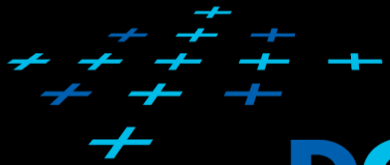


DCGI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

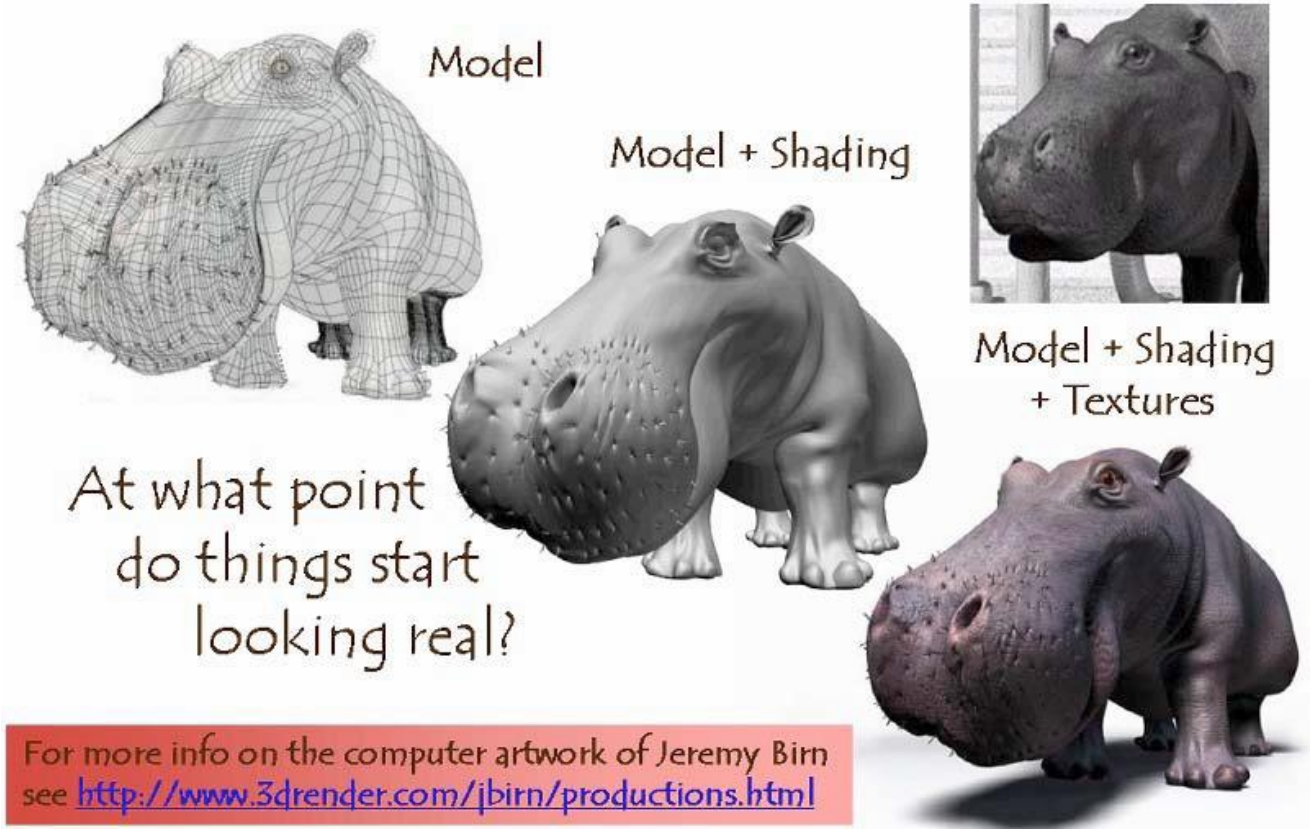
Textures

Jiří Bittner, Vlastimil Havran

Textures

- Motivation - What are textures good for? MPG 13
- Texture mapping principles
- Using textures in rendering
- Summary

Textures Add Details



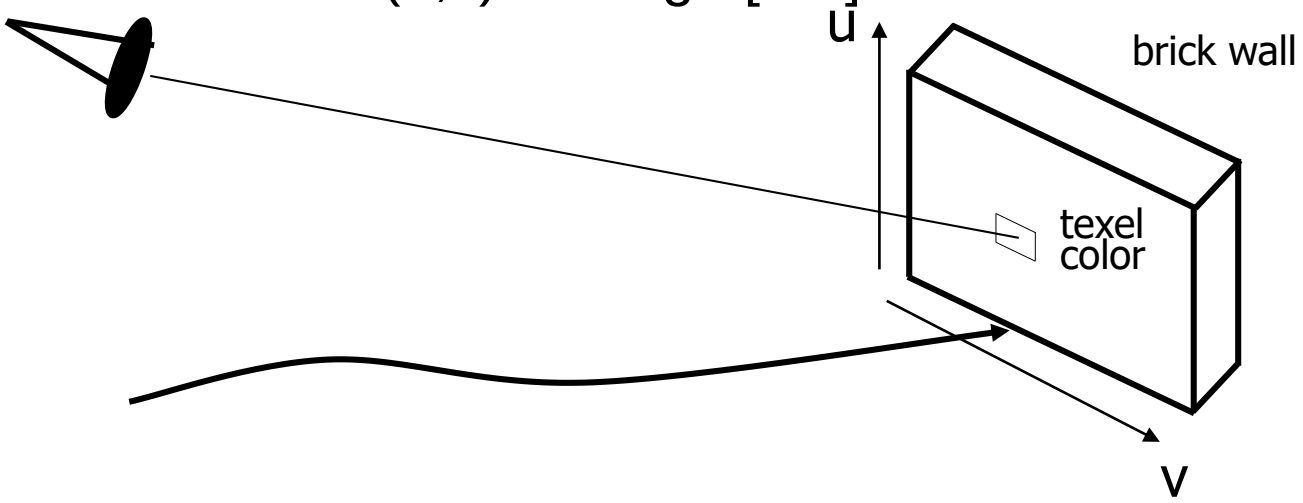
Cheap Way of Increasing Visual Quality

Textures - Introduction

- Surface macrostructure
- Sub tasks:
  - Texture definition: image, function, …
  - Texture mapping
    - positioning the texture on object (assigning texture coordinates)
  - Texture rendering
    - what is influenced by texture (modulating color, reflection, shape)

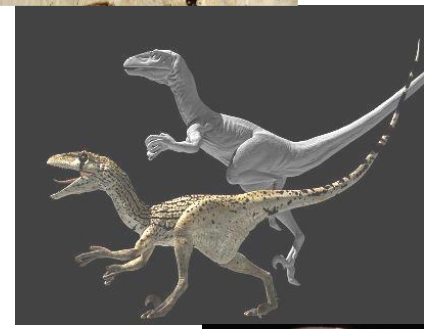
Typical Use of (2D) Texture

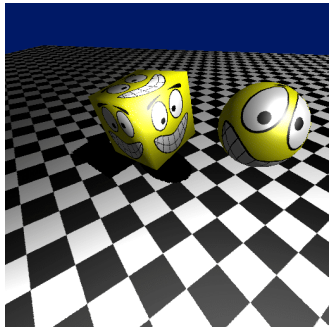
- Texture coordinates (u,v) in range $[0\text{-}1]^2$

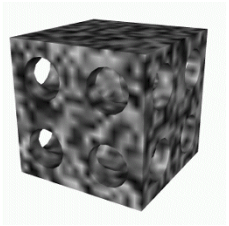


Texture Data Source

- Image
  - Data matrix
  - Possibly compressed
- Procedural
  - Simple functions (checkerboard, hatching)
  - Noise functions
  - Specific models (marvle, wood, car paint)







