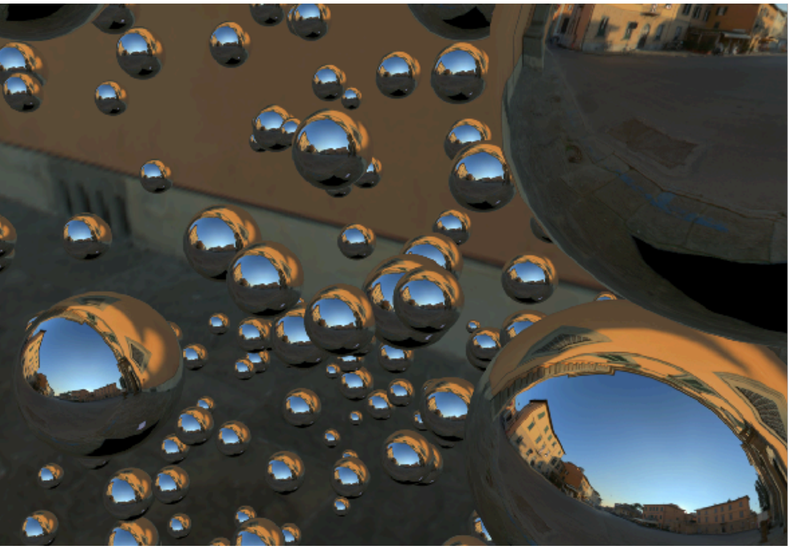
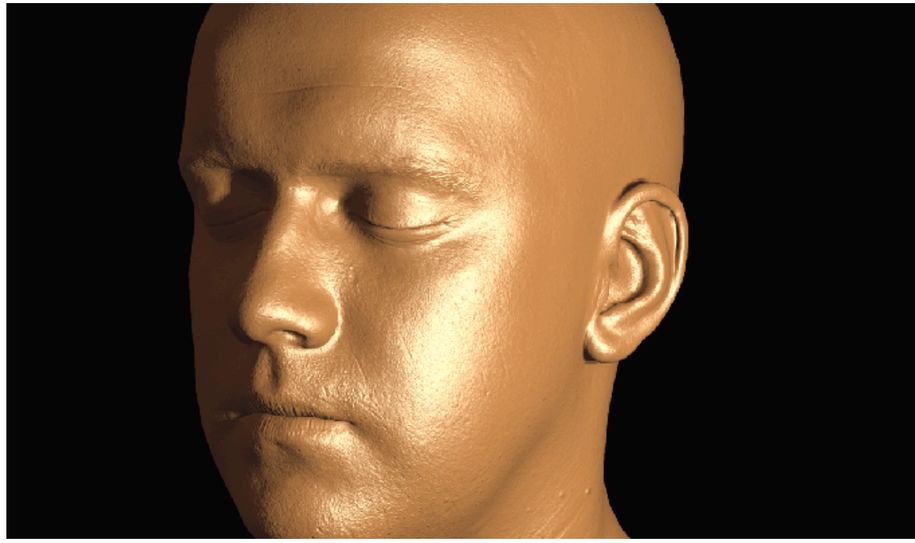
COSC342: Computer Graphics

2017



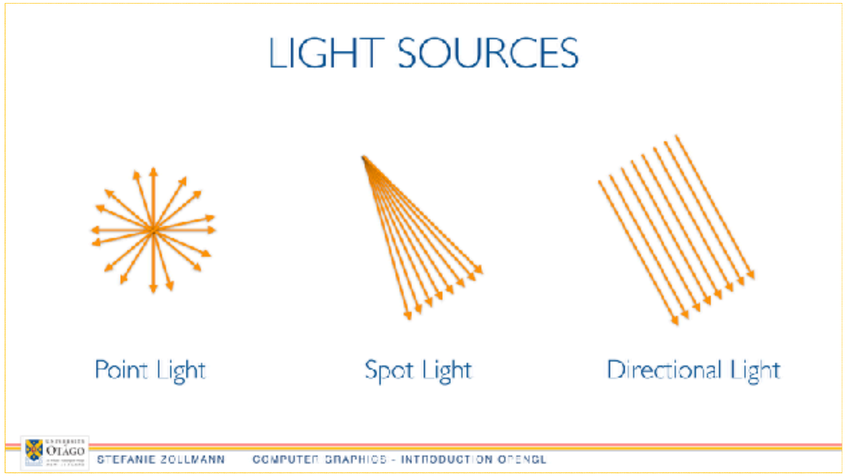


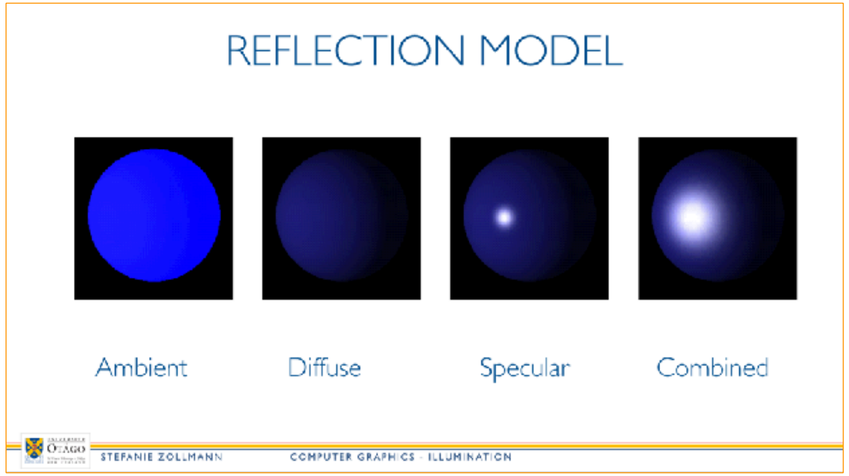
Lecture 17

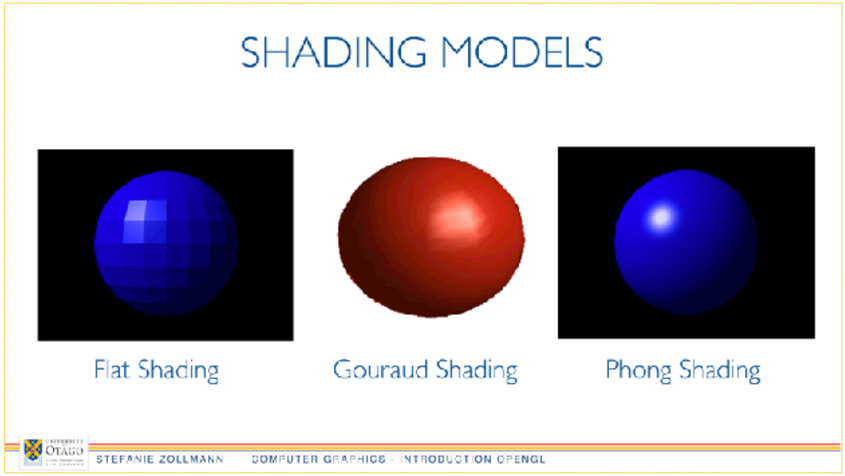
TEXTURE MAPPING

Stefanie Zollmann

LAST LECTURE





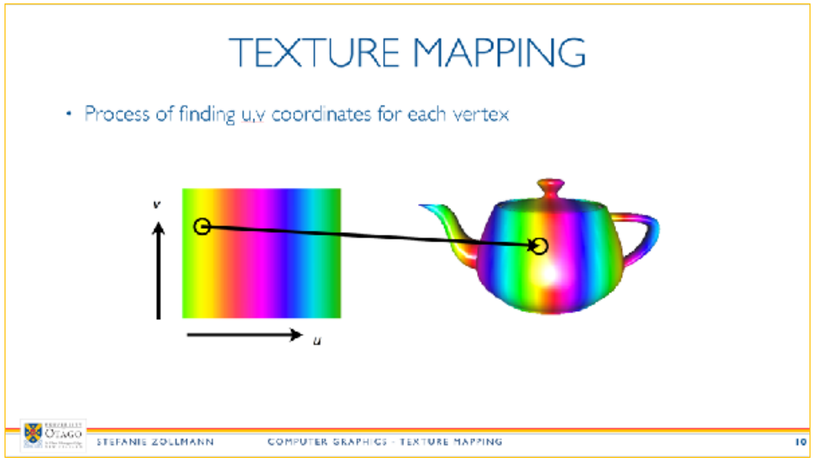


Light Sources

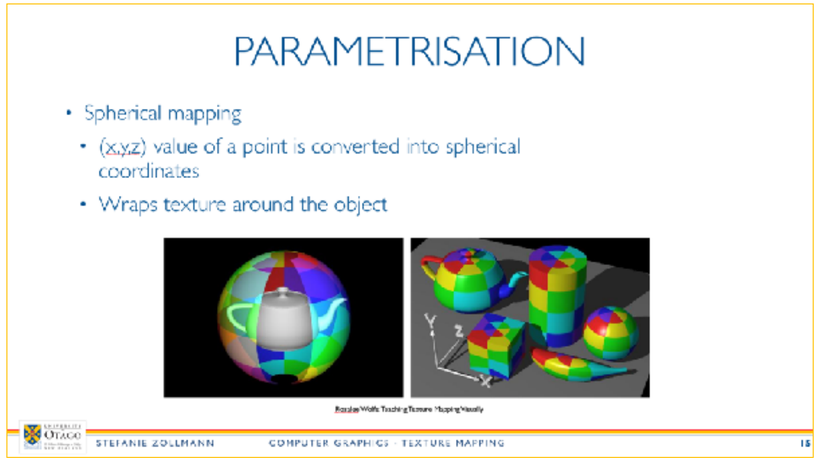
Reflection Model

Shading

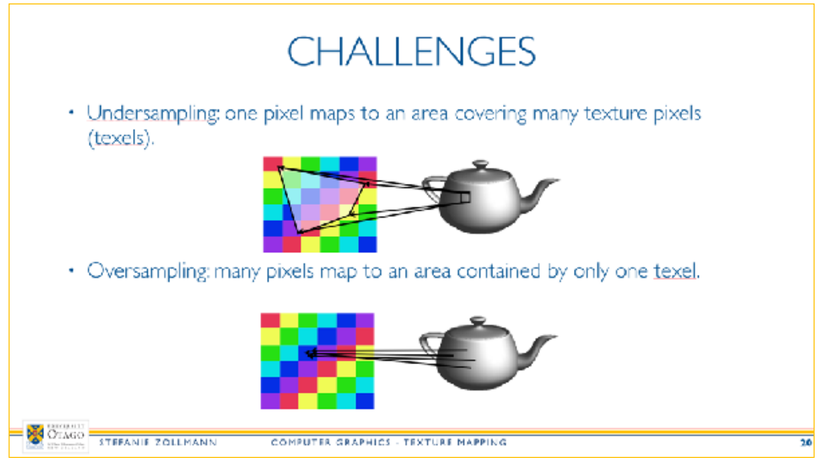
TODAY



Texture Mapping



Parametrisation



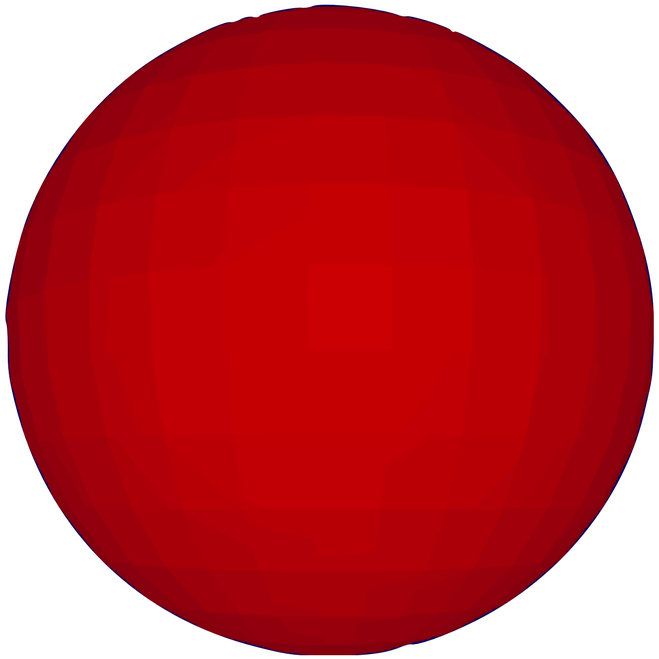
Challenges

TERMINOLOGY

- **Texture**: Array of values:
  - 2D (most common), 1D, 3D
  - Colour, depth, alpha, normals
- **Texels** (texture elements): Elements of the texture map
- **Texture coordinates:** Textures are defined in texture (u, v) coordinates
  - Coordinates u and v ranging from 0.0 - 1.0
  - Think of it as a continuous image

