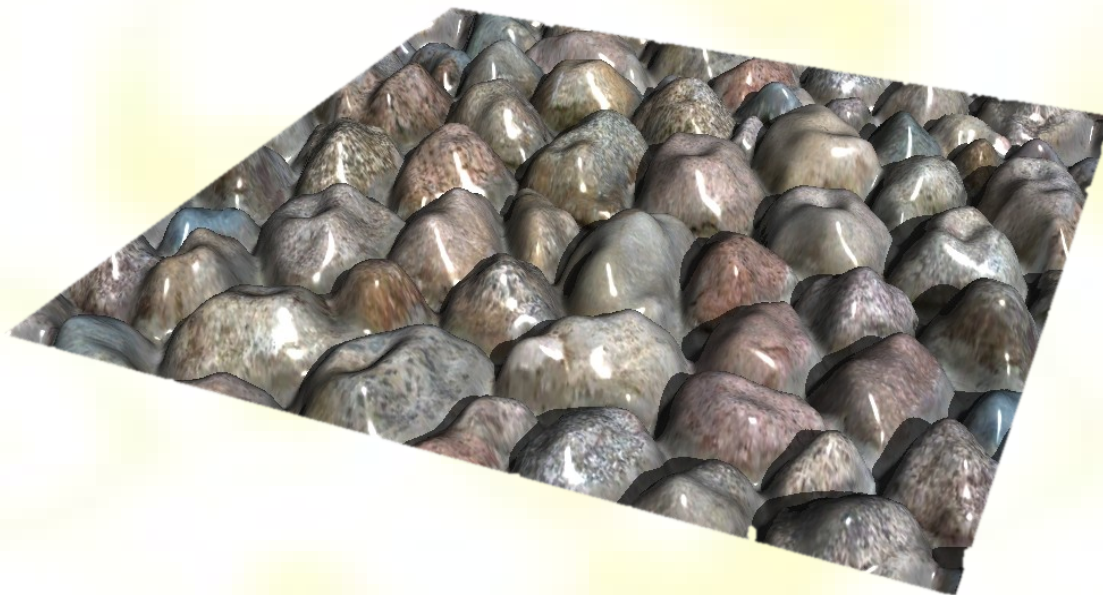


kovový lesk

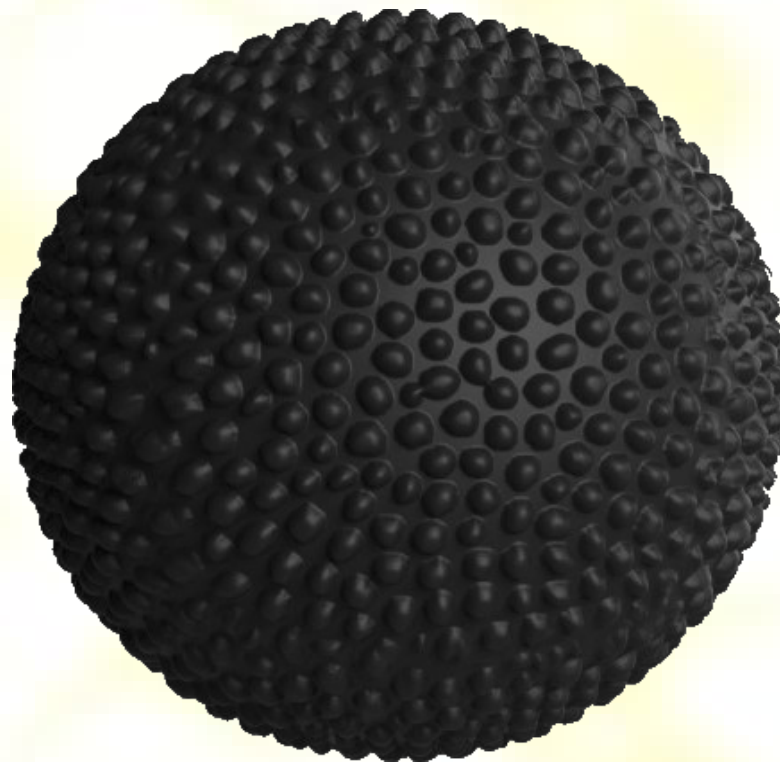
# Textúry priehľadnosti – príklad



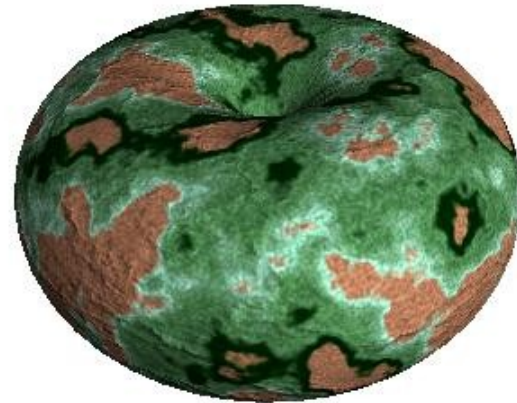
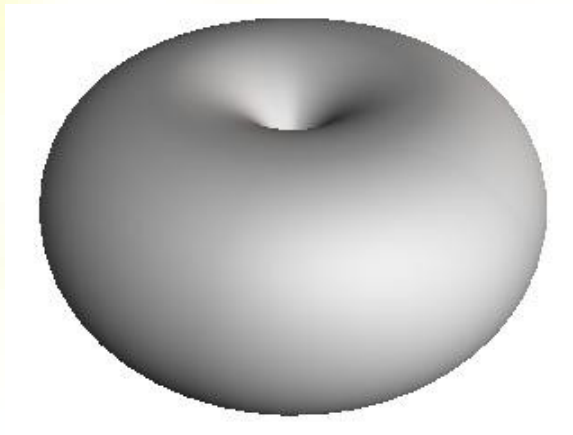
# Reliefne textúry – príklad



# Displacement textúry – príklad

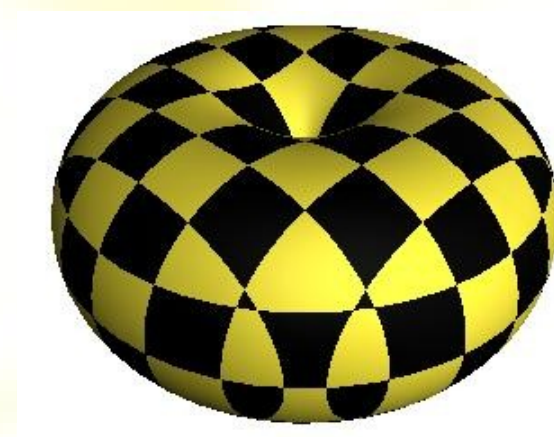
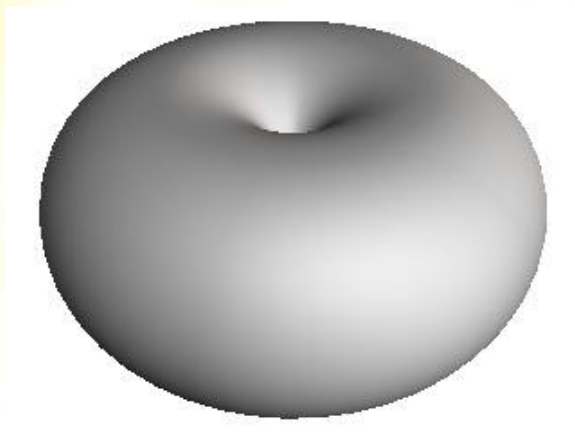