Texture Dimension

- 2D – images
- 1D – transfer function (e.g. color of heightfield)
- 3D – material from which model is manufactured (wood, marble, …)
  - Hypertexture – 3D model of partly transparent materials (smoke, hair, fire)

- +Time – animated textures

Texture Data

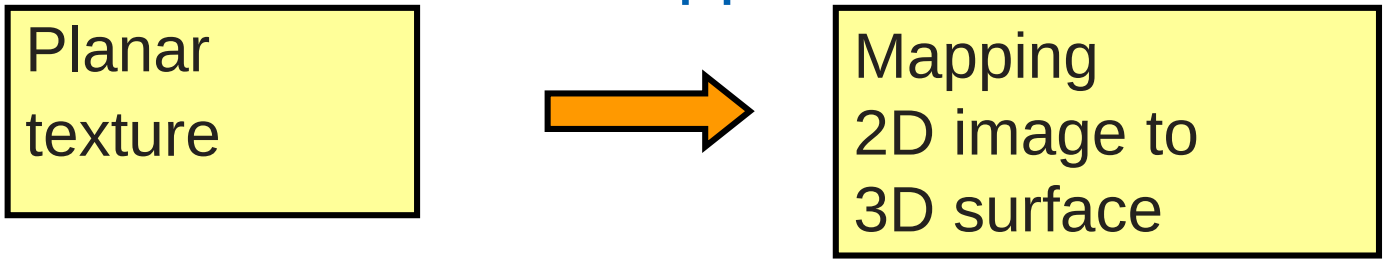
- Scalar values
  - weight, intensity, …
- Vectors
  - color
  - spectral color

Textures

- Motivation - What are textures good for? MPG 13
- Texture mapping principles
- Using textures in rendering
- Summary

Texture Mapping Principle

Texture application



(Inverse) texture mapping

T: [u v] –> Color M: [x y z] –> [u v]

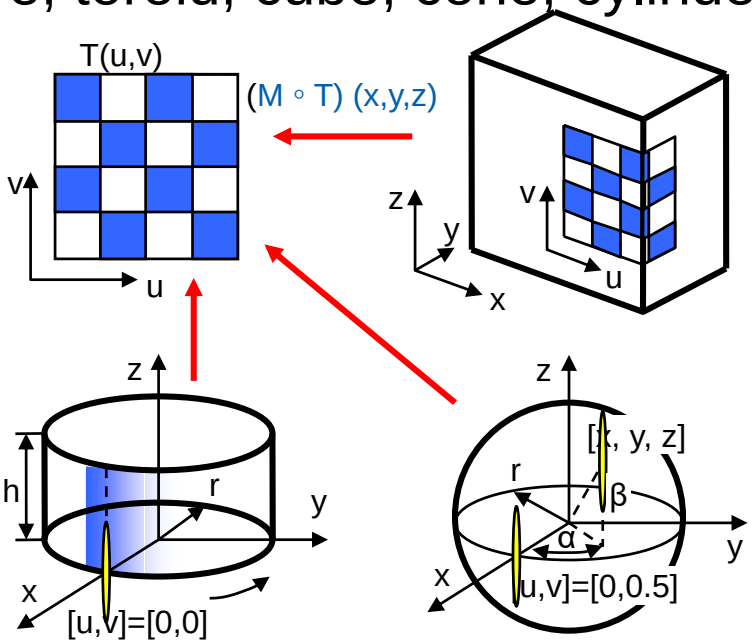
M ◦ T: **[x y z] –> [u v] –> Color**

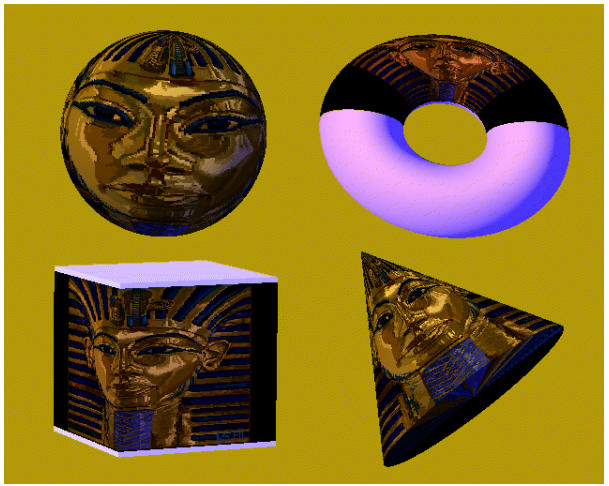
Texture Mapping – Basic Principles

- Inverse mapping
- Geometric mapping using proxy surface
- Environment mapping

Inverse Texture Mapping – Simple Shapes

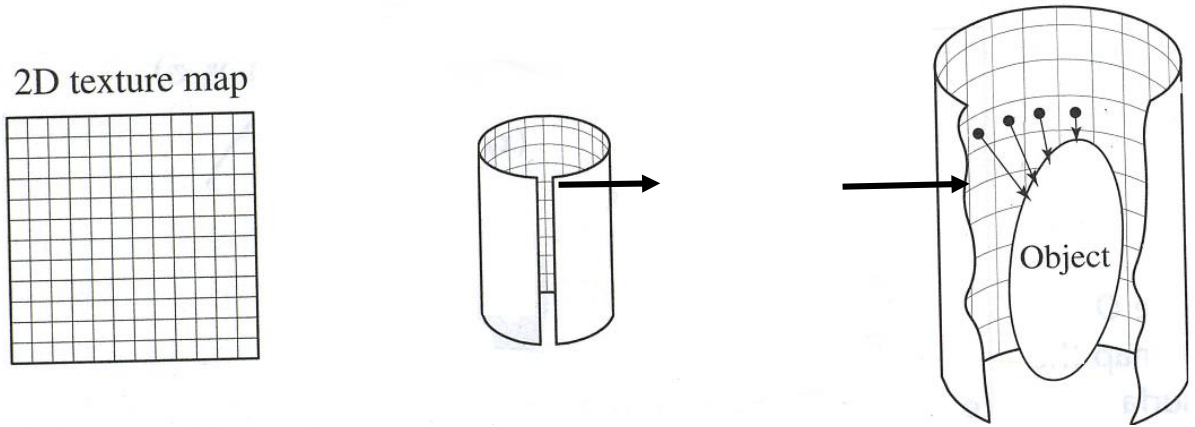
- sphere, toroid, cube, cone, cylinder





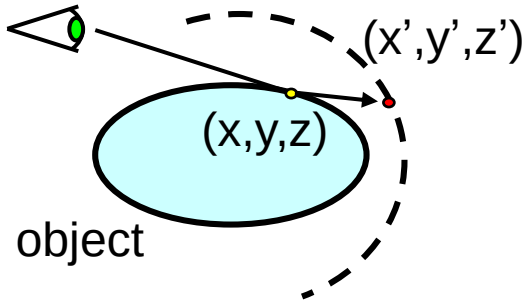
Texture Mapping using Proxy Surface

- Proxies: sphere, toroid, cube, cone, cylinder
  - Proxy attached to object and "*projected*"
- *First step: texture to proxy*
- *Second step: proxy to object*