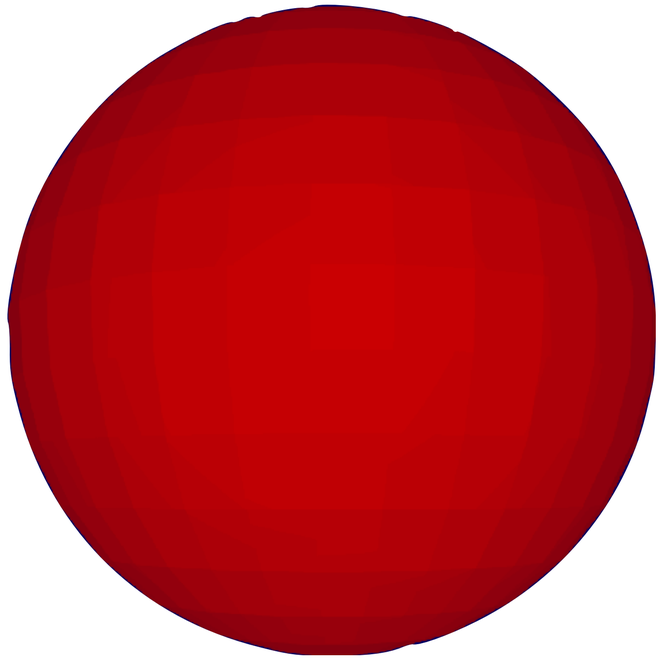
WHY TEXTURING?

- Enhance visual appearance of plain surfaces and objects by applying texture details
- Two spheres with identical geometry
- The earth model appears to involve more complex materials for the objects in the scene



WHY TEXTURING?

- Detail is expensive to model
- Often the surface of an object is viewed only from a distance
- We can 'paint' the detail on the object instead of increasing the complexity of the object



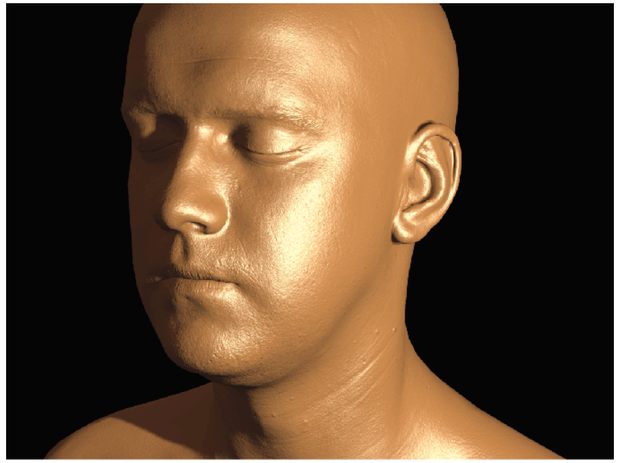
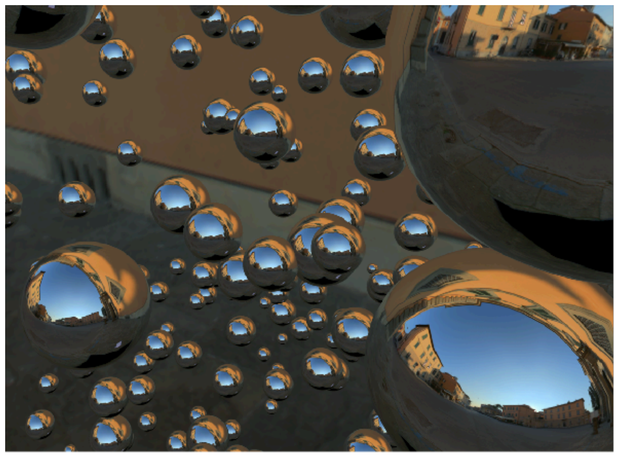
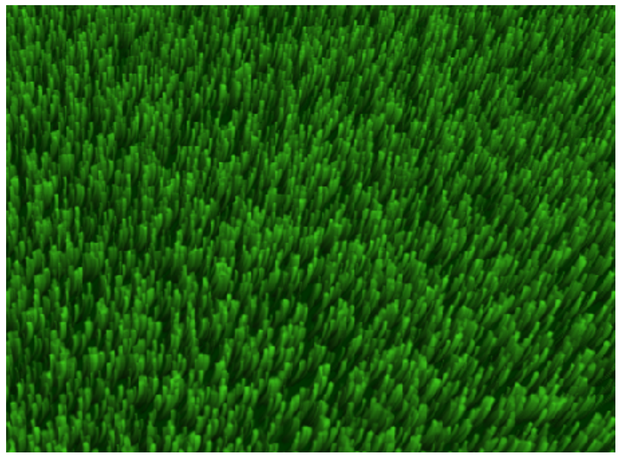
ANOTHER TEXTURE MAPPING EXAMPLE

Image from the Blender Guru tutorial at http://www.blenderguru.com/tutorials/introduction-to-texture-nodes/

TEXTURING

Simulation of different material properties:

- Color
- Reflection
- Gloss
- Transparency
- Bumpiness

HOW?

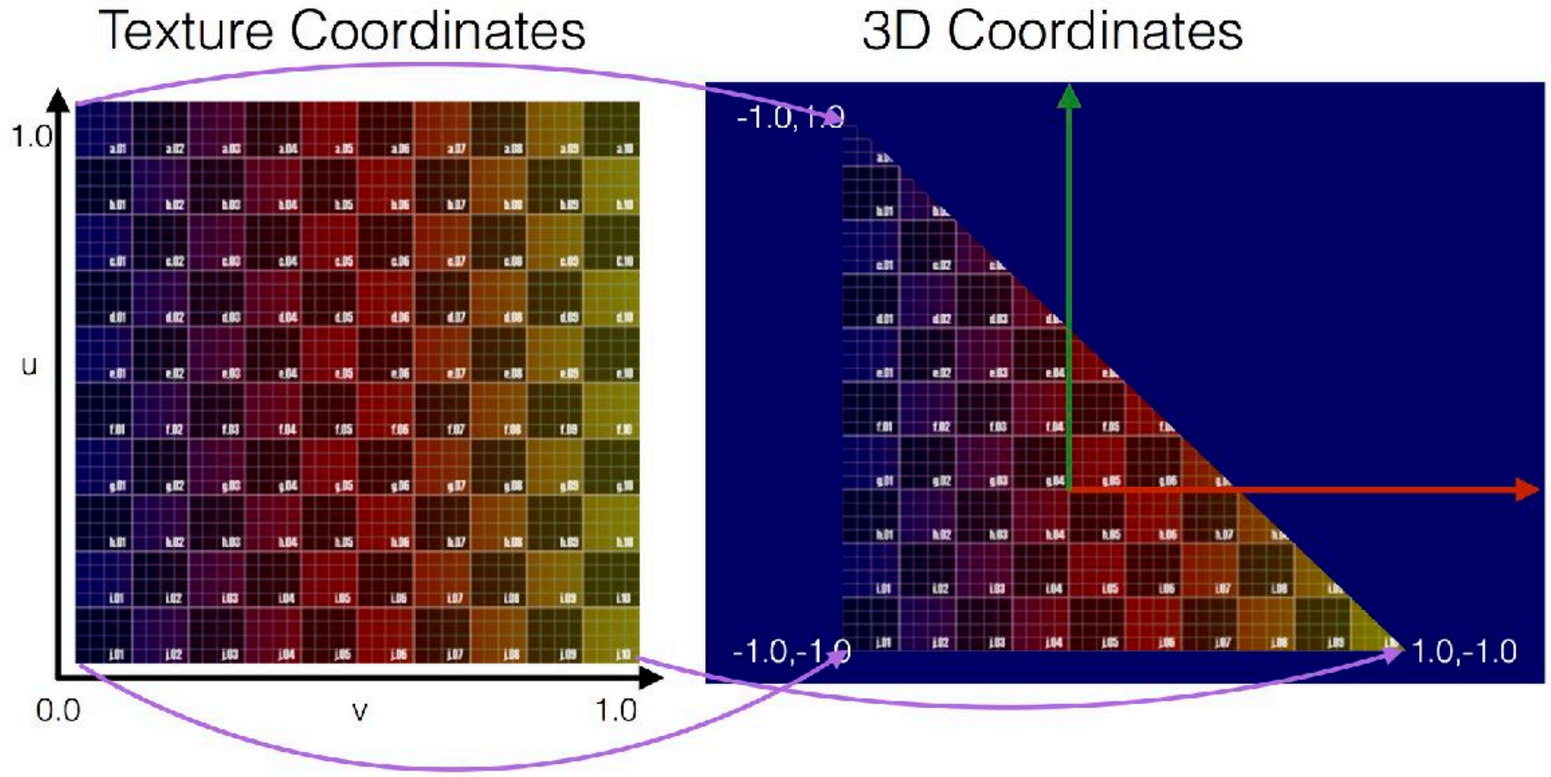
- When drawing fragments, we use the local colour from the texture image
- Usually an image
- Can hold arbitrary information:
  - Interpreted by shader
  - Basis for most advanced rendering effects
- We need a coordinate system on the surface to find the right part of the texture

```
vec3 textureVal = texture( myTexture, UV ).rgb;
```

OBJECT OR WORLD COORDINATES?