# Kombinované textúry – príklad



camouflage

# 3D textúry – príklad



solid cubes

# Environment mapping

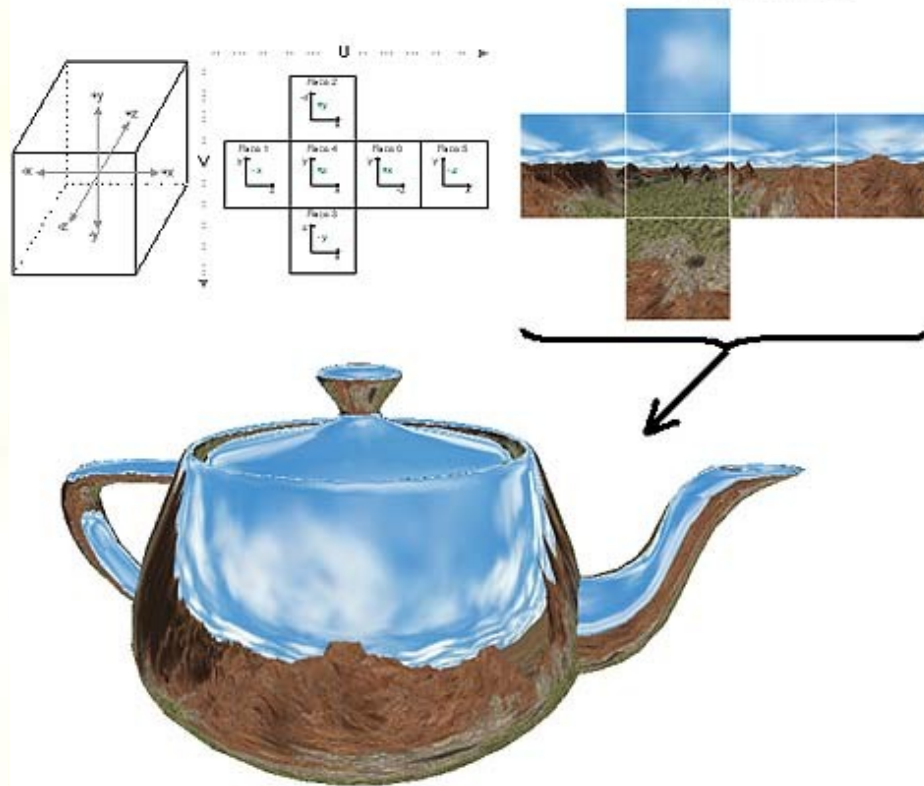
- Ako textúru vezmime okolité prostredie



- Ako získať takúto textúru?
- Blinn & Newell: vzorkovanie cez polárne súradnice ( $\theta$ ,  $\phi$ )

# Cubic environment mapping

- Gene 1986
- Okolité prostredie sa zobrazuje na 6 stien kocky
- Najviac používaná v GPU



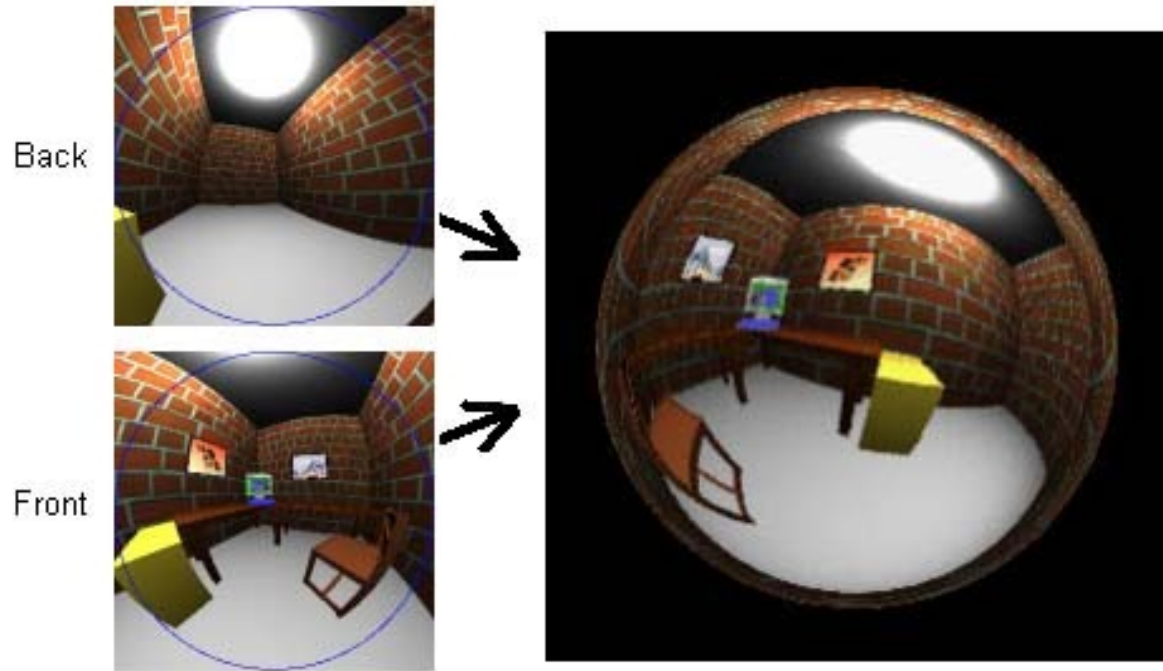


# Sphere environment mapping

- Okolie je zobrazené na guľu



# Paraboloid environment mapping



Zadná:

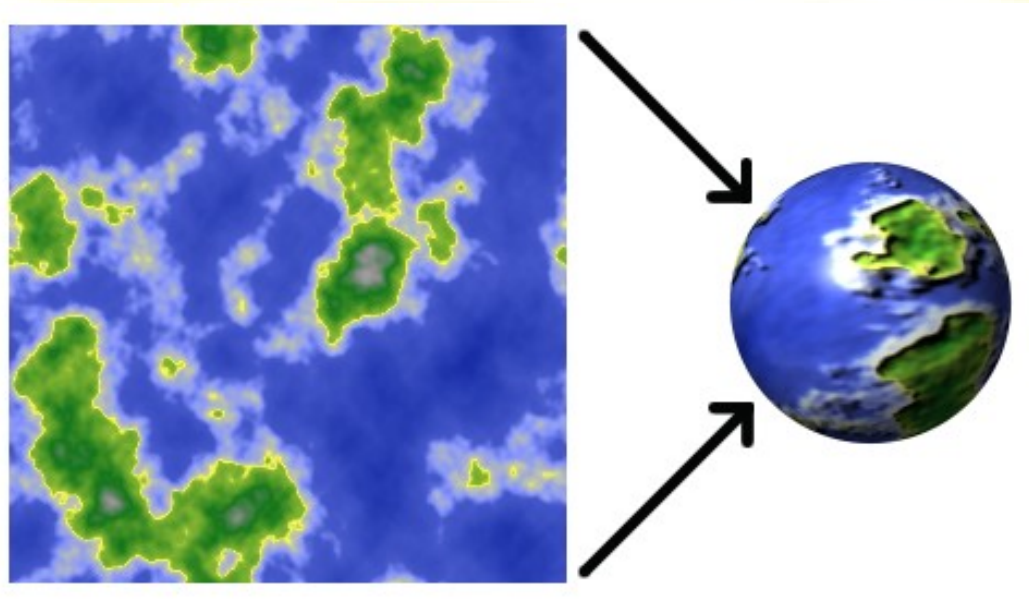
$$u = \frac{1}{2b} \frac{x}{1-z} + \frac{1}{2}$$
$$v = \frac{1}{2b} \frac{y}{1-z} + \frac{1}{2}$$

Predná:

$$u = \frac{1}{2b} \frac{x}{1+z} + \frac{1}{2}$$
$$v = \frac{1}{2b} \frac{y}{1+z} + \frac{1}{2}$$

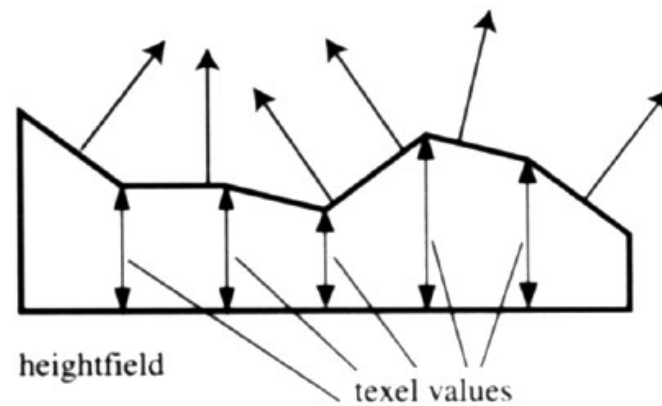
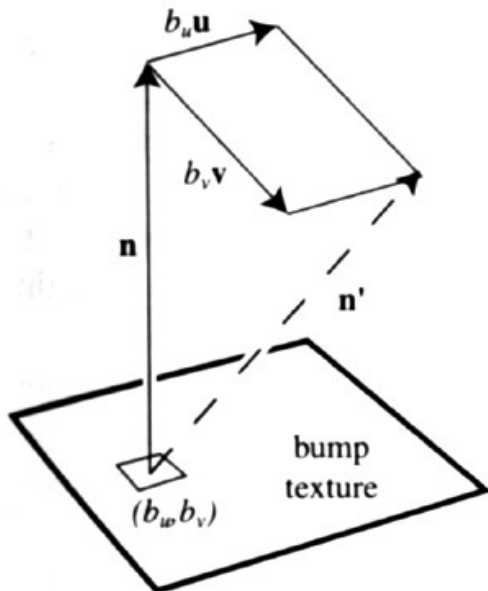
# Bump Mapping

- Dodávajú objektom realistickejší vzhľad
- Vytvárajú vizuálne nerovný povrch na objektoch, ktoré sú často úplne hladké
- Používa sa pripravená výšková mapa, ktorá je aplikovaná na danú textúru



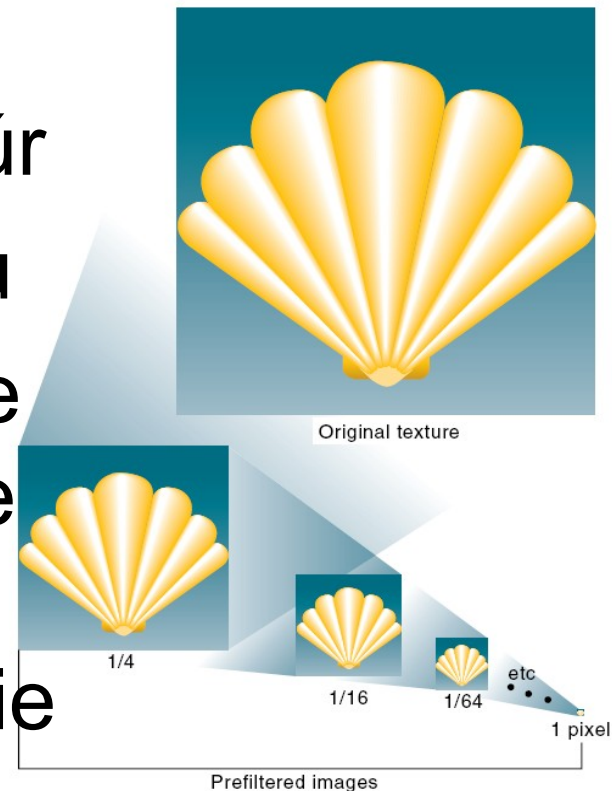
# Bump mapping

- Možnosti uloženia textúry:
  - 2 hodnoty bod určujúce posunutie ( $u, v$ )
  - výšková mapa
  - pole normálových vektorov



# Mipmapping

- MIP = Multum In Parvo (mnoho v malom)
- Predpovypočítanie zmenšených, filtrovaných kópií textúry a ich uloženie v pamäti
- Urýchlenie zobrazovania textúr
- Pyramída s  $(u, v, d)$  adresáciou
- Vyberieme buď kópiu, ktorá je najbližšie, alebo interpolujeme medzi dvoma najbližšími kópiami = trilineárne filtrovanie



# Filtrovanie textúr

- Odstraňuje „štvorčekový efekt“, ktorý vzniká pri aplikovaní malých textúr na veľký objekt
- Používa sa buď jednoduchý bilineárny či trilineárny filter, alebo anizotropické filtrovanie



No bilinear filtering



bilinear filtering

# Ako získať textúru?

- Použiť predpripravené dáta (fotografia)
- Upraviť predpripravené dáta (filtrovanie)
- Vytvoriť textúru procedurálne