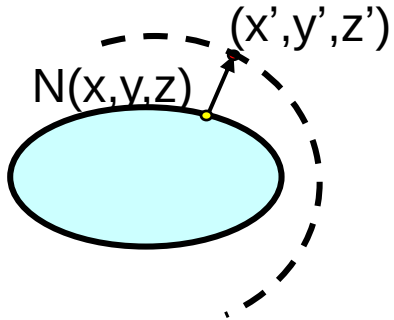


Texture to Proxy Proxy to Object

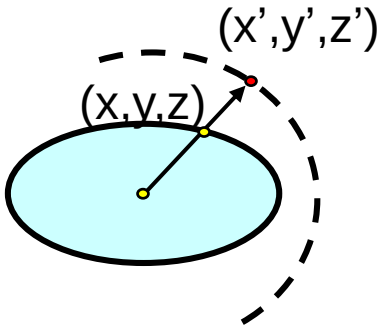
Proxy To Object Inverse Mapping



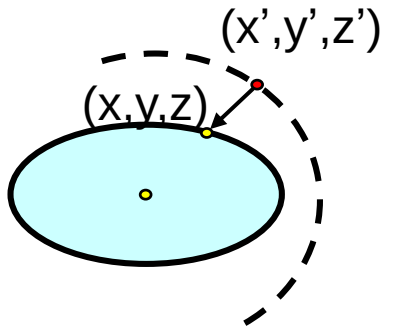
- Reflected ray



- Object surface normal

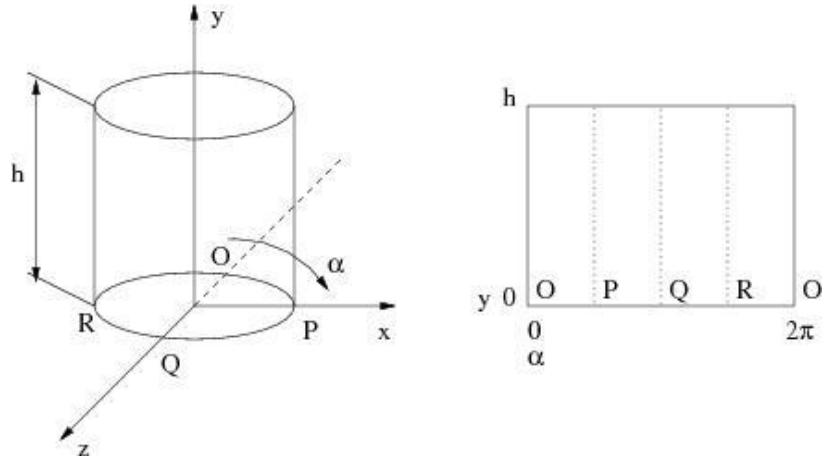


- Object centroid



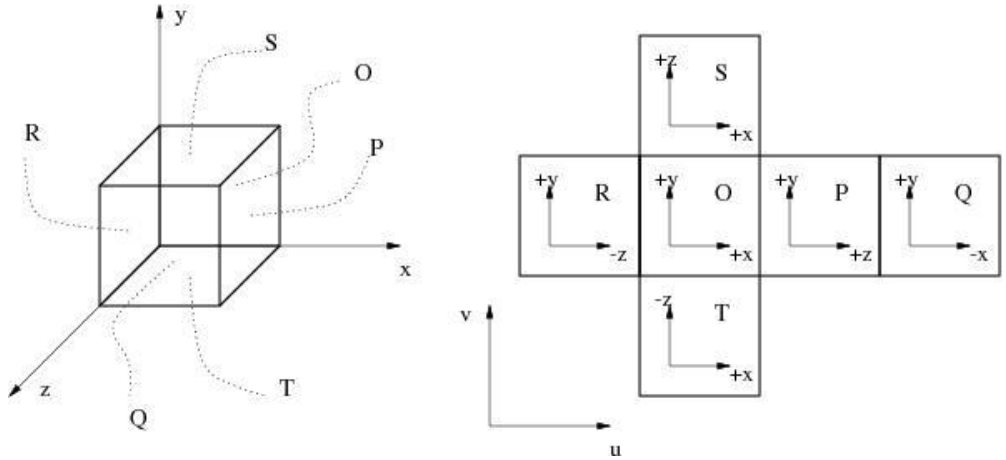
- Proxy surface normal

Cylinder Proxy



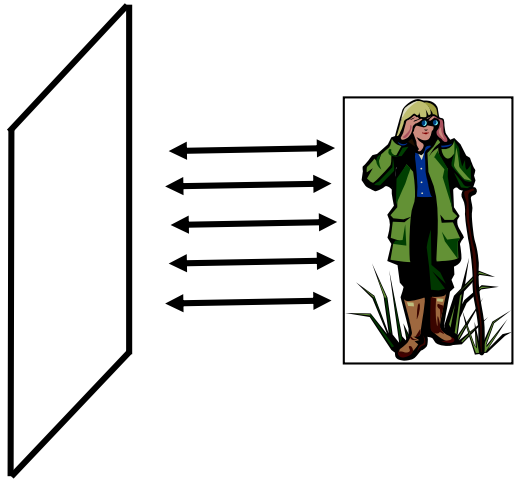
Cube Proxy

- 6 textures



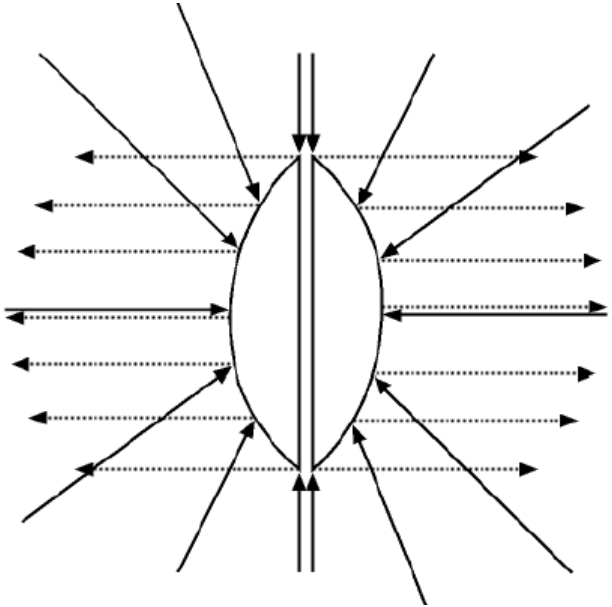
Planar Proxy

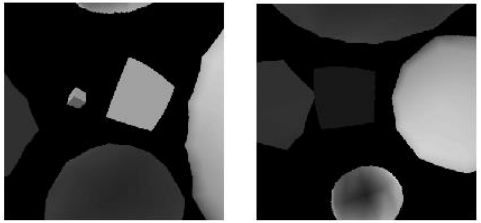
- Orthographic projection



Dual Paraboloid

- Heidrich and Seidel, 1998





Environment Mapping

- Cheap alternative to ray tracing
- Direction of reflected ray -> texture lookup
- Proxy – sphere, cube, dual paraboloid, tetrahedron, octahedron
- Two phases:
  - Creating environment map (using expected camera position)
  - Using environment map during rendering

Environment Map vs Ray Tracing

Ray Traced

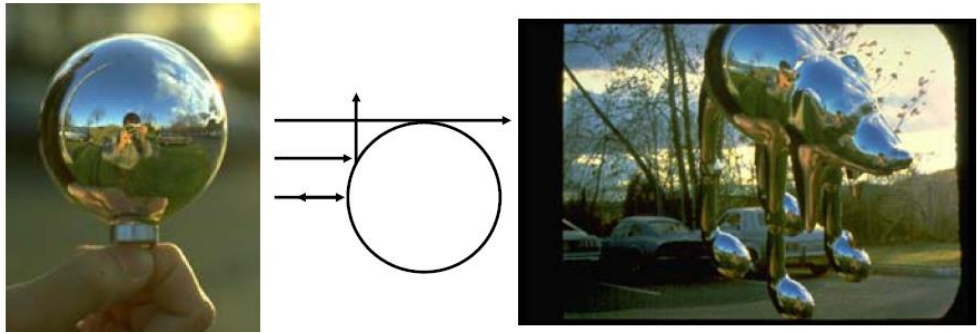
Environment Map

Terminator II (1991)

