Texture Mapping

Texture mapping

- Objects have properties that vary across the surface

Texture Mapping

- So we make the shading parameters vary across the surface

•[Foley et al. / Perlin]

Examples

- Wood gym floor with smooth finish
  - diffuse color k_D varies with position

- Glazed pot with finger prints
  - specular exponent n varies with position

- Simulating stone, fabric, …
  - to approximate effects of small-scale geometry
  - they look flat but a lot better than nothing

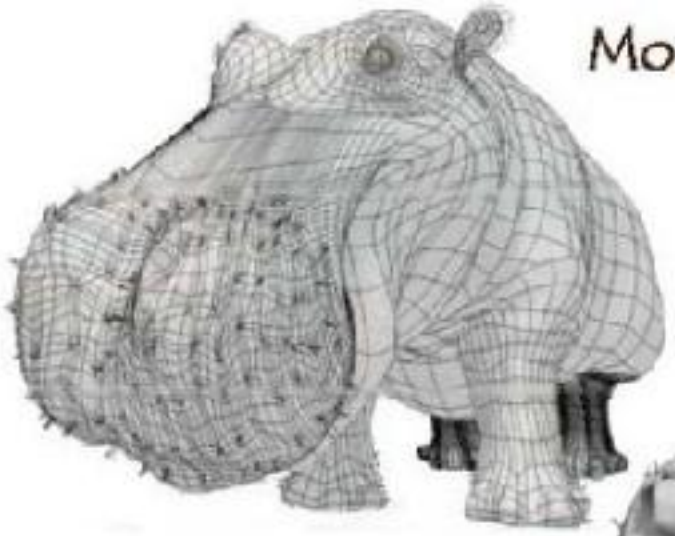
Texture mapping

- Adds visual complexity; makes appealing images

•[Pixar / *Toy Story*]

Texture mapping

- Color is not the same everywhere on a surface
  - one solution: many triangles
- Want a function that assigns a color to each point
  - the surface is a 2D domain, so that is essentially an image
  - can represent using any image representation
  - raster texture images are very popular