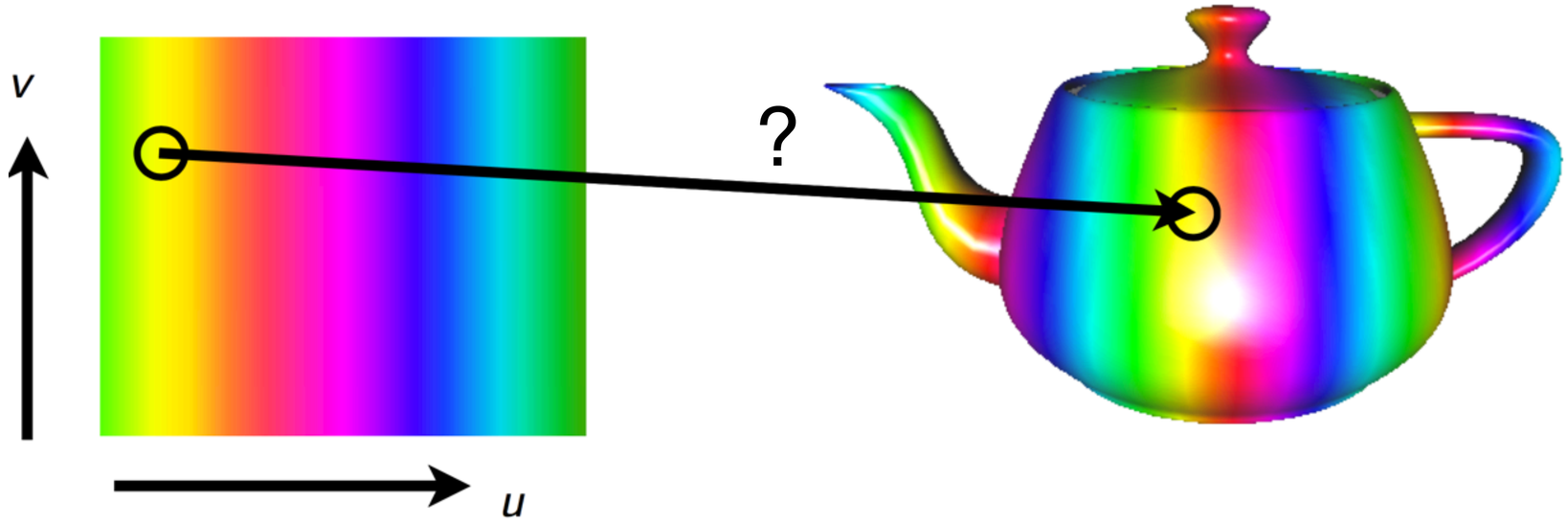
TEXTURE MAPPING

- Process of finding u,v coordinates for each vertex
- Texture is defined in a normalised space (e.g. for 2Dtextures:(u,v)∈[0...1,0...1]

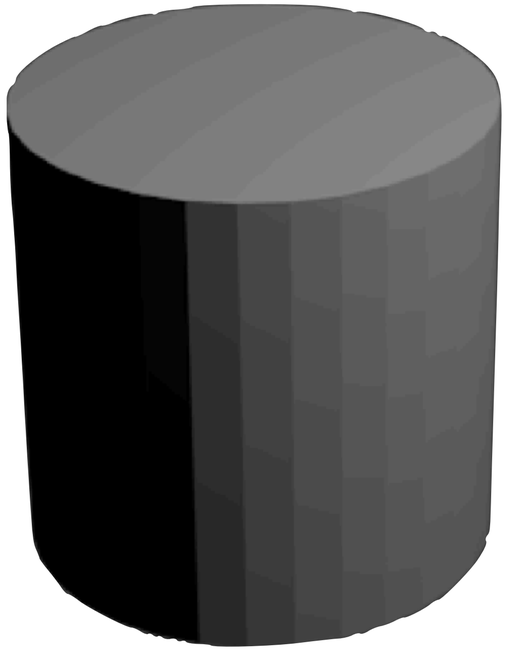


TEXTURE MAPPING

- Often objects are not that simple



PARAMETERISATION

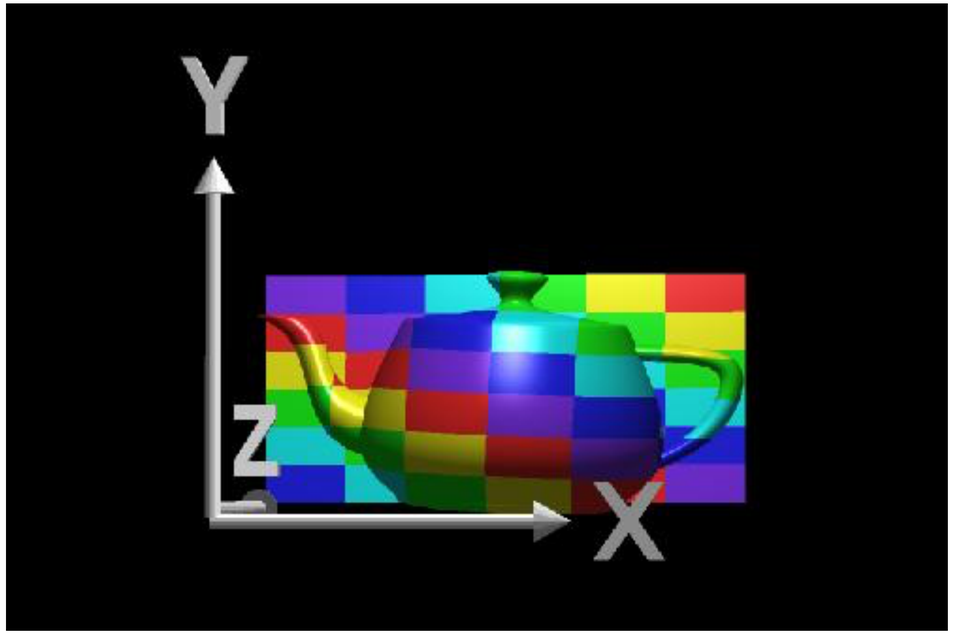


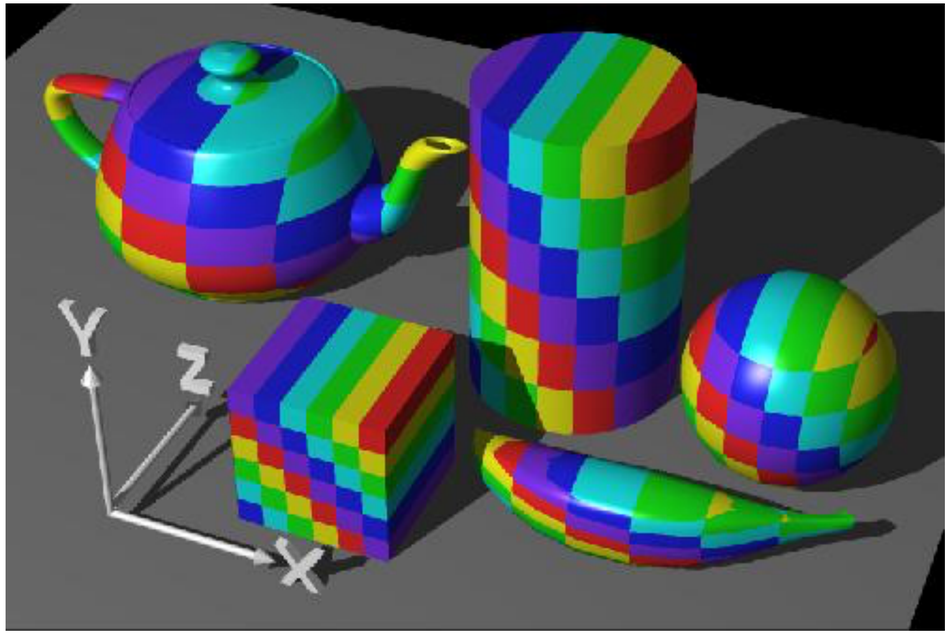
+

=

PARAMETRISATION

- Planar mapping
  - Eliminates one axis (z) and use x and y directly: (u,v)=(x,y)
  - Looks only good from front

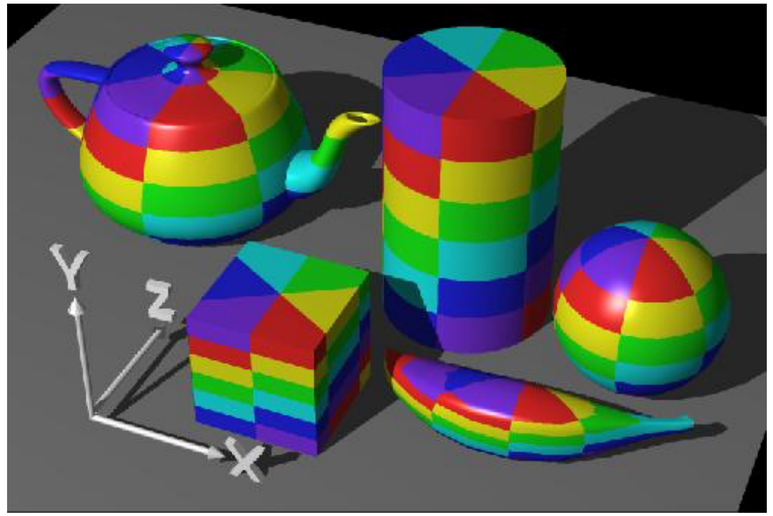




Rosalee Wolfe: Teaching Texture Mapping Visually

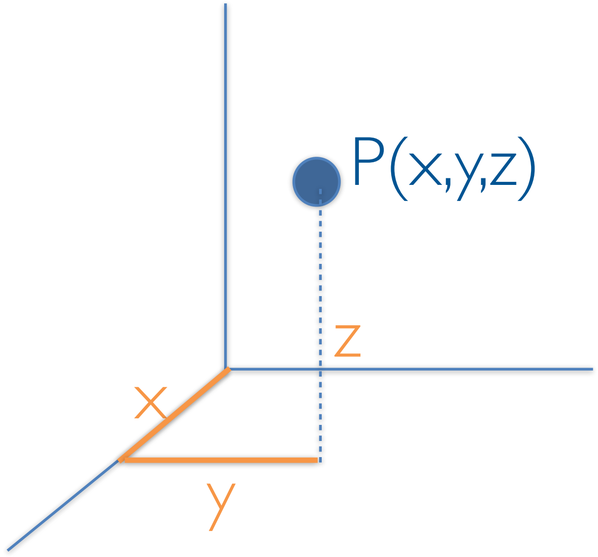
PARAMETRISATION

- Cylindrical mapping
  - Compute angles between vertex and object centre
  - Converts point P(x,y,z) to cylindrical coordinates P(r, θ,z)
  - Wraps texture around the object