Getting Environment Map

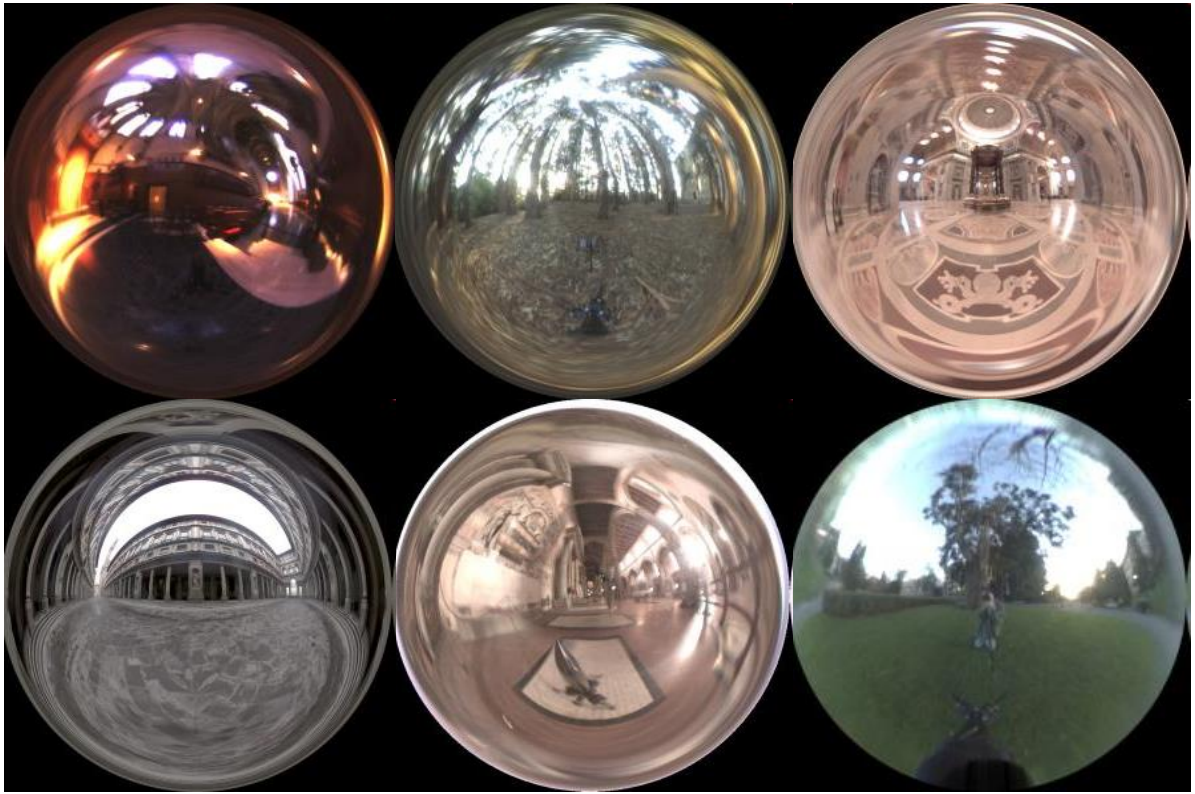
- Rendering
- Special camera
- Spherical mirror + camera with telescopic lens + processing

Miller and Hoffman, 1984



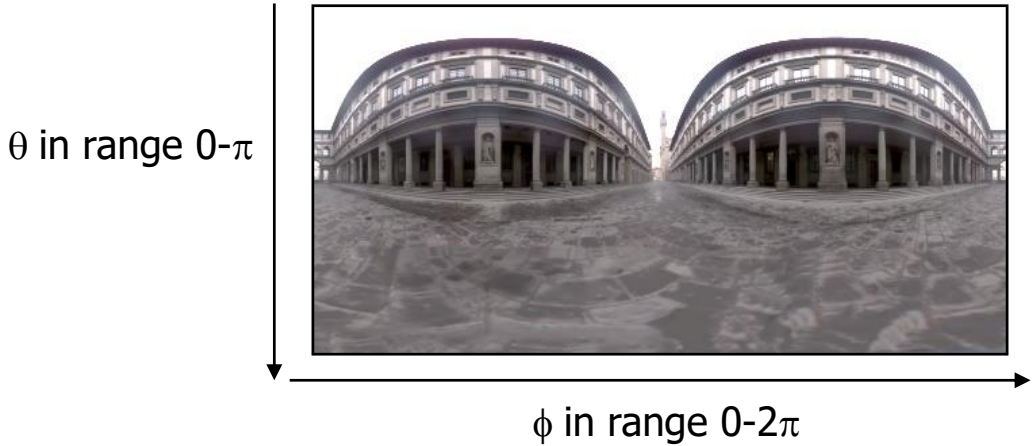
HDR Environment Maps (Light Probe)