Model

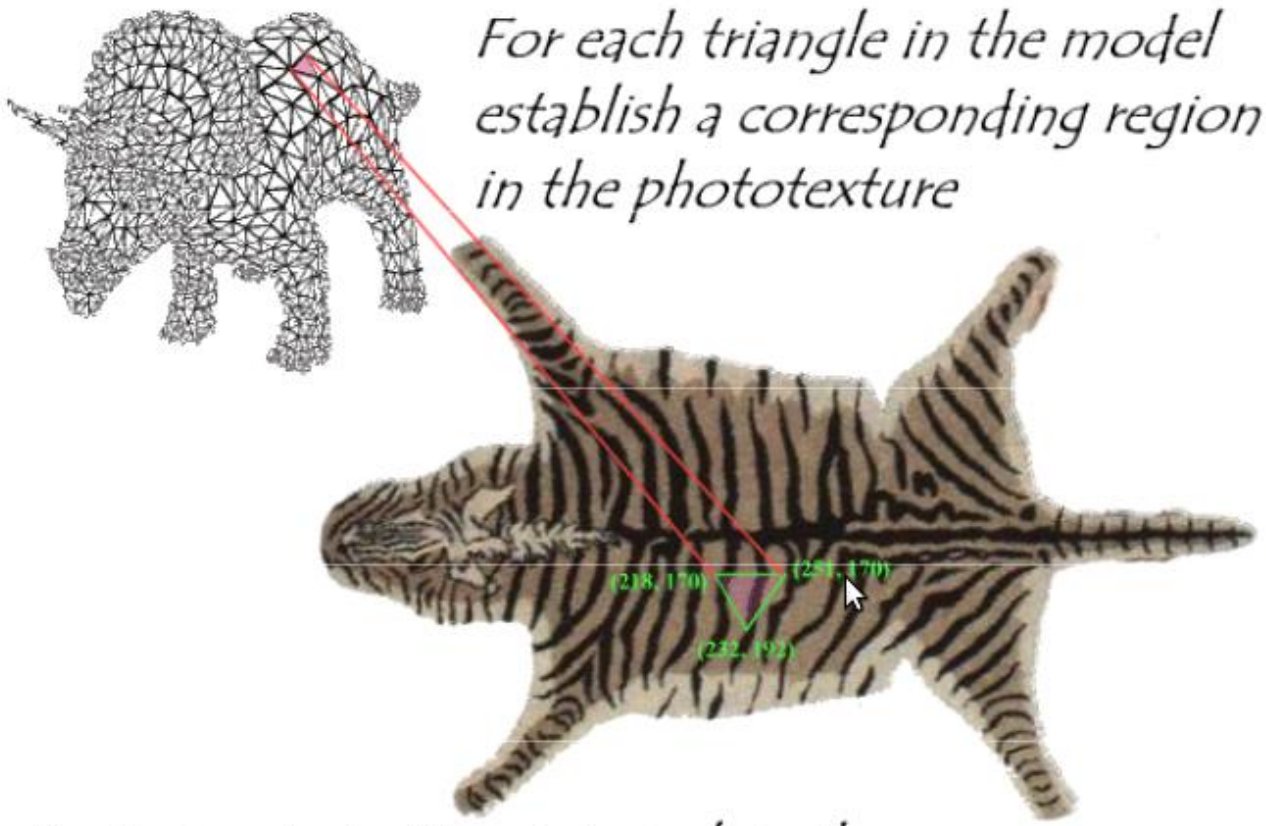
Model + Shading

Model + Shading
+ Textures

At what point
do things start
looking real?

For more info on the computer artwork of Jeremy Birn
see http://www.3drender.com/jbirn/productions.html

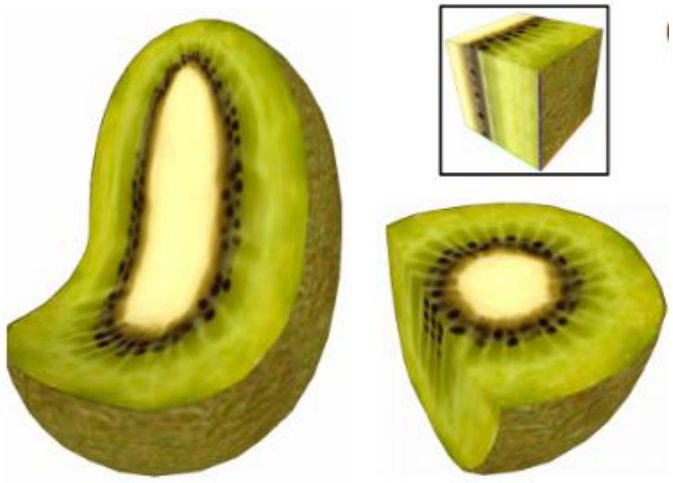
Photo textures



Excerpted from MIT EECS 6.837,

Mapping textures to surfaces

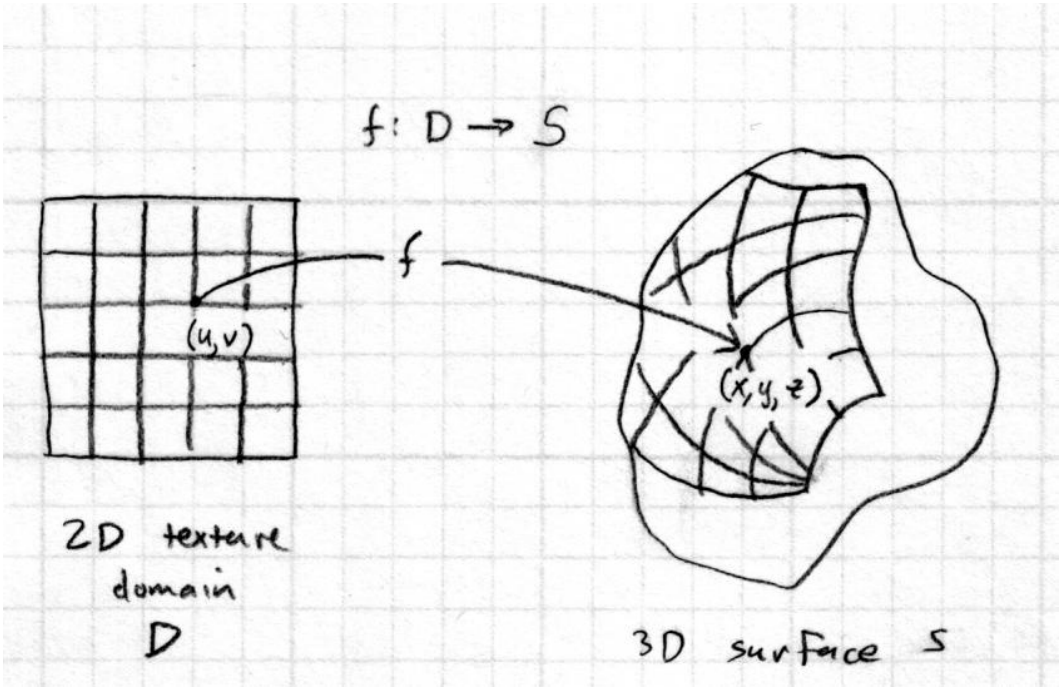
- Usually the texture is an image (function of u, v)
  - obvious only for a flat rectangle the same shape as the image
  - otherwise more interesting
- *3D textures* also exist
  - a function of (u, v, w)
  - can just evaluate texture
  - at 3D surface point
  - good for solid materials