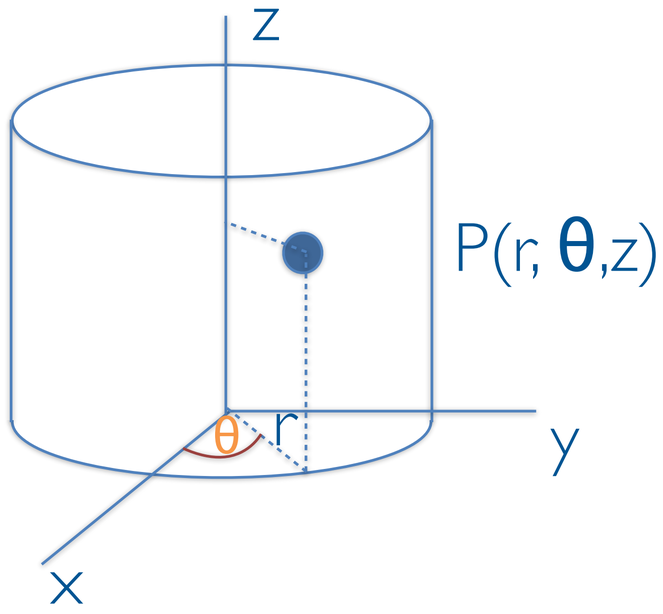


Rosalee Wolfe: Teaching Texture Mapping Visually

PARAMETRISATION

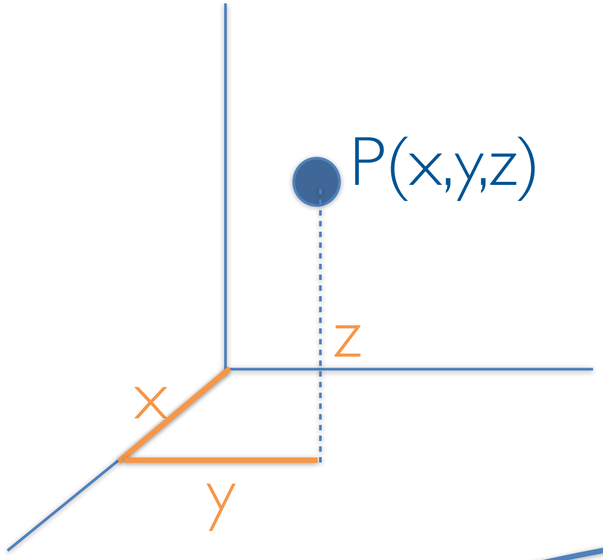


Cartesian Coordinates



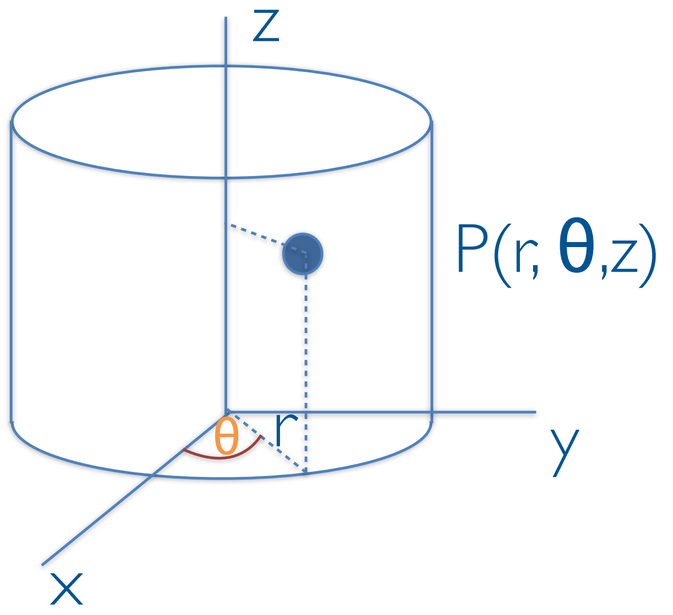
Cylindrical Coordinates

PARAMETRISATION



Conversion:

- $x = r \cdot \cos \theta$,
- $y = r \cdot \sin \theta$,
- $z = v$



Cartesian Coordinates

Cylindrical Coordinates

Take care of quadrants:

$\tan \theta = y/x \not\Rightarrow \theta = \tan^{-1}(y/x)$

Conversion:

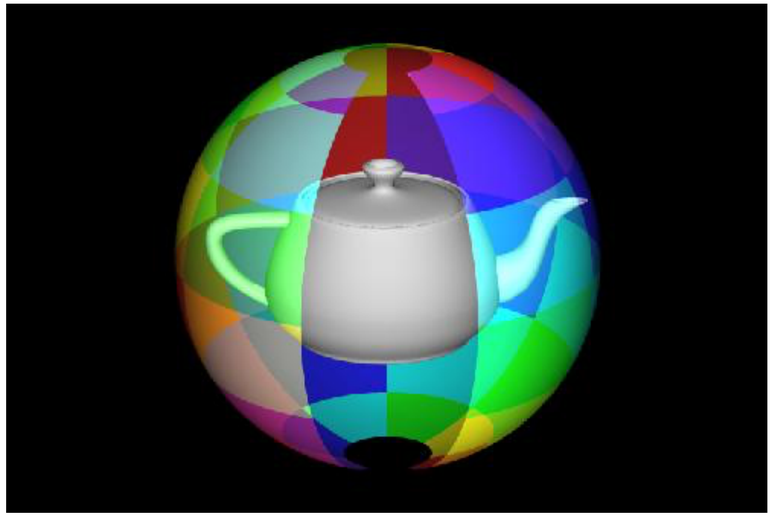
- $u = \theta = \tan^{-1}(y/x)$
- $v = z$

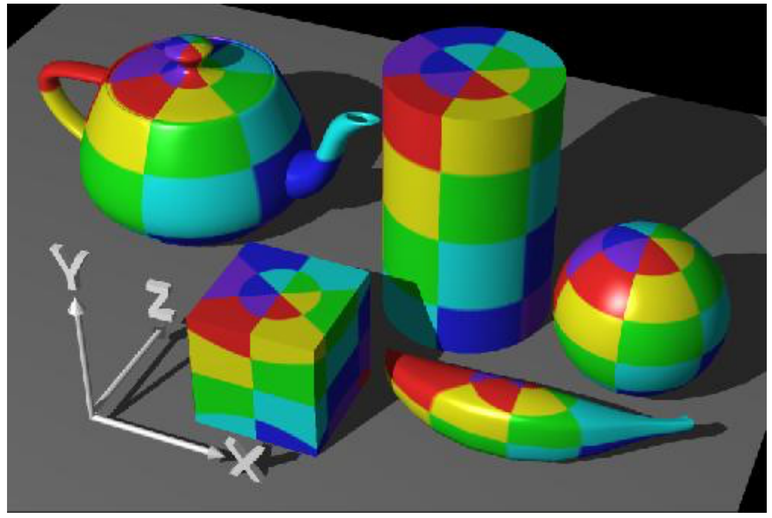
With:

- $u = \theta$, with $0 \leq \theta \leq 2\pi$
- $v = z$ with $0 \leq z \leq 1$

PARAMETRISATION

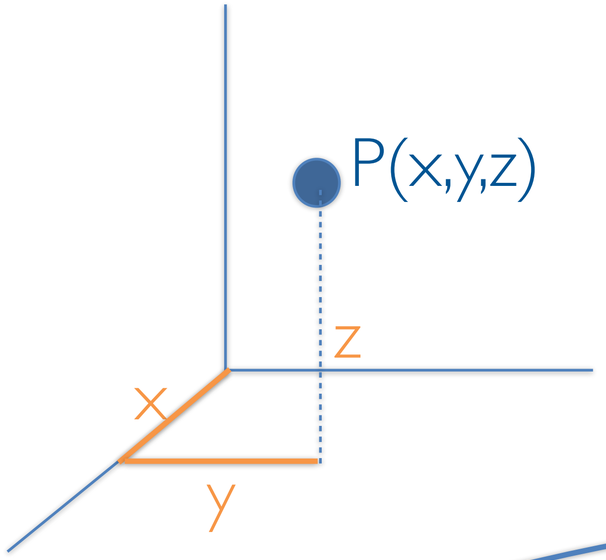
- Spherical mapping
  - p(x,y,z) value of a point is converted into spherical coordinates (theta, phi)
  - Wraps texture around the object





Rosalee Wolfe: Teaching Texture Mapping Visually

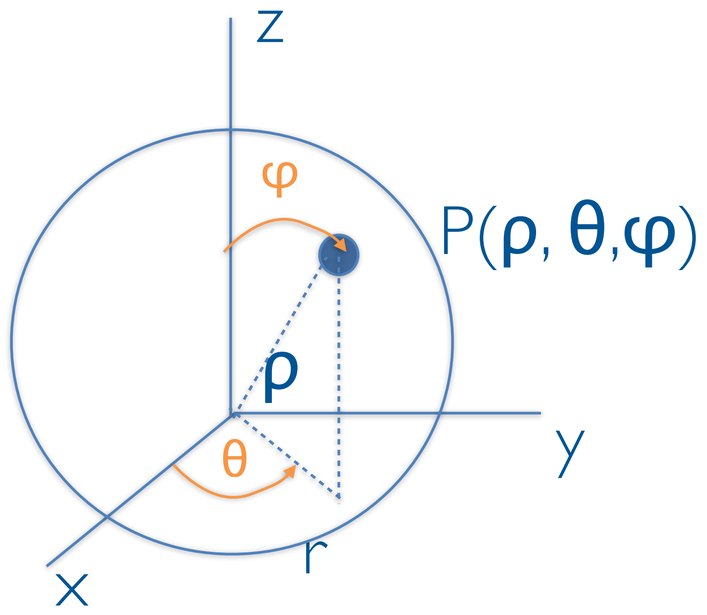
PARAMETRISATION



Cartesian Coordinates

Conversion:

- $x = \rho \sin\varphi \cos\theta$
- $y = \rho \sin\varphi \sin\theta$
- $z = \rho \cos\varphi$



Spherical Coordinates

Conversion:

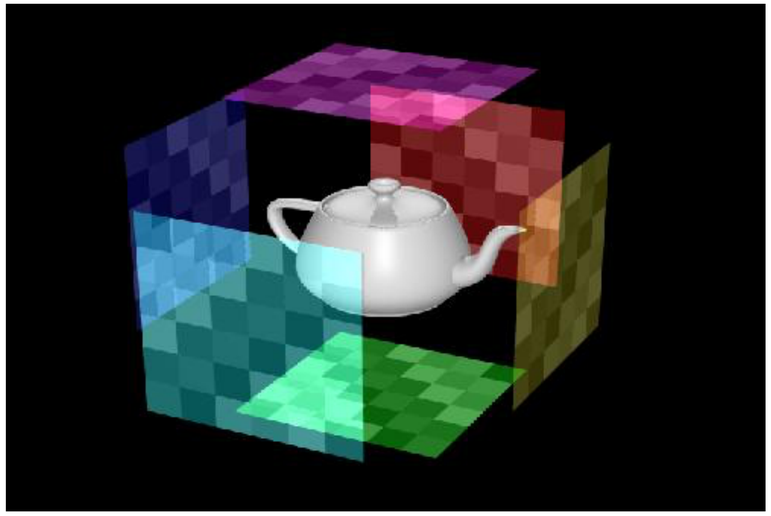
- $u = \theta = \tan^{-1}(y/x)$
- $v = \varphi = \cos^{-1}(z/\rho)$