- Paul Debevec, http://ict.debevec.org/~debevec/Probes/



Environment Map Formats

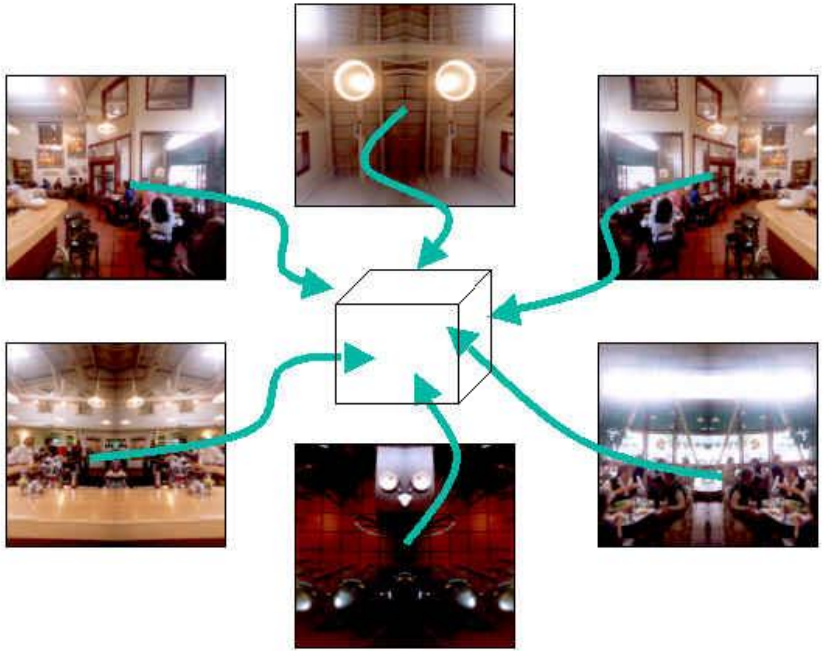
Angular map



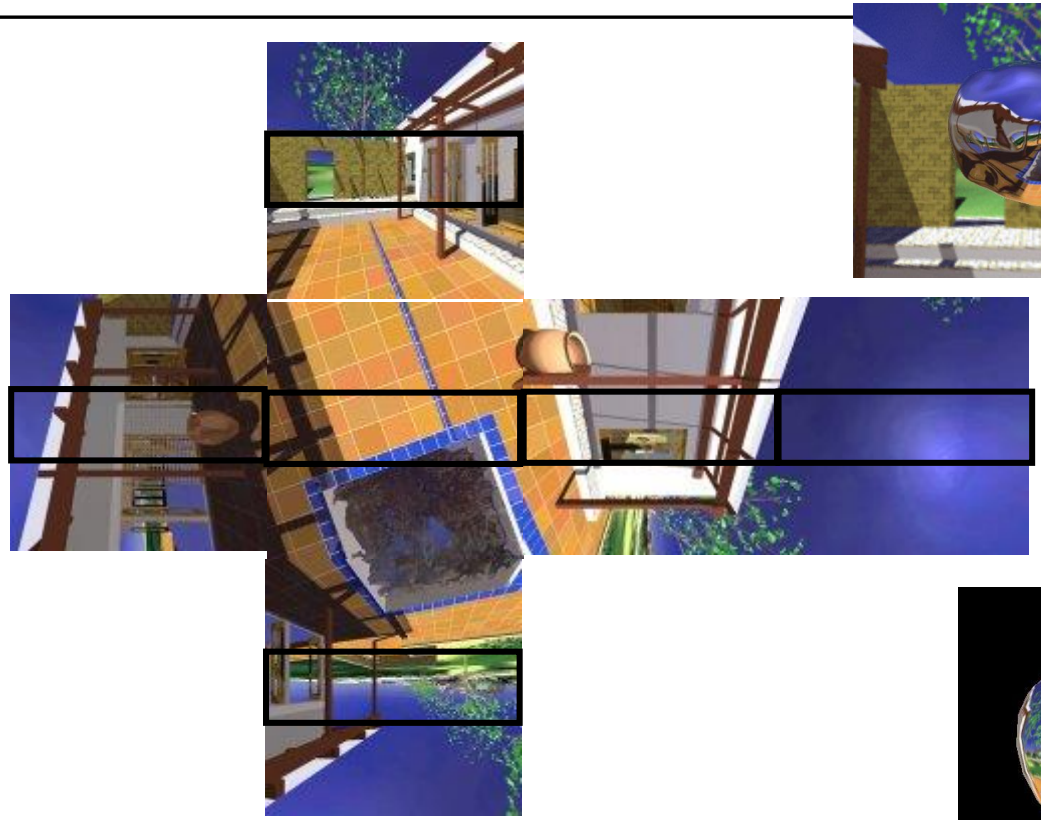
Longitude-latitude format

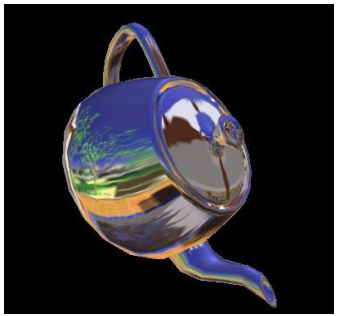
Cubemaps

- Green 1986



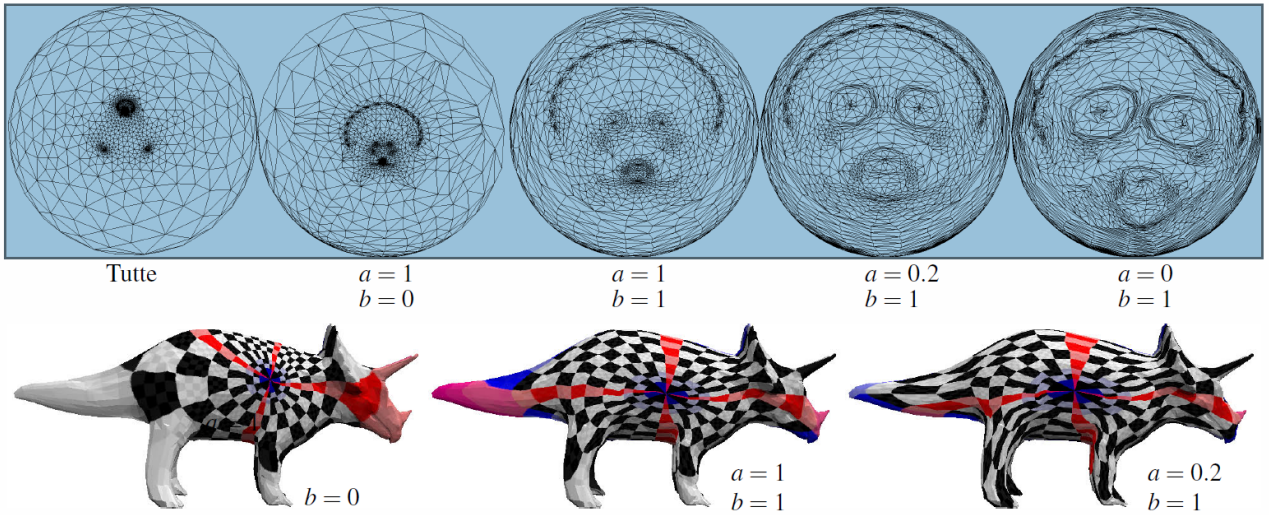
Cube Maps – Real Time Update



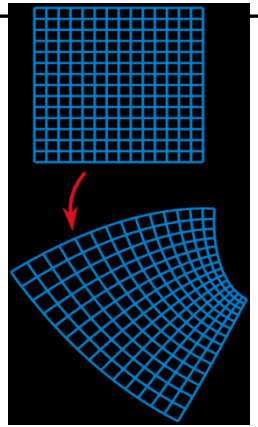


Mesh UV Parametrization

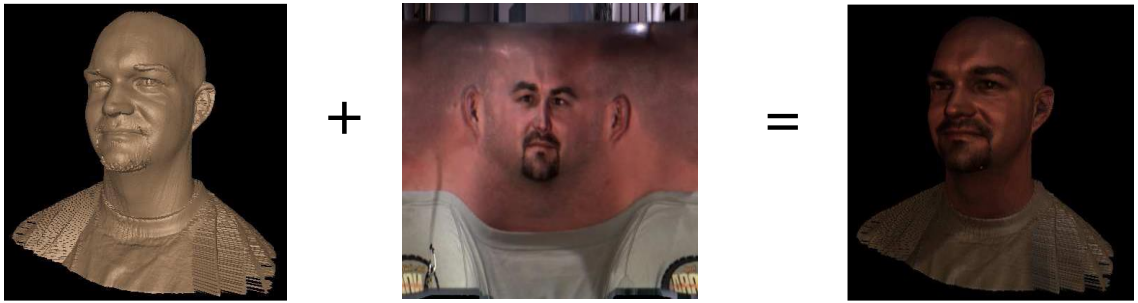
- General UV assignment methods
- Algorithmic parametrization (unwrap -> UV map)
- Editing UV map



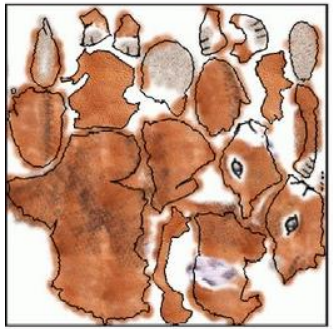
Texture Mapping Problems

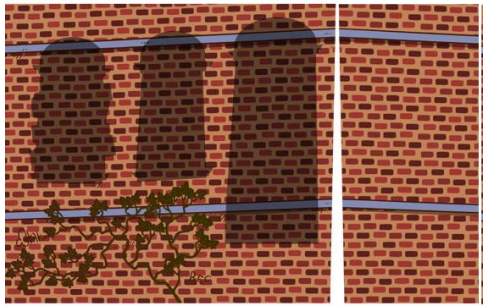


- Mapping from R^3 to R^2
  - Area preserving mapping
  - Conformal mapping (keeps angles)
- Discontinuities (seams)
  - Vertex duplication
  - Boundaries around seems in the texture, limited MIP-mapping
  - Minimization, placing to less visible areas