# Procedurálne textúry



# Procedurálne textúry

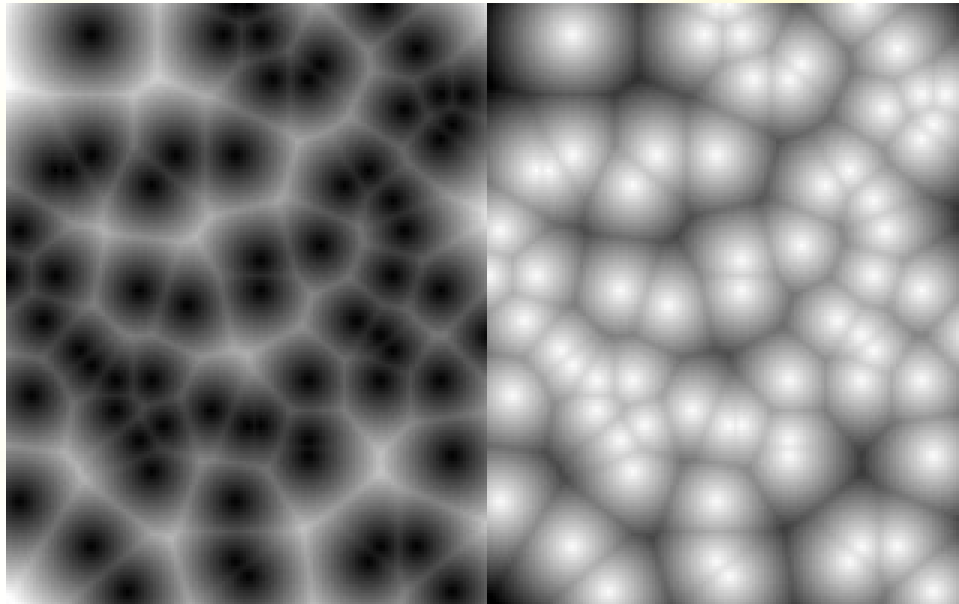
- Vyžaduje naprogramovať netriviálne algoritmy
- Veľká flexibilita, ovplyvnenie výsledku pomocou vstupných parametrov
- Neohraničené rozlíšenie
- Antialiasing pomocou nadvzorkovania
- Nízke požiadavky na pamäť
- Možné prispôbiť na ľubovoľnú geometriu objektu
- Možnosť generovať viac 2D, 3D, 4D, ... textúry

# Metódy tvorby procedurálnych textúr

- Každá rastrová (2D, 3D, 4D) procedurálna modelovacia technika
- Explicitné funkcie
- Fraktály
- Šumy, turbulencie
- Reakčná difúzia
- ...

# Tvorba textúry pomocou Voronoiových buniek

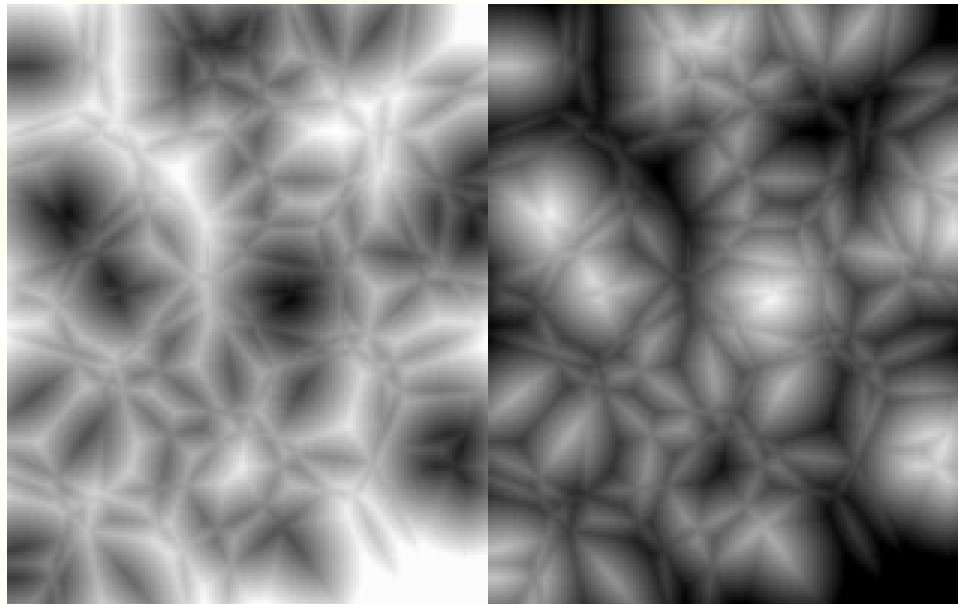
1. Náhodne vygenerovať  $n$  bodov
2. Vizualizovať vzdialenosť k najbližšiemu susedovi



- Výsledok = Voronoiov šum

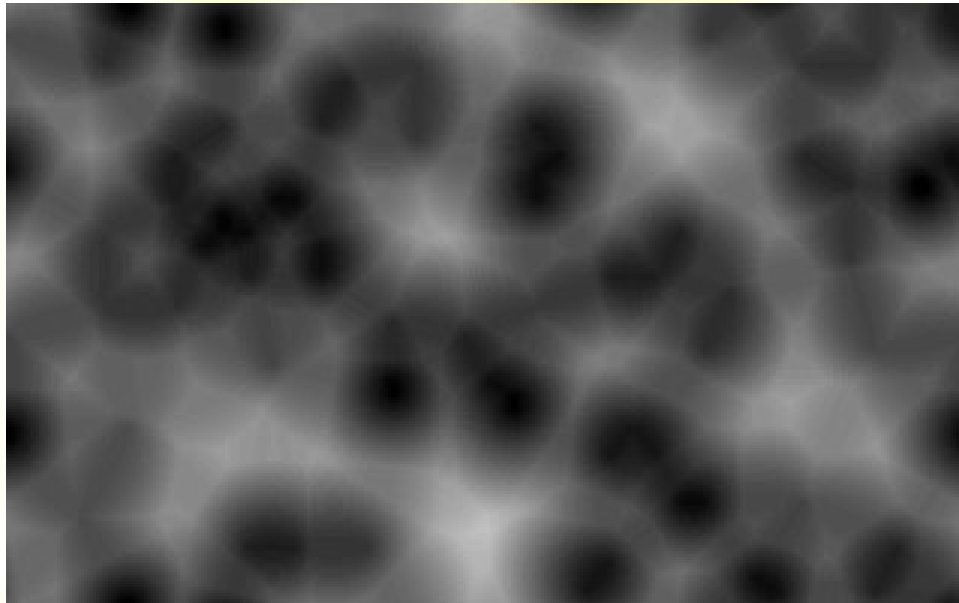
# Modifikácie metódy najbližšieho suseda