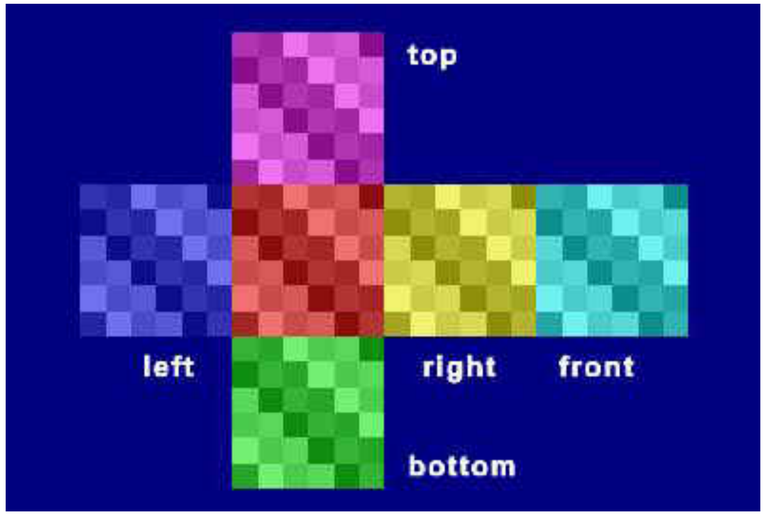
With:

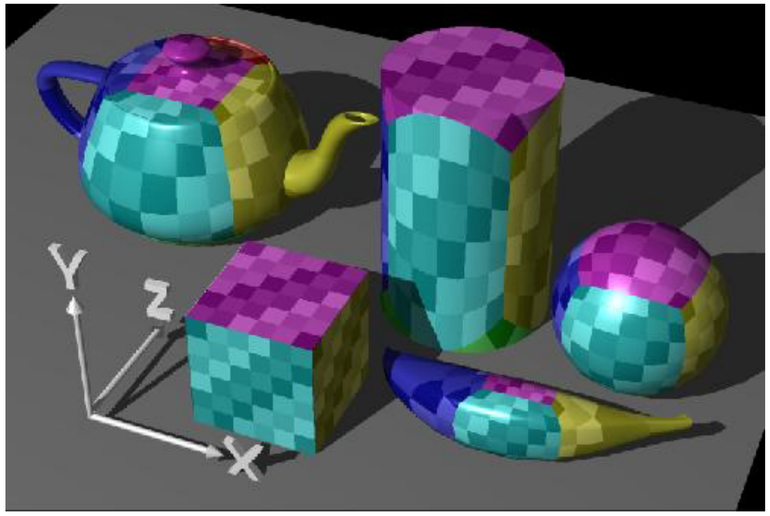
- $u = \theta$, with $0 \leq \theta \leq 2\pi$
- $v = \varphi$ with $0 \leq \varphi \leq 2\pi$

PARAMETRISATION

- Box mapping
  - Similar to planar mapping
  - But uses 6 textures instead of 1
  - Used mainly for environment mapping







Rosalee Wolfe: Teaching Texture Mapping Visually

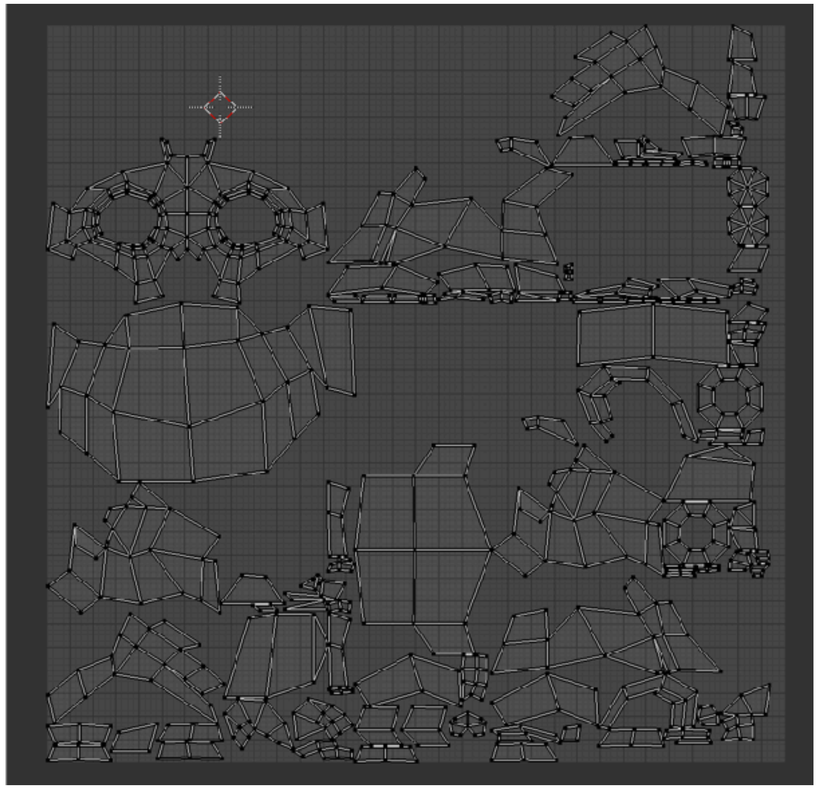
PARAMETRISATION

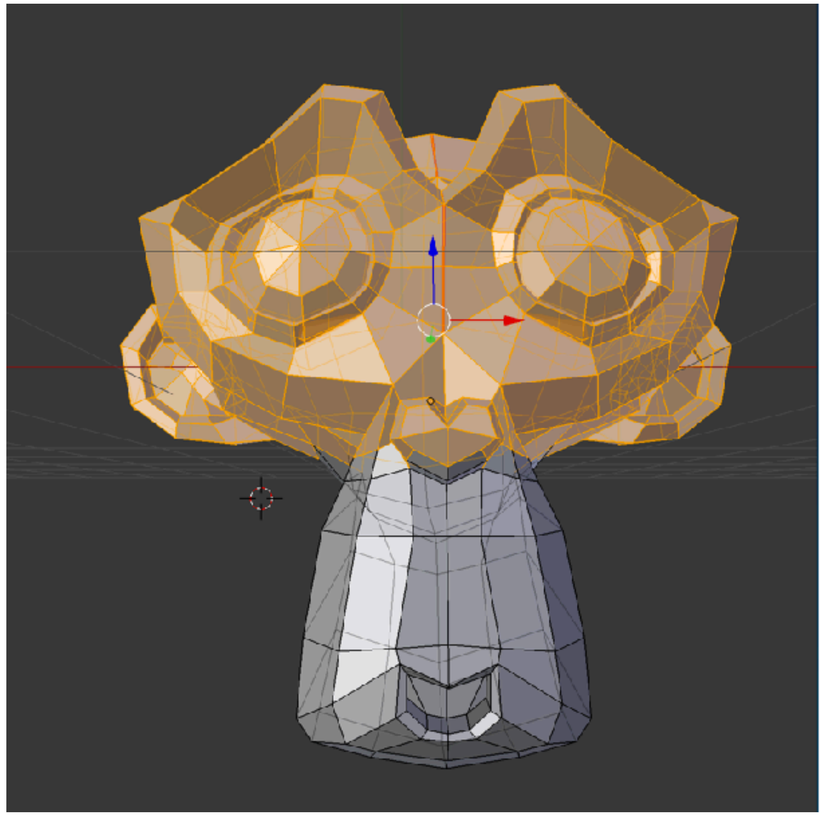
- Box mapping
  - Similar to planar mapping
  - But uses 6 textures instead of 1
  - Used mainly for environment mapping

Rosalee Wolfe: Teaching Texture Mapping Visually

PARAMETRISATION

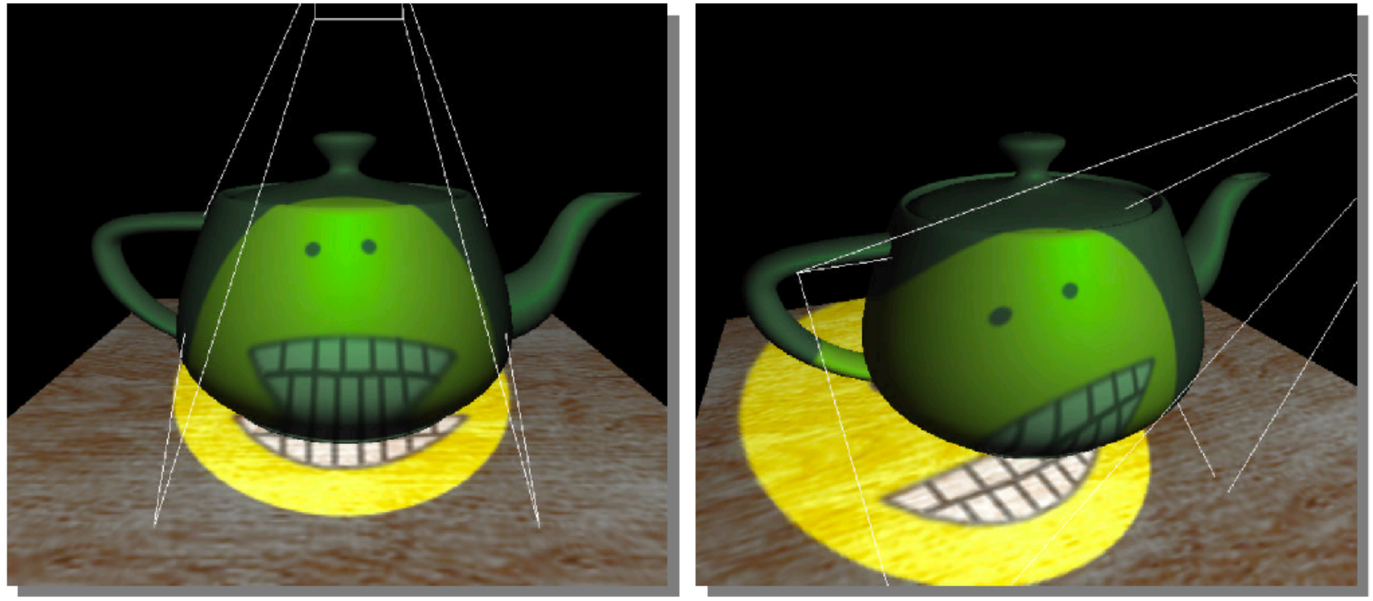
- Manual mapping
  - Unwrap object (e.g. in Blender)





PROJECTIVE TEXTURE MAPPING

- Texture is used as light source
- Like a slide projector
- A good model for shading variations due to illumination
- But also has other use cases



Everitt: Projective Texture Mapping

PROJECTIVE TEXTURE MAPPING

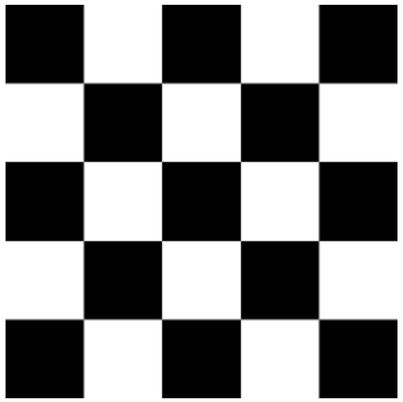
Other use case:

- Create billboard models from photos
- Uses input photos as textures with projective texture mapping

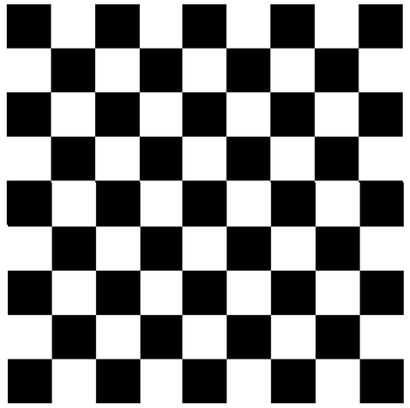
TEXTURE ADDRESSING

- At some point, texture coordinates have to be mapped to texels
- What happens if coordinates are outside of the image?