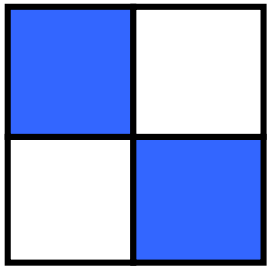
Texture Atlas



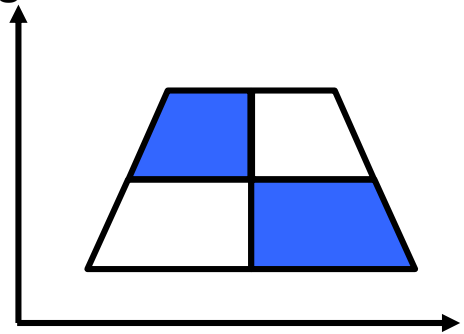


Perspectivelly Correct Texture Mapping

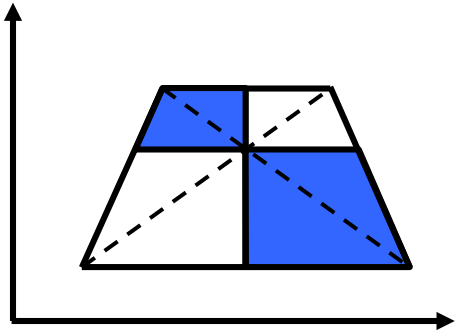
- Rasterization
  - Interpolating u,v coordinates



texture



linear interpolation



correct interpolation

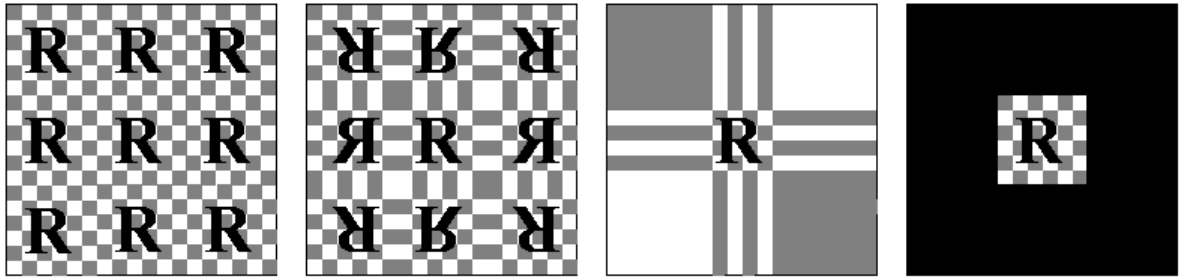
- Note: Ray Tracing
  - Resolved implicitly
  - Using barycentric coords resulting from ray/tri intersection

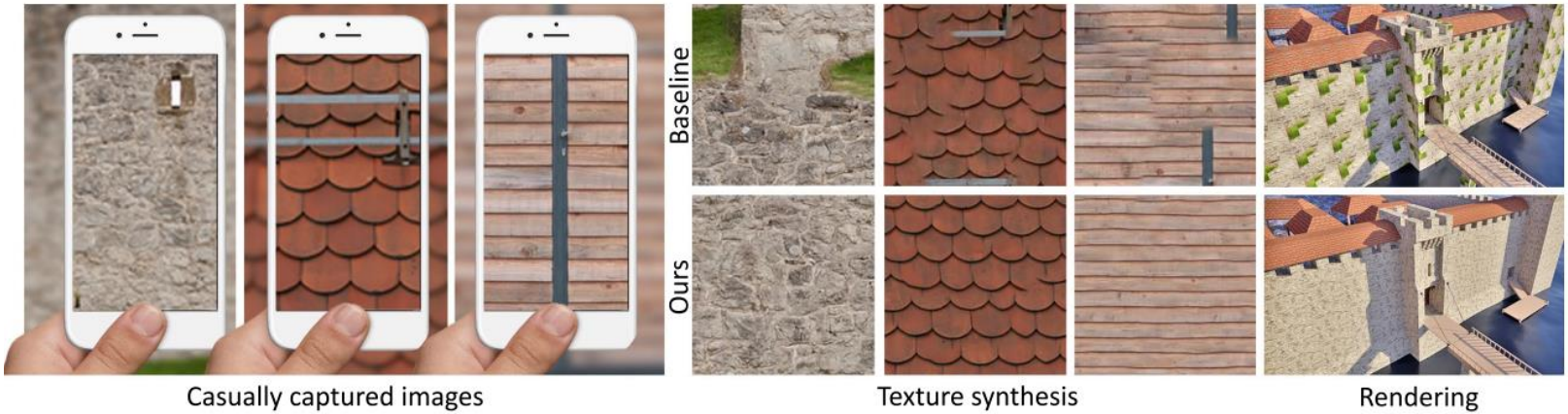
Perspectivelly Correct Texture Mapping

- For each vertex compute u'=u/w, v'=u/w, w'=1/w
  - recall that for perspective w ~ z
- Bilinear interpolation of u', v', w'
- For each fragment u''=u'/w', v''=v'/w'

Texture Expansion

- wrap, repeat
- mirror
- clamp
- border





Moritz et al. Texture Stationarization: Turning Photos into Tileable Textures

Textures

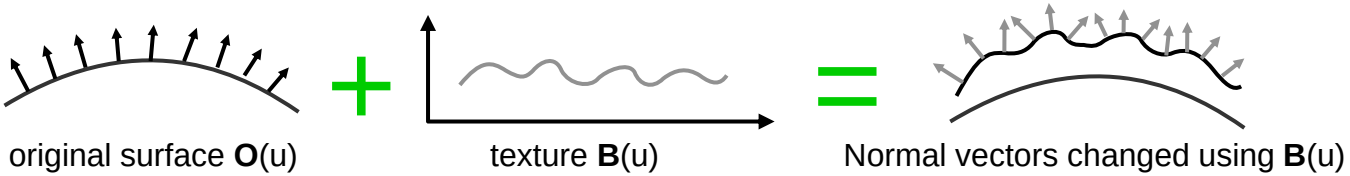
- Motivation - What are textures good for? MPG 13
- Texture mapping principles
- Using textures in rendering
- Summary

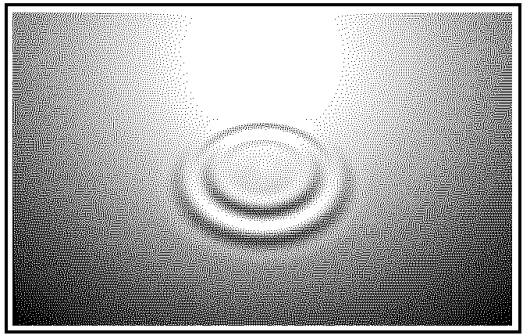
Modulation: What does the texture modify?