[Lapped Solid Textures, SIGGRAPH 2008]

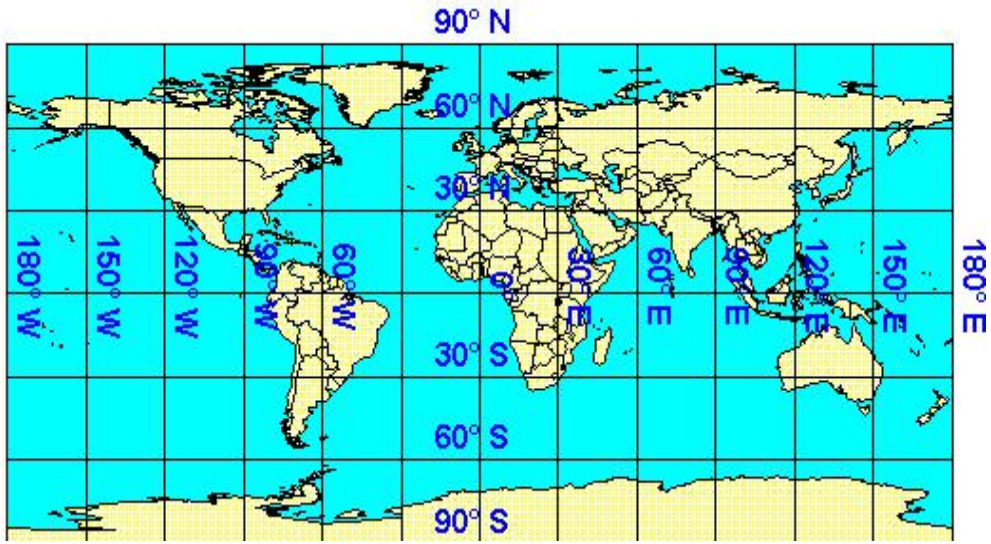
Mapping textures to surfaces

- "Putting the image on the surface"
  - this means we need a function *f* that tells where each point on the image goes
  - this looks a lot like a parametric surface function
  - for parametric surfaces you get *f* for free



Examples of coordinate functions

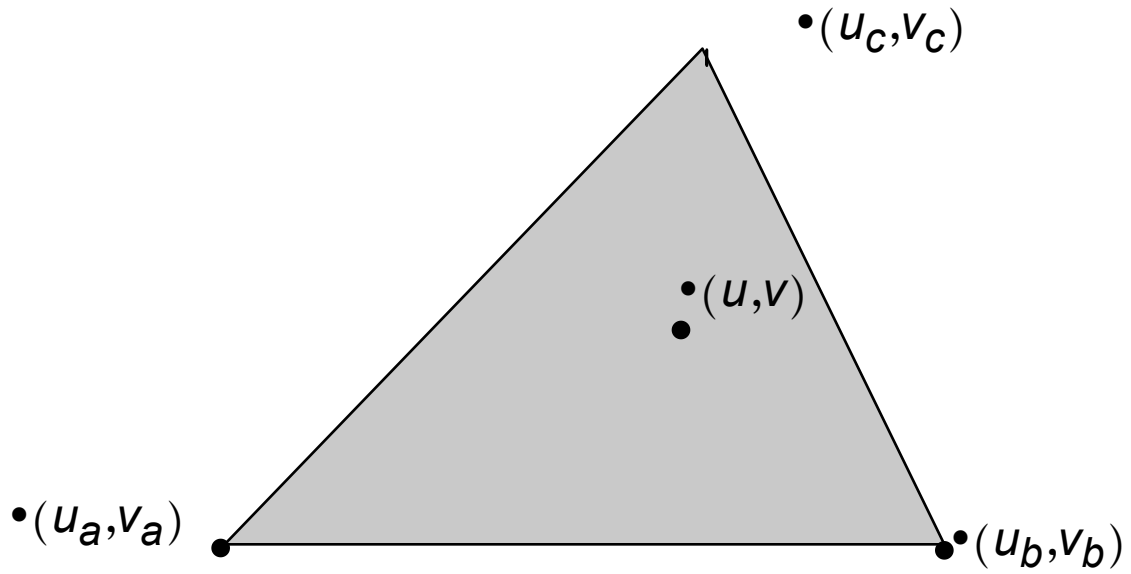
- For a sphere: latitude–longitude coordinates
  - φ maps point to its latitude and longitude