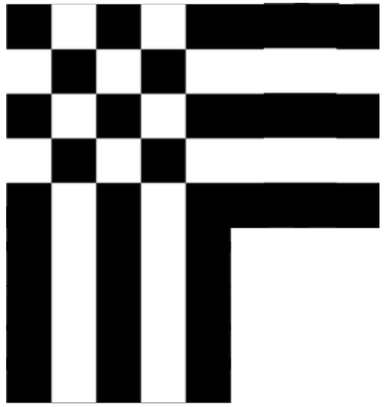
  -> "texture addressing"



Image



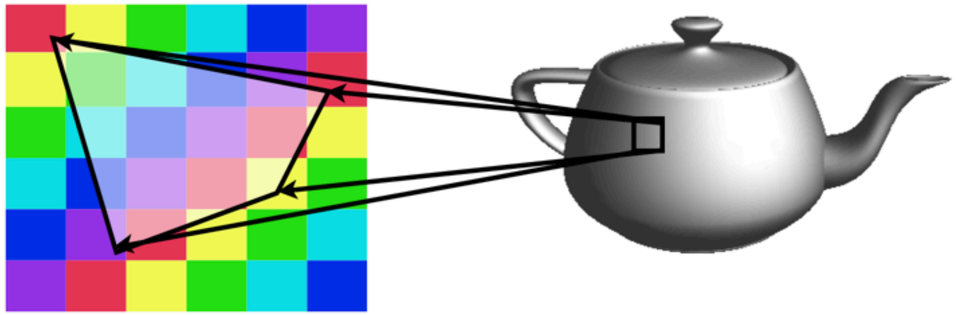
GL_WRAP



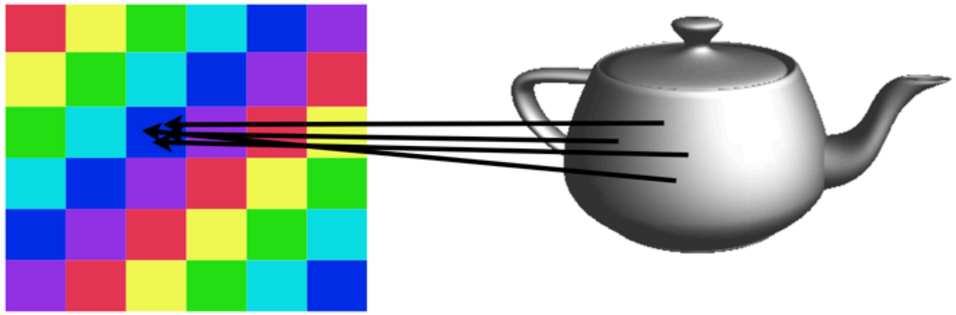
GL_CLAMP

CHALLENGES

- Undersampling: one pixel maps to an area covering many texel (Minification)

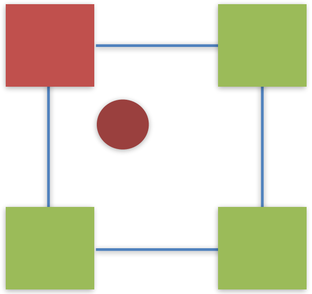


- Oversampling: many pixels map to an area contained by only one texel (Magnification)

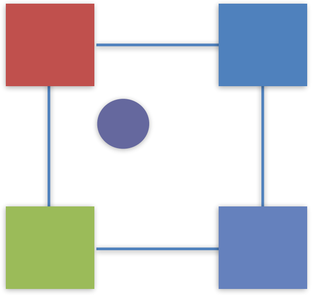


TEXTURE FILTERING

- Interpolate texel value from neighbours



Nearest Neighbour
(lower quality)

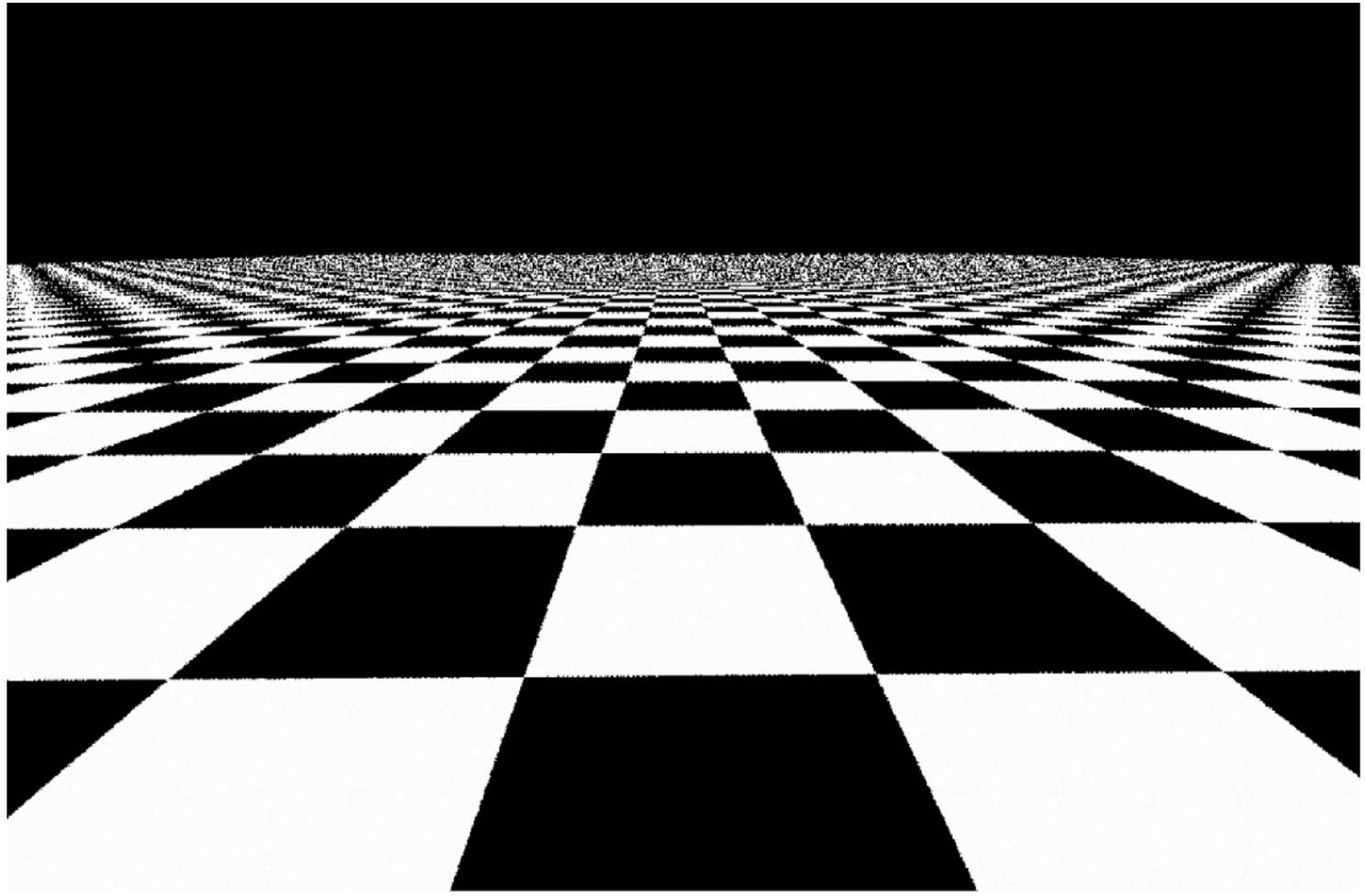


Linear interpolate the neighbours
(better quality, slower)