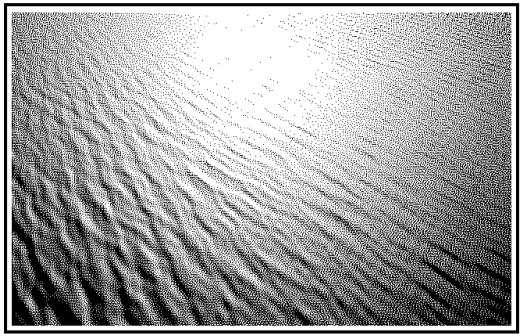
- Color (color mapping, gloss mapping)
- Normals (*bump mapping*)
- Incomming light (*reflection mapping, environment mapping)*
- Surface shape (*displacement mapping*)
- Transparency (*alpha mapping*)

Bump Mapping

- Input: grayscale image => normals using derivation
- Input: color image => directly encoded normals

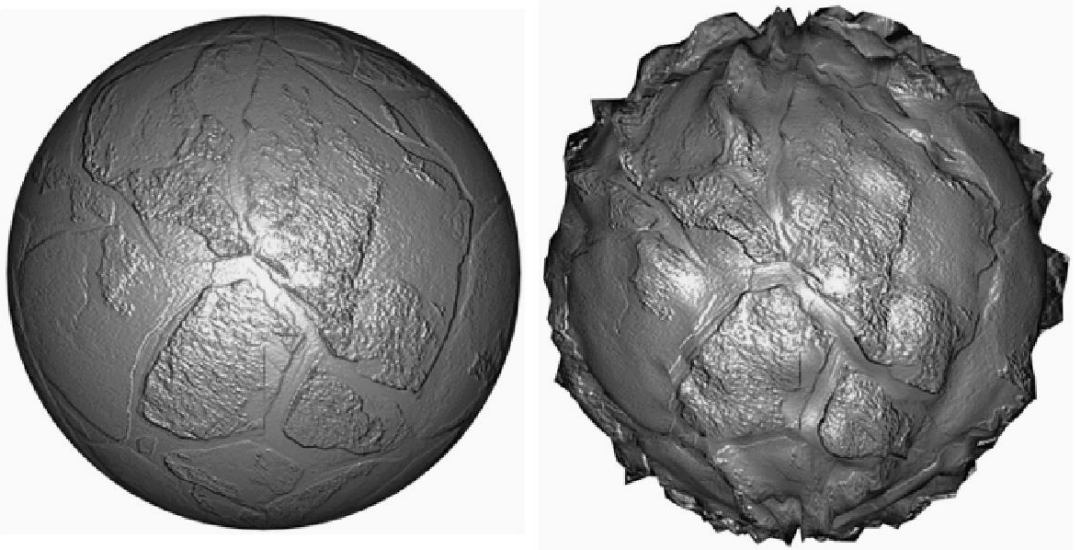




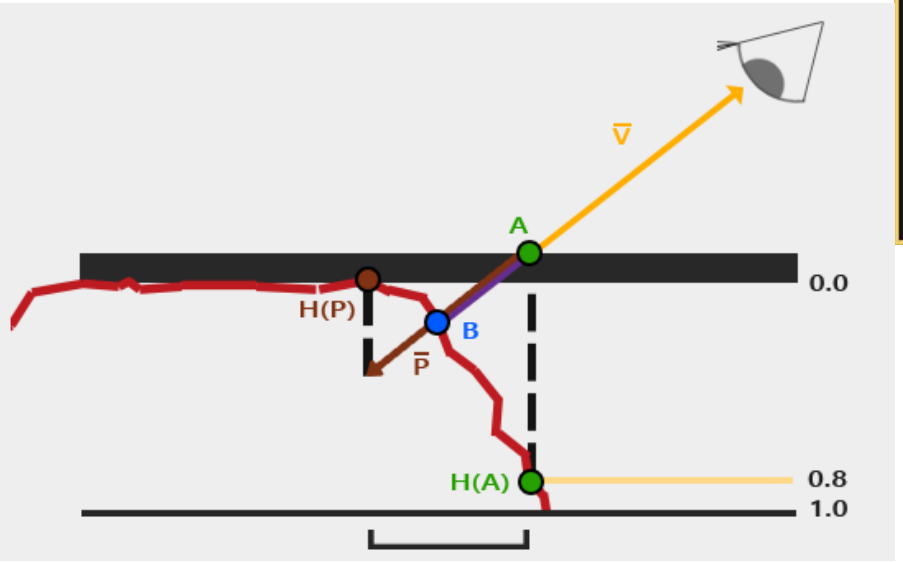


Displacement Mapping

- Surface geometry shifted



Paralax Mapping




```
vec2 ParallaxMapping(vec2 texCoords, vec3 viewDir) {
 float height = texture(depthMap, texCoords).r;
 vec2 p = viewDir.xy / viewDir.z * (height * height_scale);
 return texCoords - p;
}
```

- Parallax occlusion mapping
  - Vícekrokové hledání průsečíku
  - Offset průměr z posledních dvou kroků

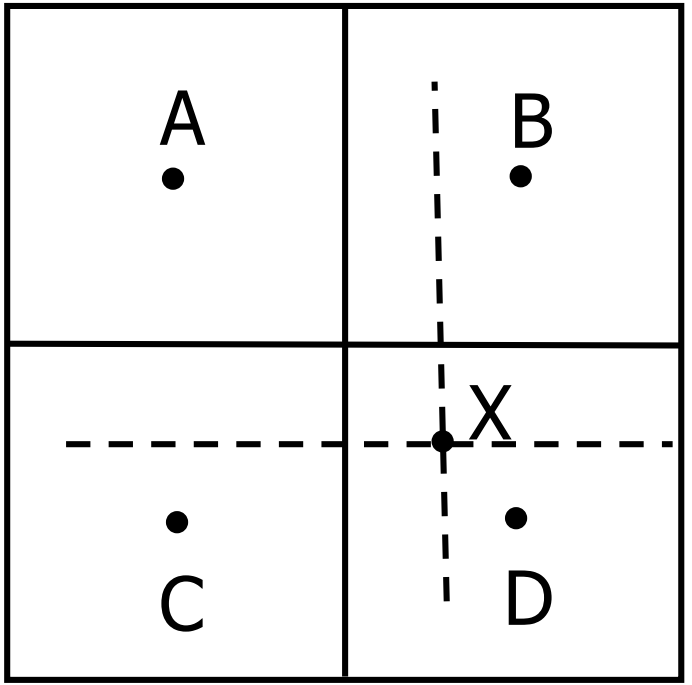
Brawley and Tatarchuk. ShaderX3, 2005

Texture Filtering

- Magnification
  - one texel projects to more pixels
- Minification
  - more texels on one pixel