Zovšeobecnenie:  $i$ -ty najbližší sused



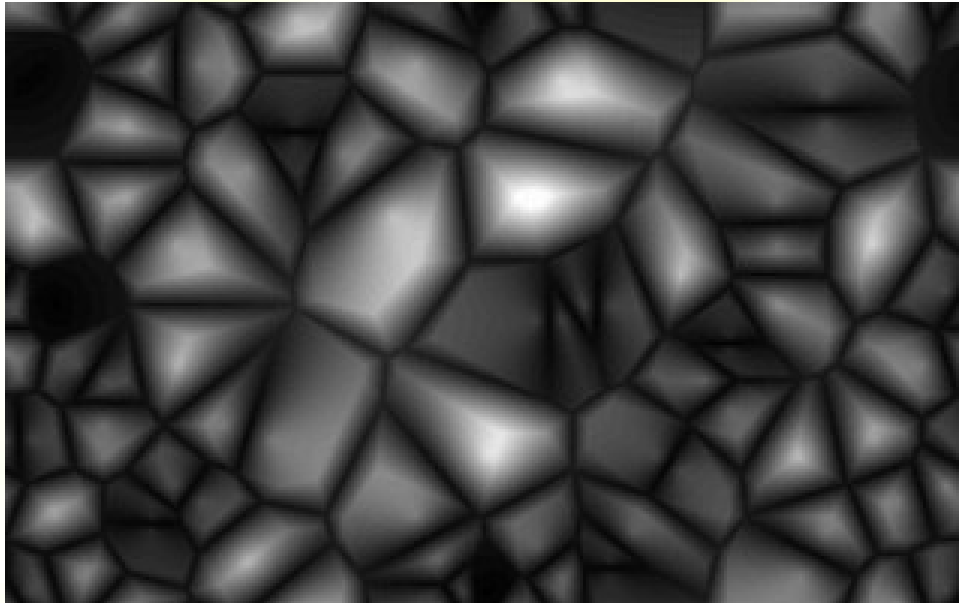
tretí najbližší sused

# Súčet vzdialeností



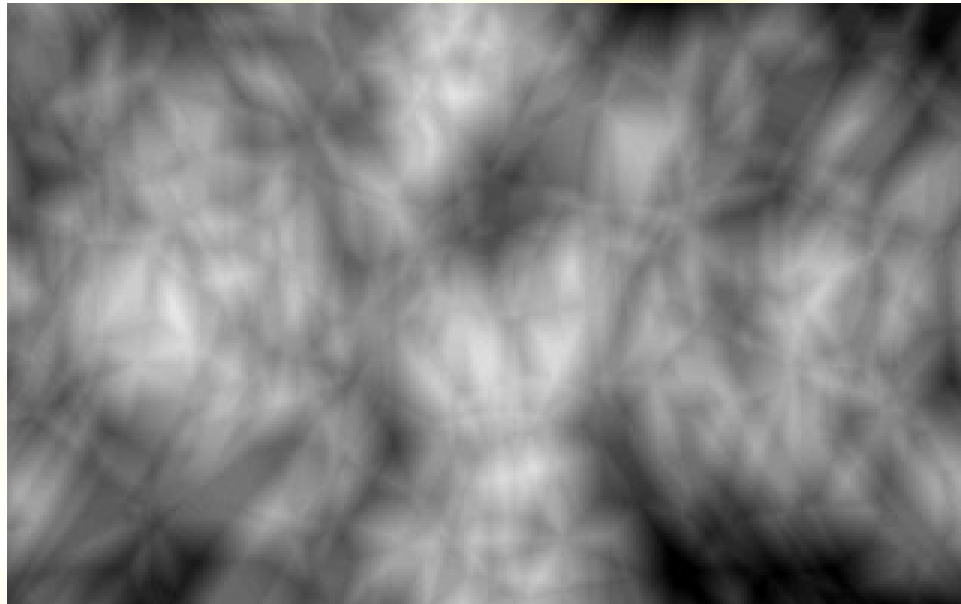
1. najbližší plus 2. najbližší sused

# Rozdiel vzdialeností



2. najbližší mínus 1. najbližší sused

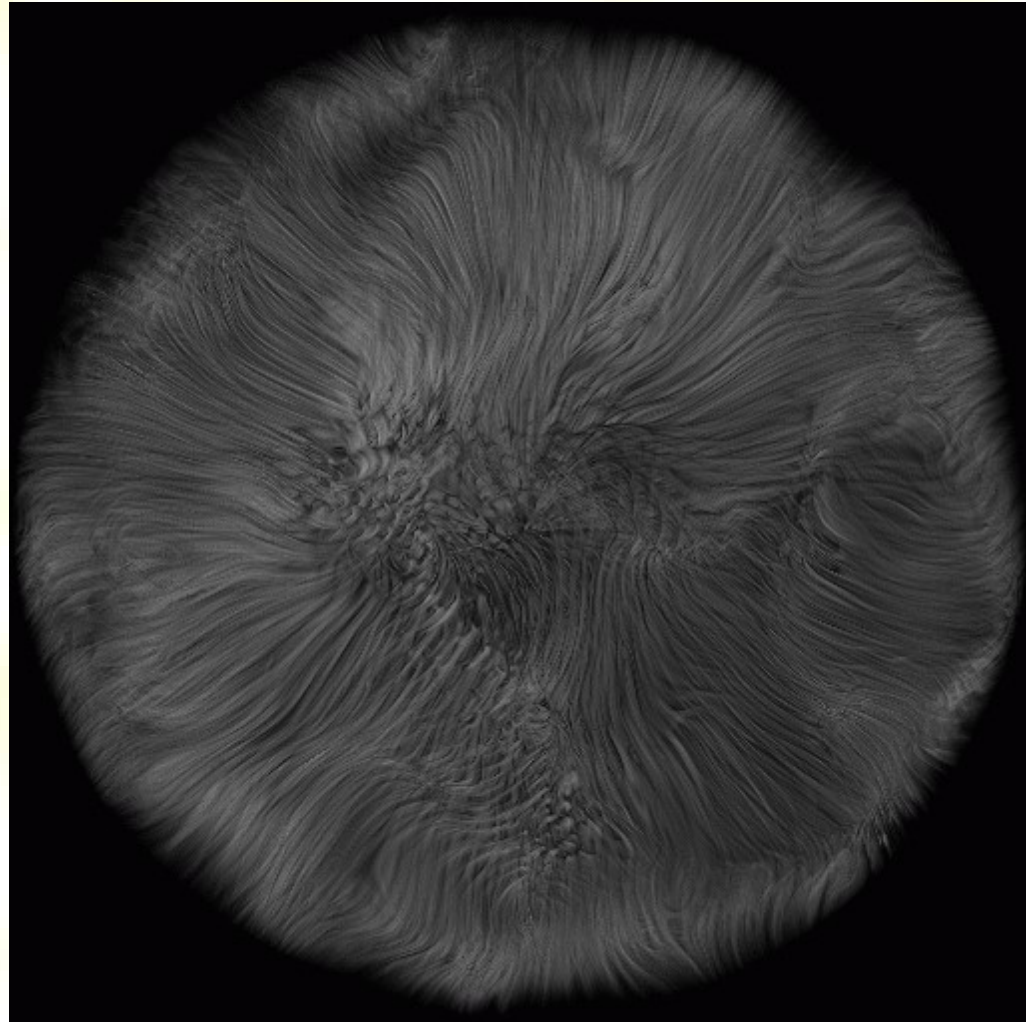
# Kombinácia najbližšieho a najvzdialenejšieho