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
```

TEXTURE FILTERING

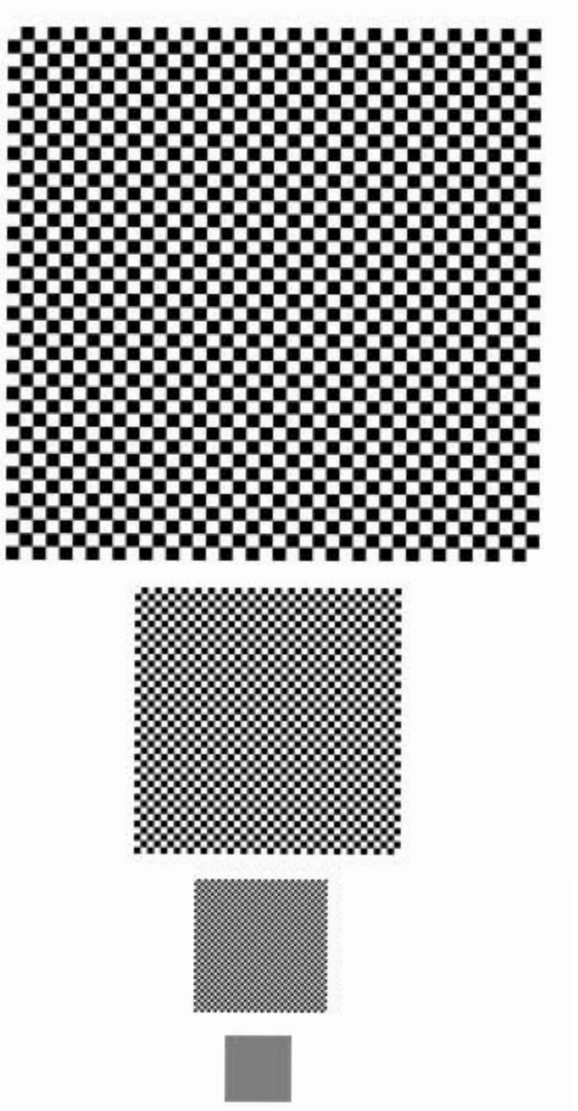
MIPMAPS



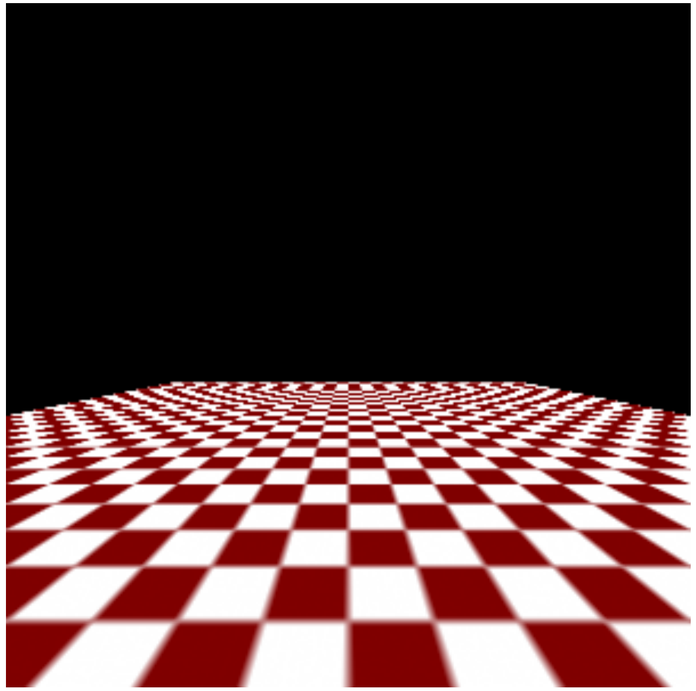
Problem of undersampling

MIPMAPS

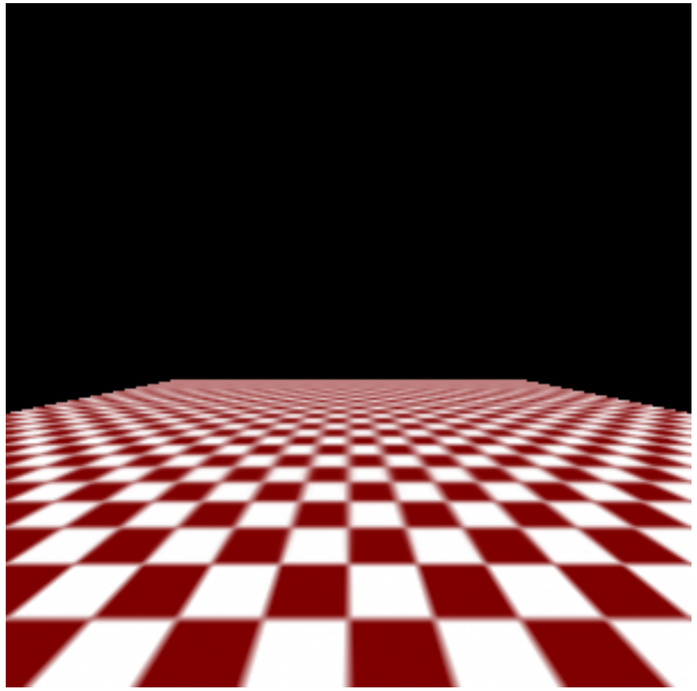
- Start with most detailed texture
- Reduce to half size by combining (averaging) adjacent four texels
- Continue to minimum.
- During rendering choose texture level according to distance



MIPMAPS



No mipmaps



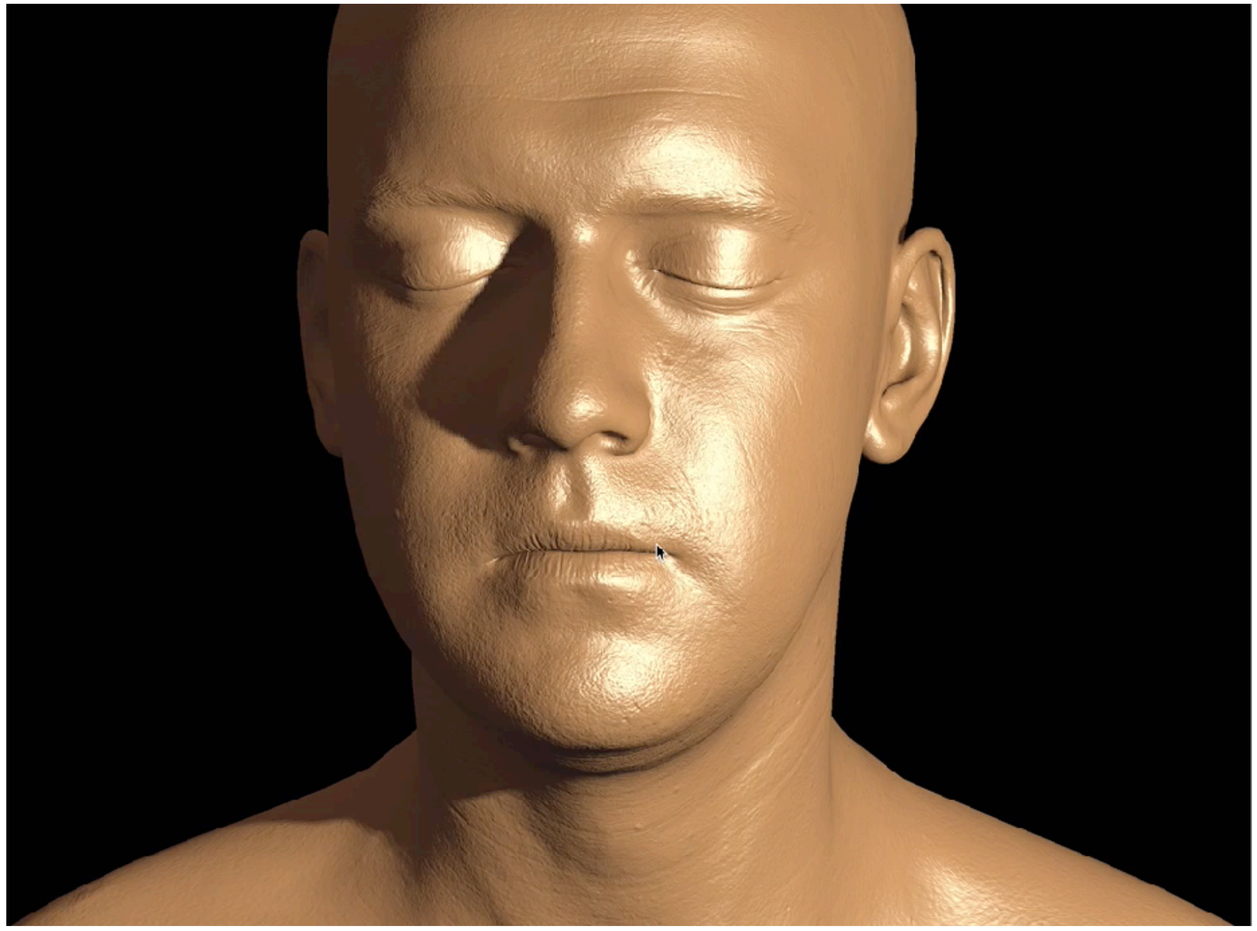
Mipmaps

MIPMAPS


Floor (128x128)
mag: Linear
min: LinearMipMapLinear
Painting (748x600)
mag: Linear
min: Linear
Floor
mag: Nearest
min: NearestMipMapNearestFilter
Painting
mag: Nearest
min: Nearest

APPLICATIONS

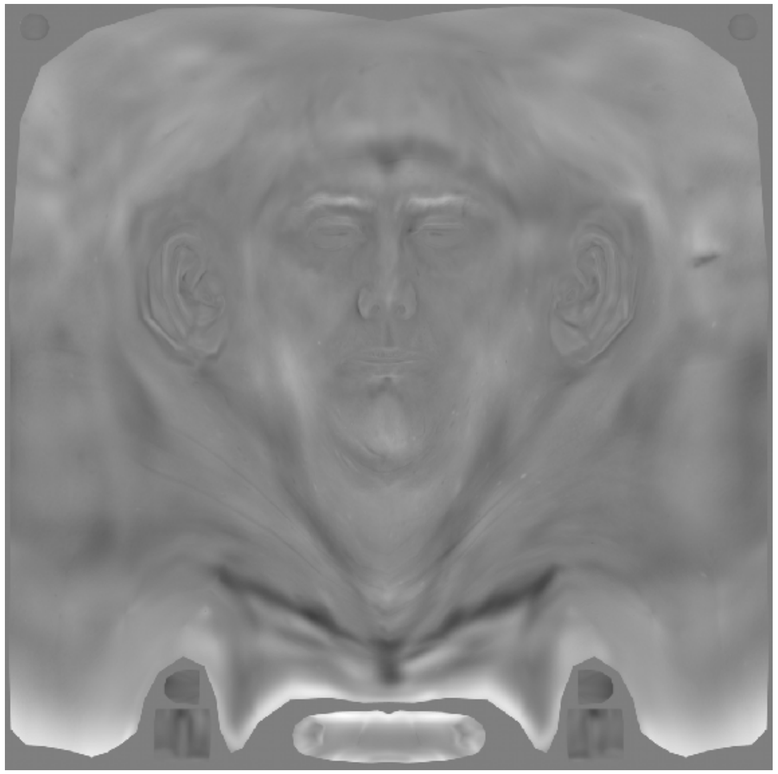
BUMP MAPPING

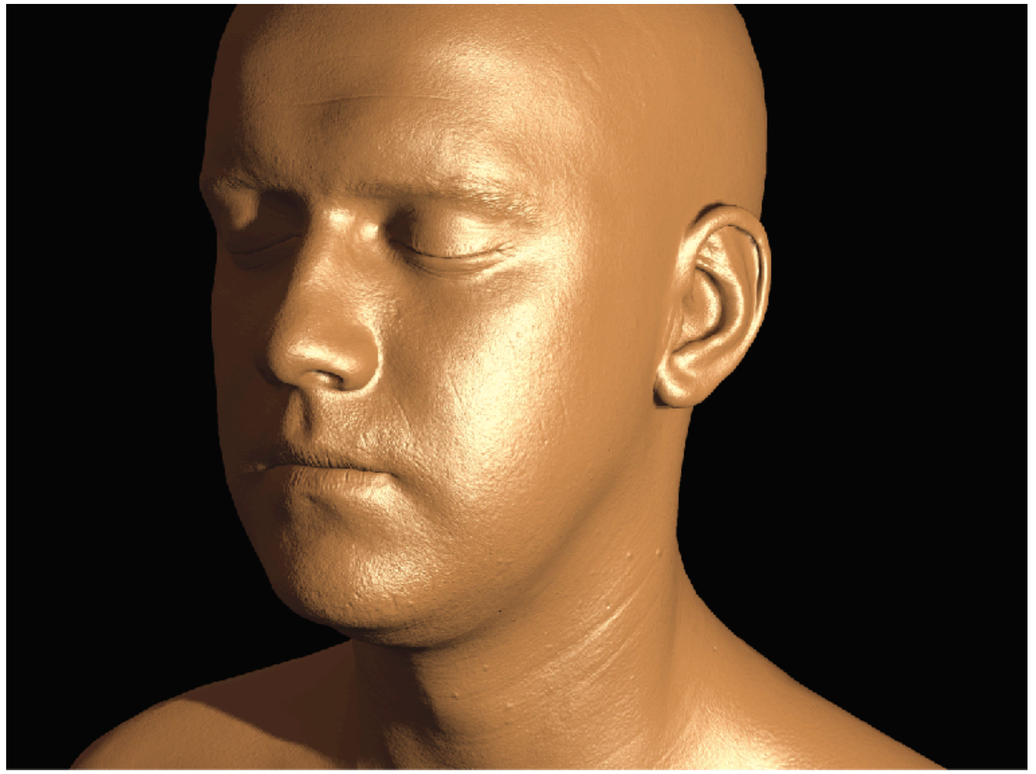


Source: https://github.com/mrdoob/three.js/

BUMP MAPPING

- Create details with changing geometry
- Modifies the normals used for computing illumination

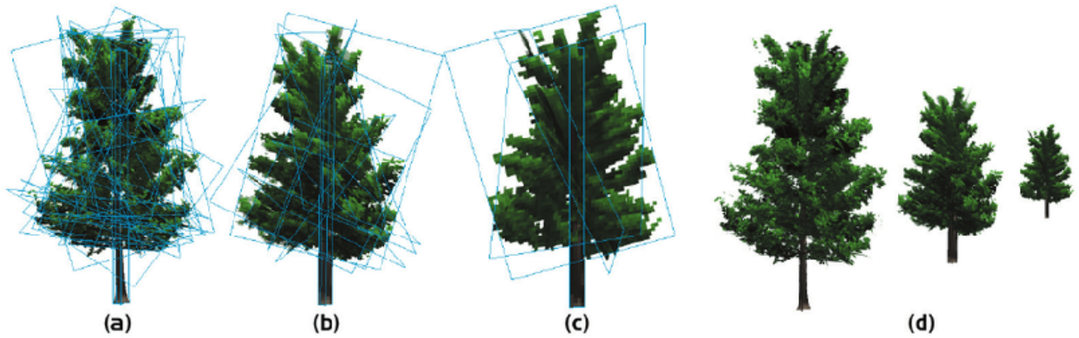




Source: https://github.com/mrdoob/three.js/

ALPHA MAPPING

- Alpha map contains opacity values