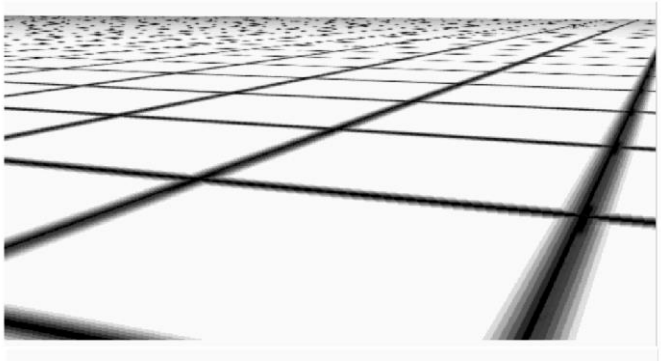
Texture Filtering - Magnification

- Nearest neighbor
- Bilinear interpolation
- Bicubic (Hermite) interpolation

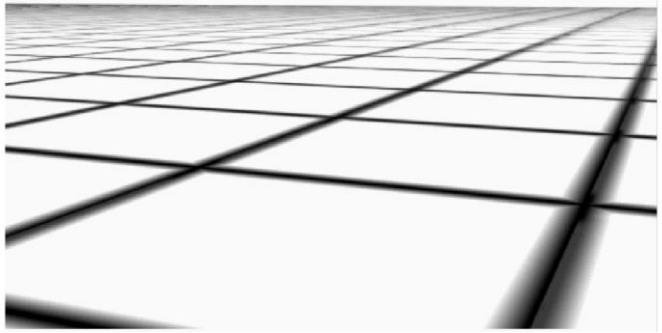


Texture Filtering - Minification

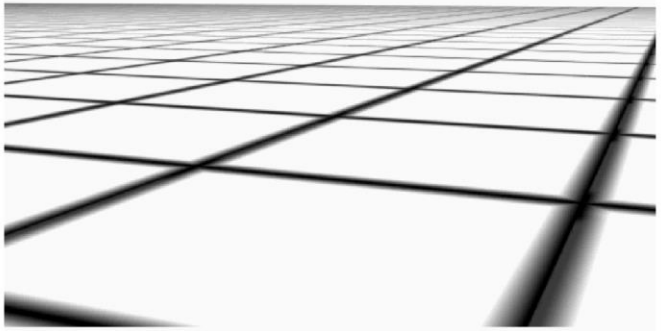
Nearest neighbor



Mipmap



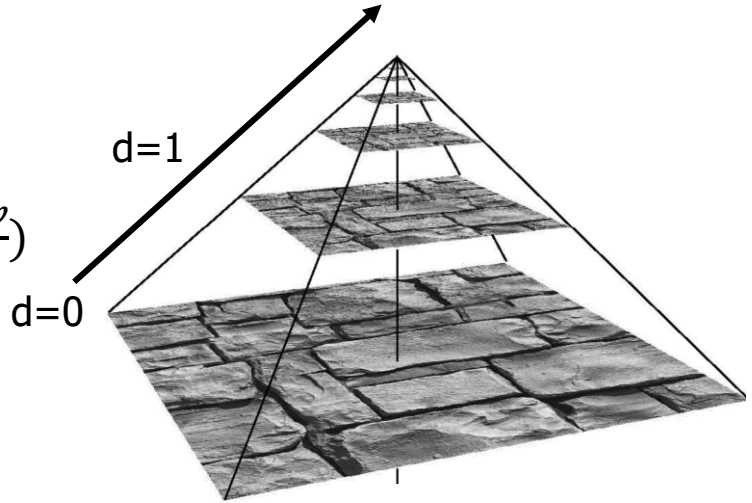
Summed area table



Minification – Mip Mapping

- *Mipmapping*: more resolutions in single image
  - Precomputed / on the fly
  - 33% more memory

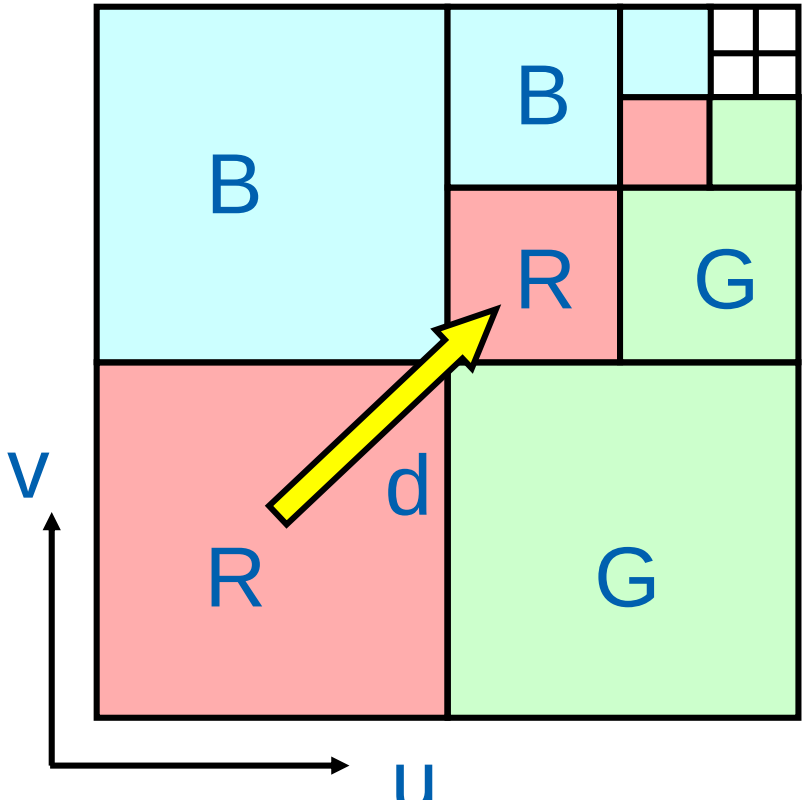
- Using MipMap
  - Mipmap level d based on distance $(\frac{\delta uv}{\delta x}, \frac{\delta uv}{\delta y})$
  - Resolution 2^k x 2^k
  - Trilinear interpolation




```
float mip_map_level(in vec2 texture_coordinate) {
        // The OpenGL Graphics System: A Specification 4.2 // - chapter 3.9.11, equation 3.21
        vec2 dx_vtc = dFdx(texture_coordinate); vec2 dy_vtc = dFdy(texture_coordinate);
        float delta_max_sqr = max(dot(dx_vtc, dx_vtc), dot(dy_vtc, dy_vtc));
        return 0.5 * log2(delta_max_sqr); // == log2(sqrt(delta_max_sqr));
}
```

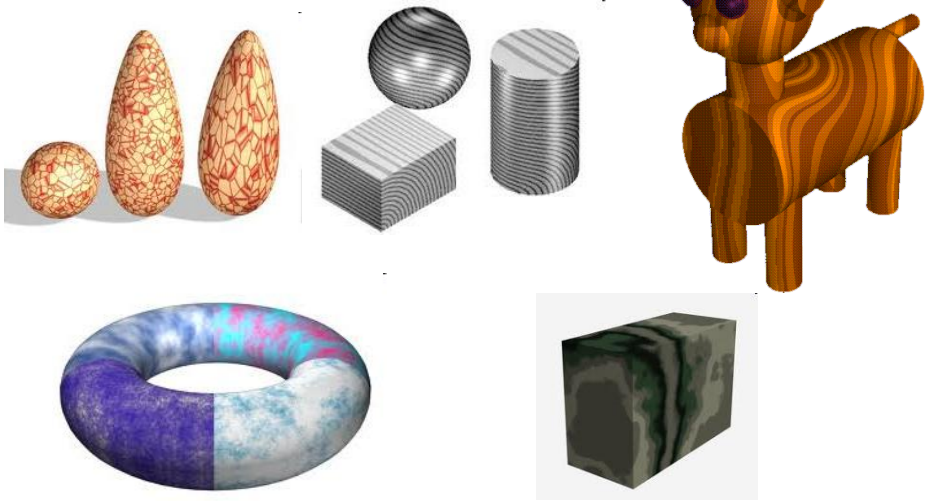
Mipmap Storage (on Disk)

- Storage of RGB mipmap in grayscale image



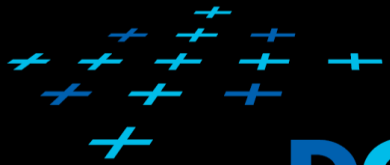
3D Textures

- 3D grid or function
  - Captures interior material (wood, marble, …)
- Direct mapping from 3D to texture coordinates
  - Easier than for 2D textures!



Textures

- Motivation - What are textures good for? MPG 13
- Texture mapping principles
- Using textures in rendering
- Summary



DCGI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

Questions?

Look Back to History

- 1974 idea of texture mapping (Catmull/Williams)
- 1976 env. mapping (Blinn/Newell)
- 1978 bump mapping (Blinn)
- 1983 mipmap (Williams)
- 1984 illumination map (Miller/Hoffman)
- 1985 procedural 3D texture (Perlin)
- …