3. najbližší plus 3. najvzdialenejší

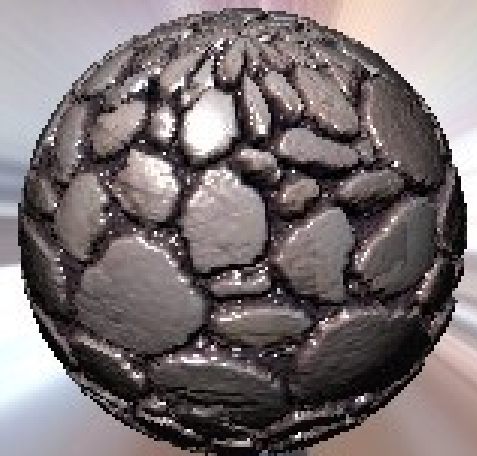
# 3D Textúry: Hypertextúry

- Perlin 1989  
3D funkcia hustoty  
 $d(x) \in \langle 0, 1 \rangle$
- Fuzzy objekty
  - oblaky
  - Srst', vlasy,
  - plyn, oheň, dym
- Komplexné objekty
  - tkaniny
  - minerály
- Použitie šumu

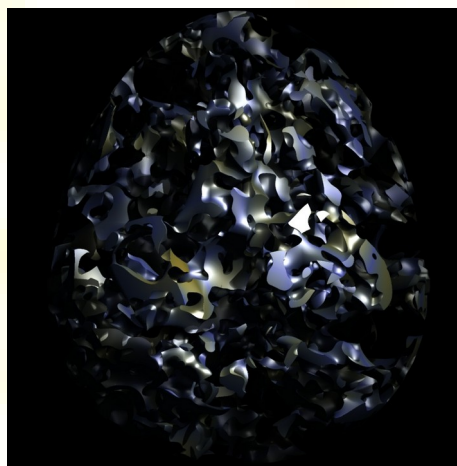
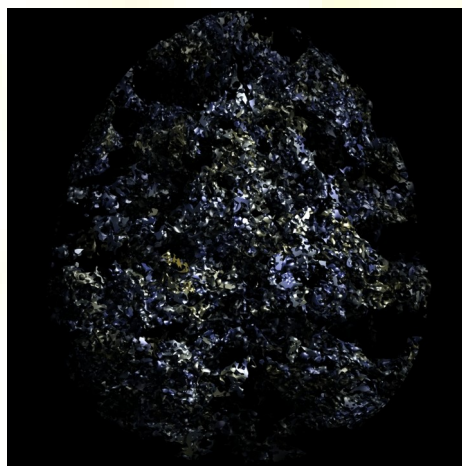
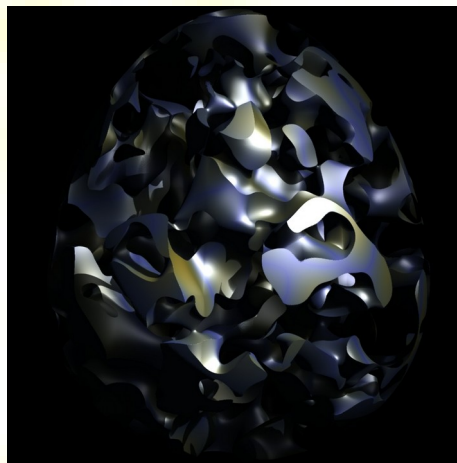
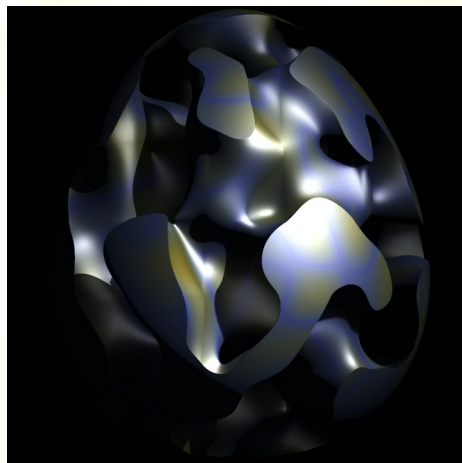




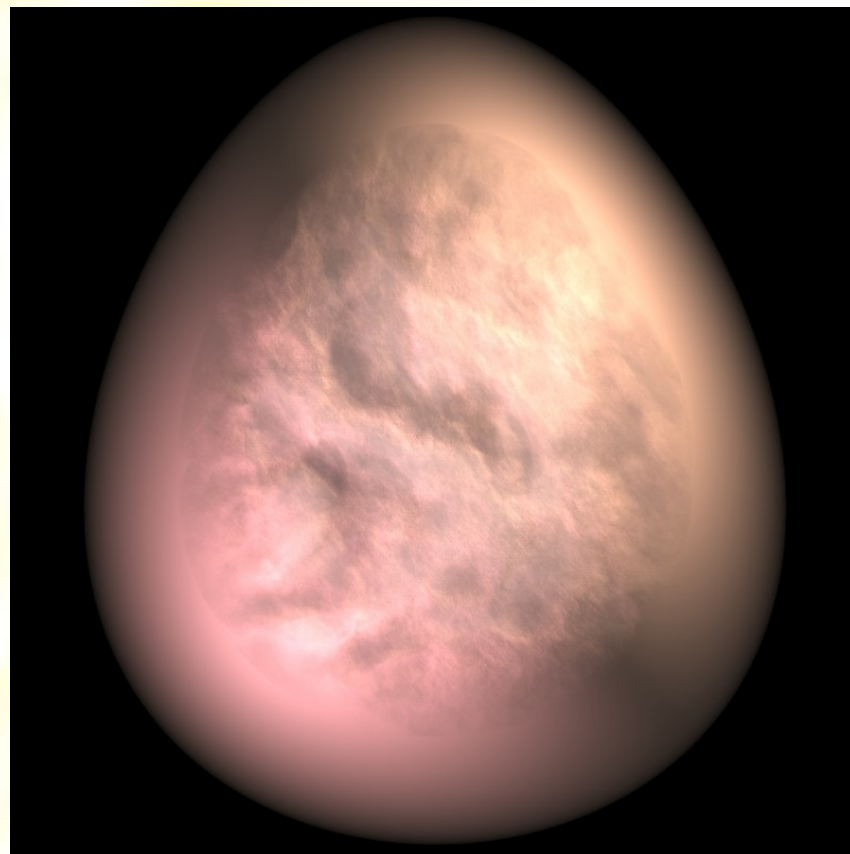
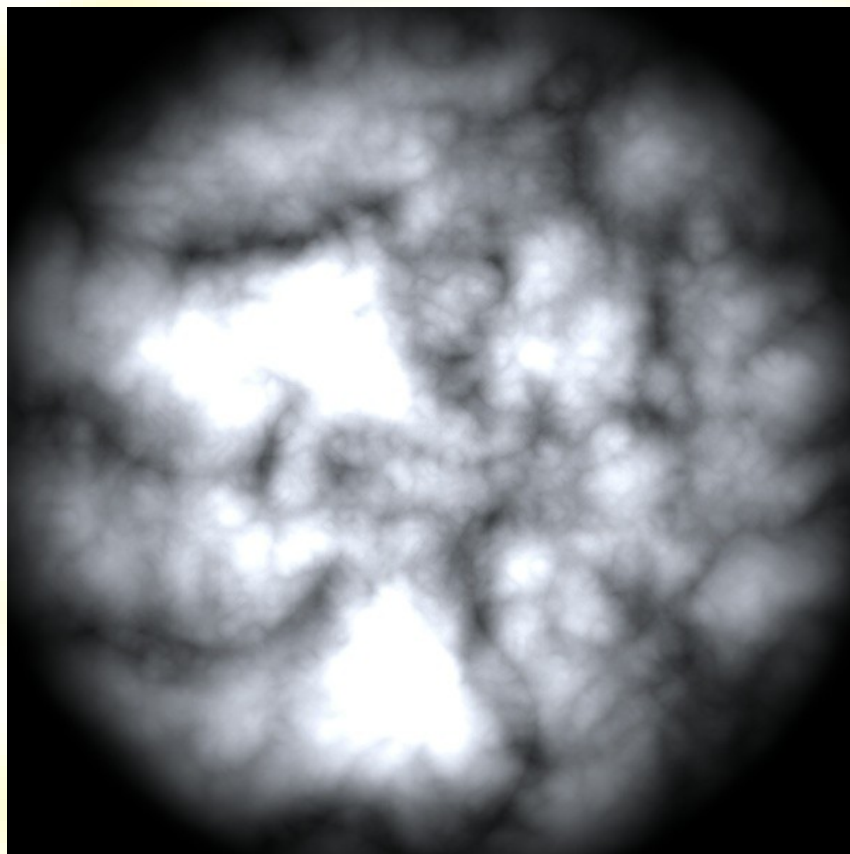
# Hypertextúry