- Use for blending between multiple textures
- Creating transparency effects
- Billboard rendering



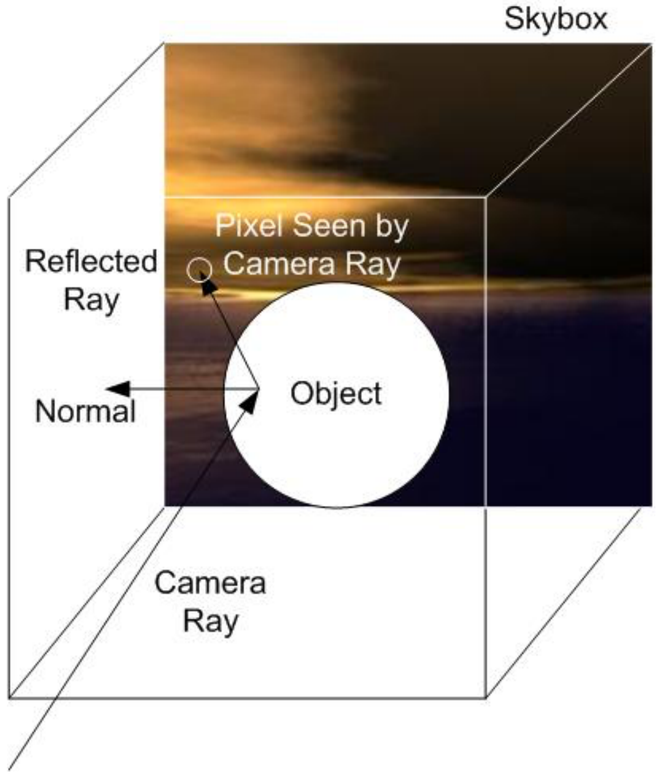
Fuhrman et al.: Extreme Model Simplification for Forest Rendering.

Source: http://jeromeetienne.github.io/

ENVIRONMENT MAPPING

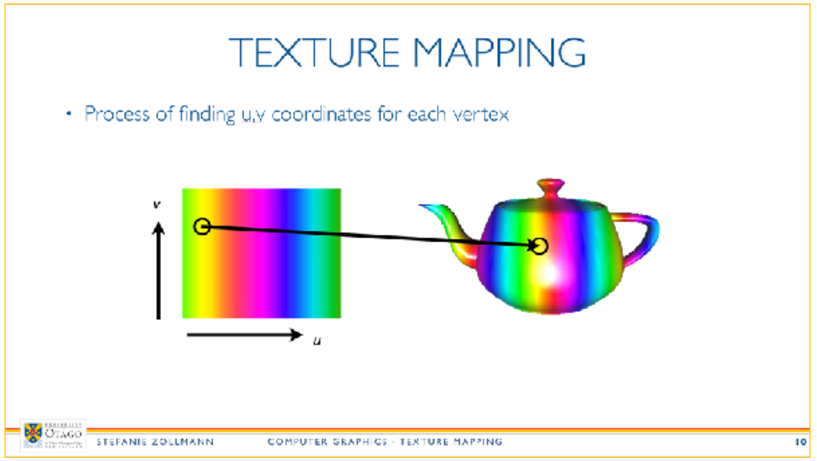
ENVIRONMENT MAPPING

- Reflection mapping
- Defined over surface of enclosing object (e.g. Sphere, Cube, Cylinder)
- Differences to previous techniques:
  - Instead of using P(x,y,z), use the direction of the reflection ray to compute point enclosing object
- Texture contains information about:
  - Light sources
  - Sky
  - Background objects

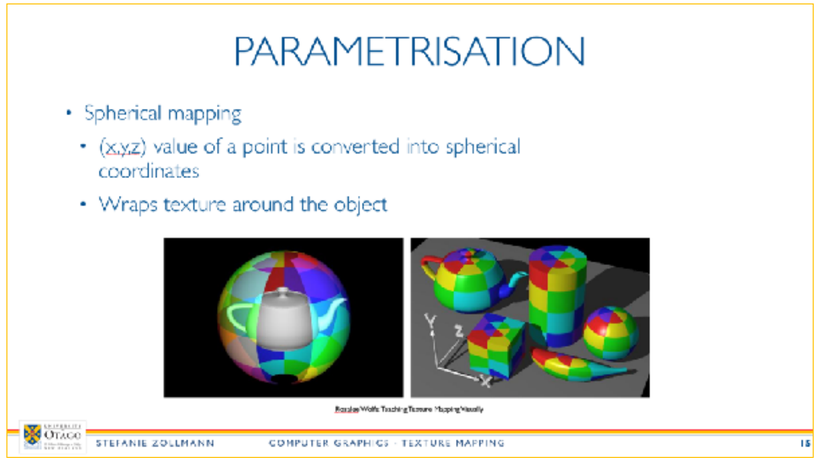


https://commons.wikimedia.org/wiki/File%3ACube_mapped_reflection_example.jpg

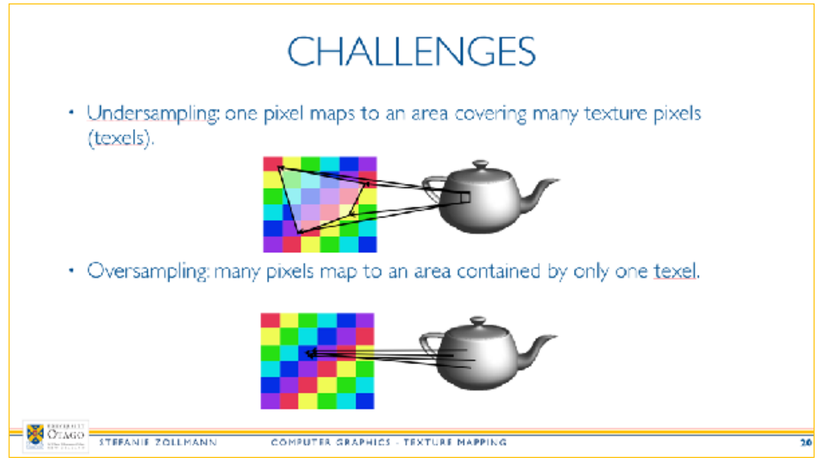
SUMMARY



Texture Mapping

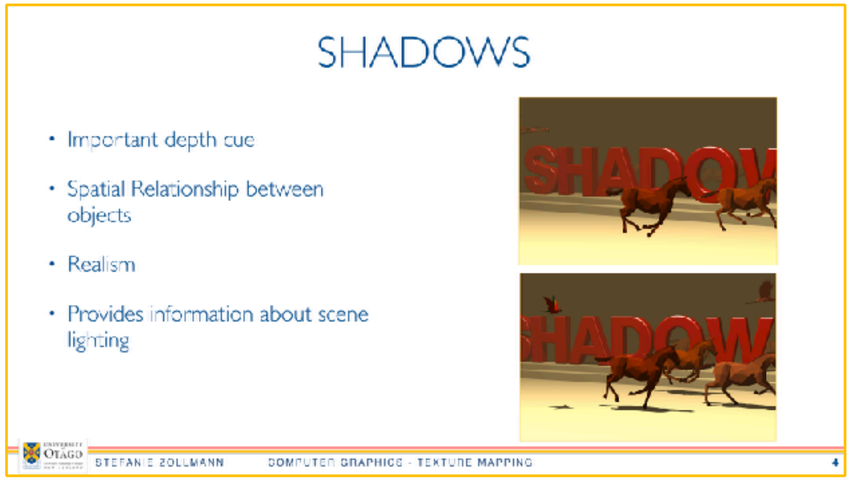


Parametrisation

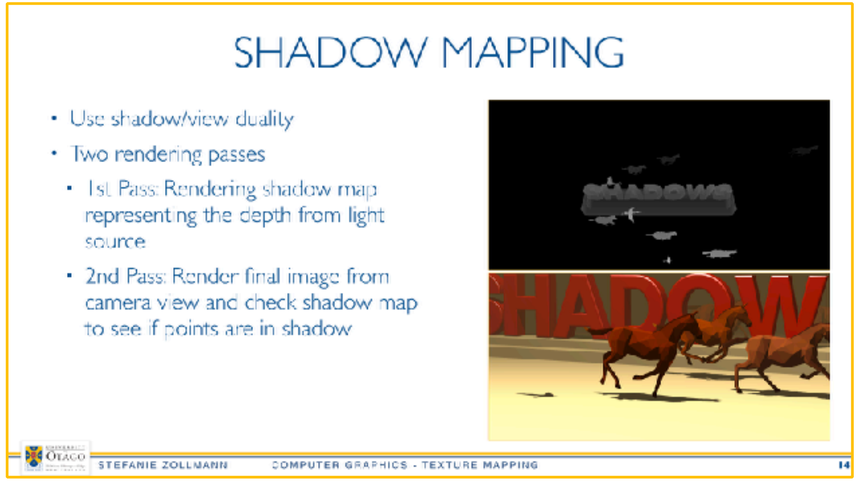


Challenges

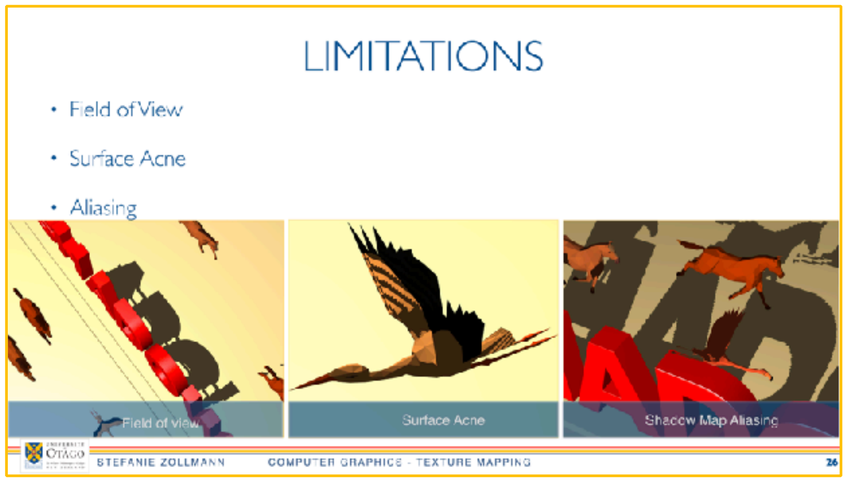
WHAT'S NEXT



Shadows



Shadow Mapping



Limitations & Improvements

Thank You!

For more material visit

http://www.cs.otago.ac.nz/cosc342/