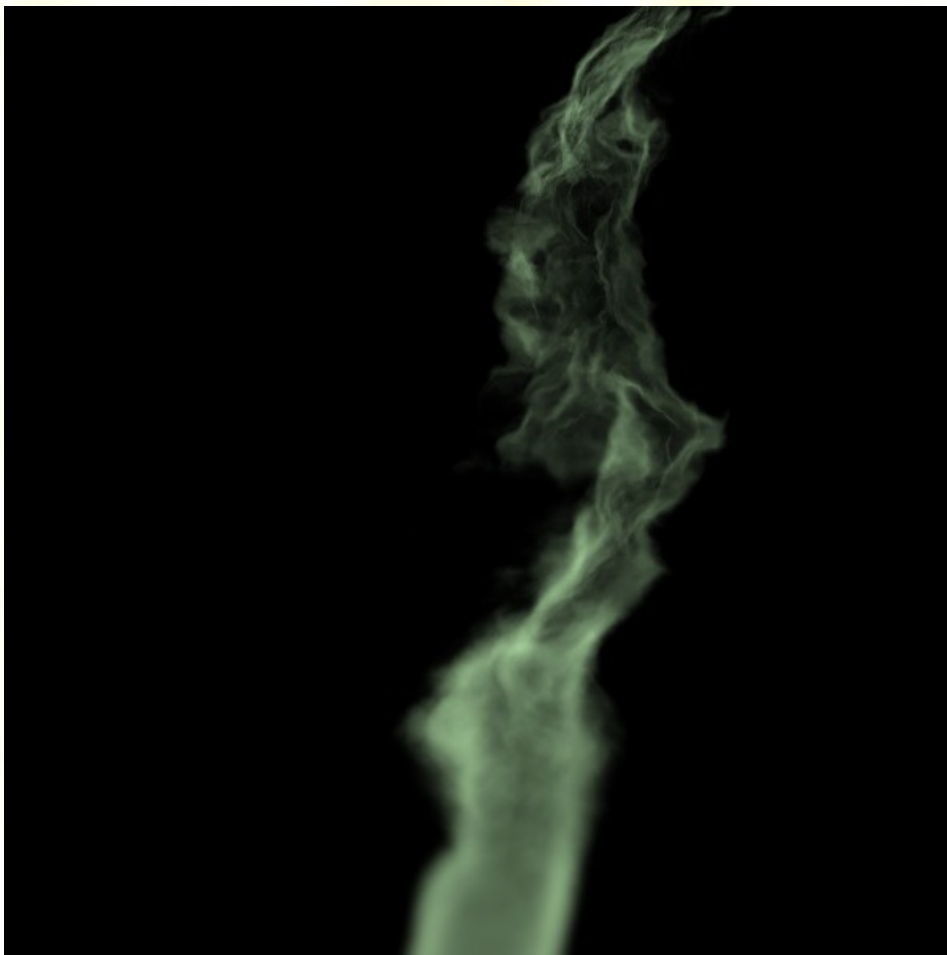
# Perlinová hypertextúra



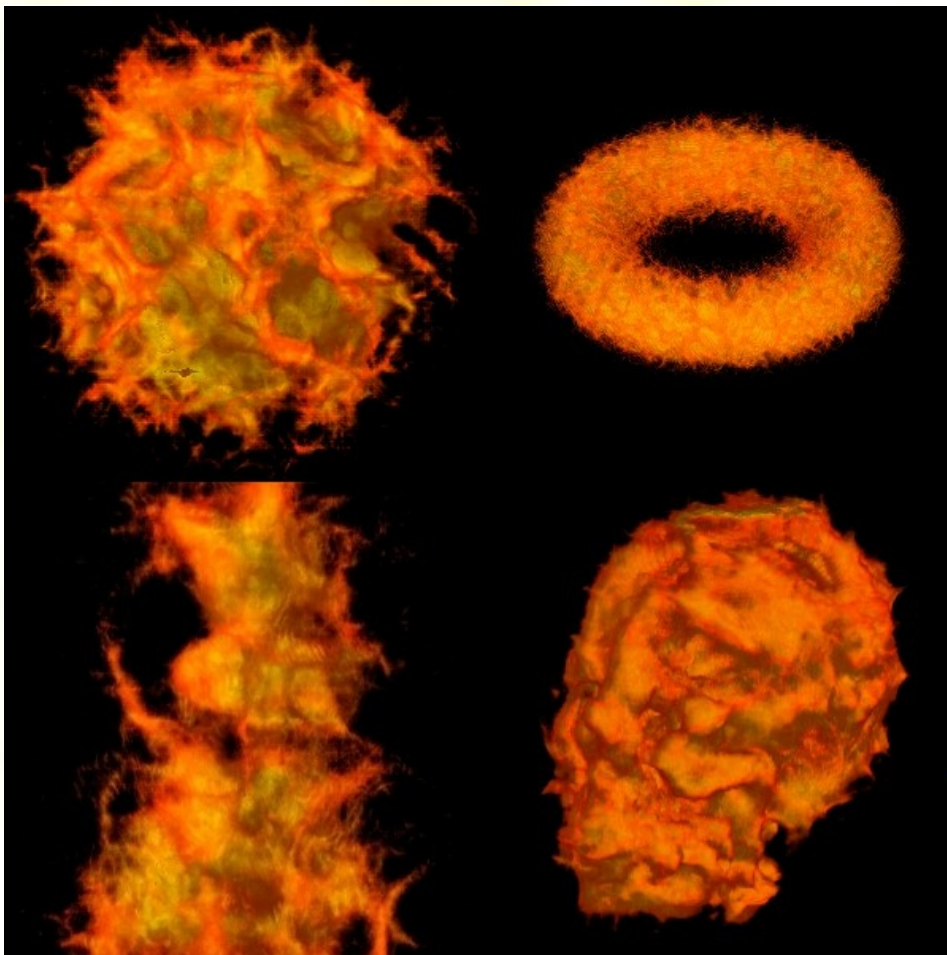
# Fuzzy hypertextúry



# Hypertextúry dymu

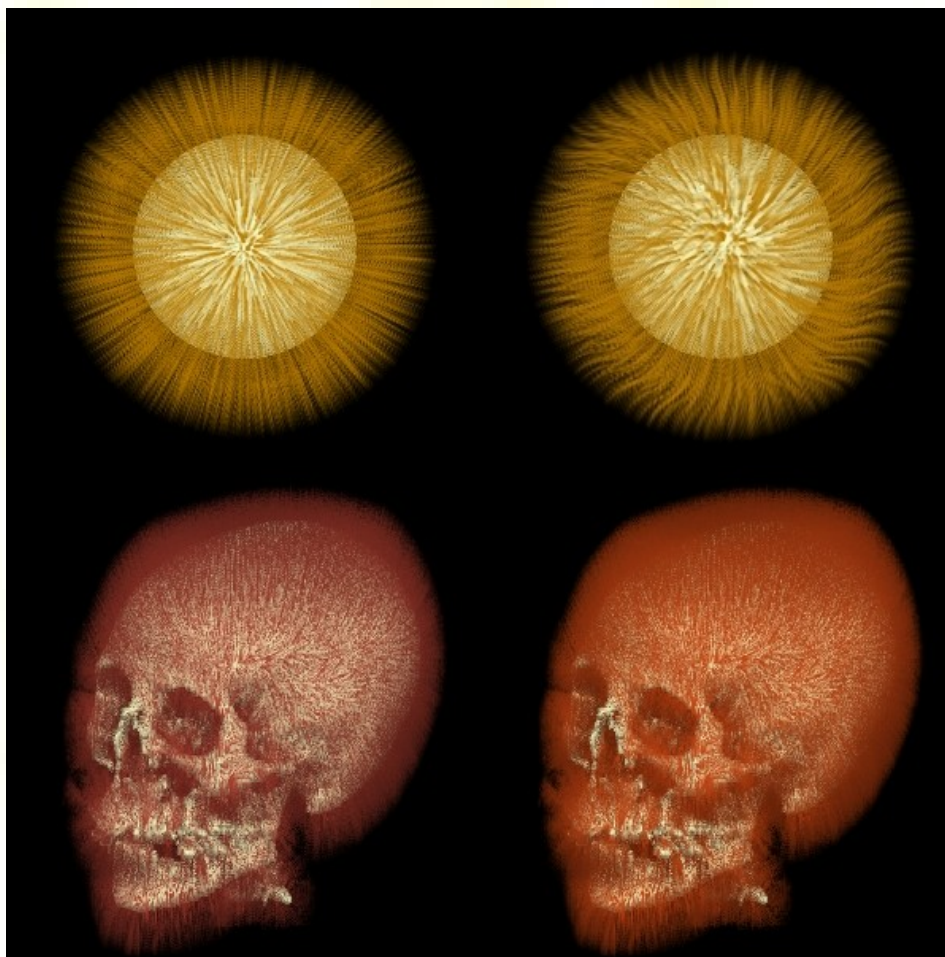


# Hypertextúry ohňa

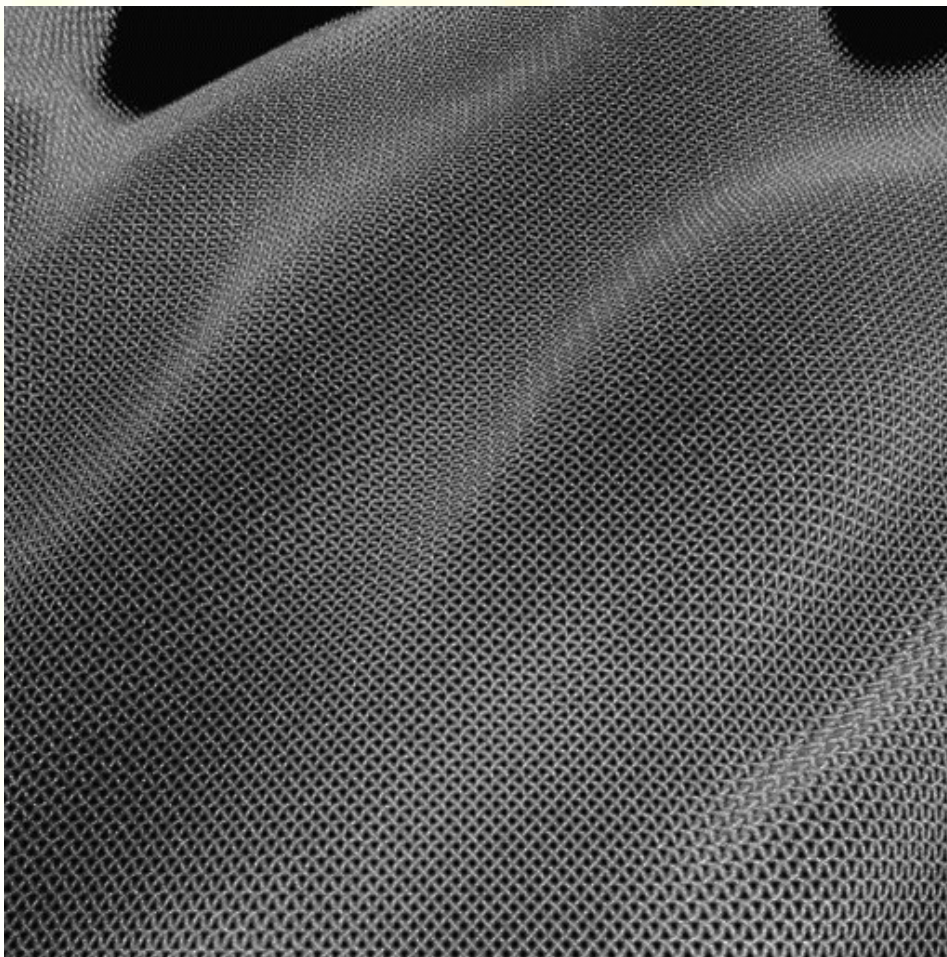




# Hypertextúry srsti, vlasov



# Hypertextúry tkaniny



# Niektoré zdroje

- Knihy:
  - Tomas Akenine-Moeller, Eric Haines: Realtime Rendering, 2nd edition, AK Peters 2002
  - D.S. Ebert et al: Texturing and Modeling, 1994, 1998
- Web:
  - [www.opengl.org](http://www.opengl.org)
  - [www.gamedev.net](http://www.gamedev.net)
  - [www.nvidia.com](http://www.nvidia.com)