•[map: Peter H. Dana]

Examples of coordinate functions

- Triangles
  - specify (u,v) for each vertex
  - define (u,v) for interior by linear interpolation



Texture coordinates on meshes

- Texture coordinates become per-vertex data like vertex positions

- How to set vertex (u,v)s?
  - manually using box, spherical, cylindrical maps
  - or use some kind of optimization
    - try to choose vertex (u,v)s to result in a smooth, low distortion map

Uses of Texture Maps

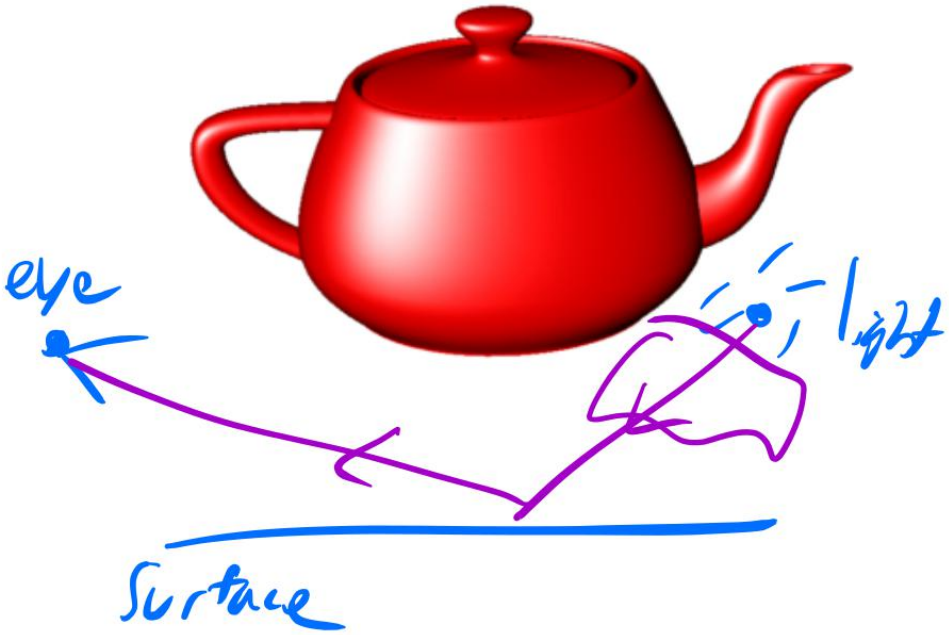
- Texture maps are used to add complexity to a scene
  - Easier to paint or capture an image than geometry
- Model light
- Model geometry, etc

KAIST

Modeling lighting

Local illumination – Fast

- assumes single-bounce path: light – surface – eye



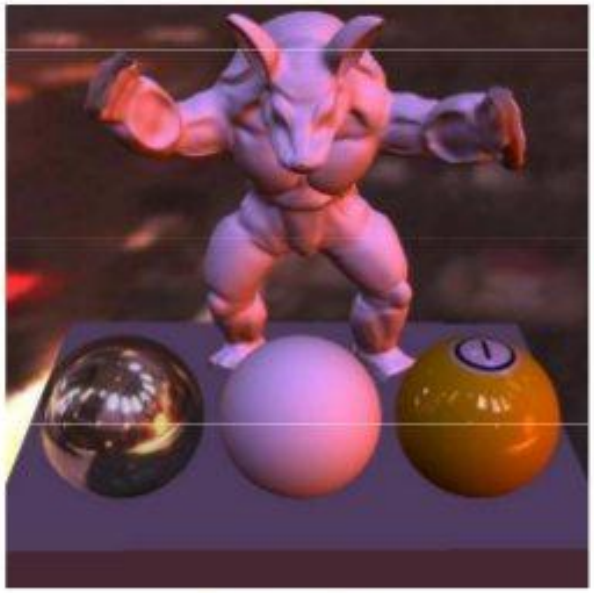
Global illumination – Slow

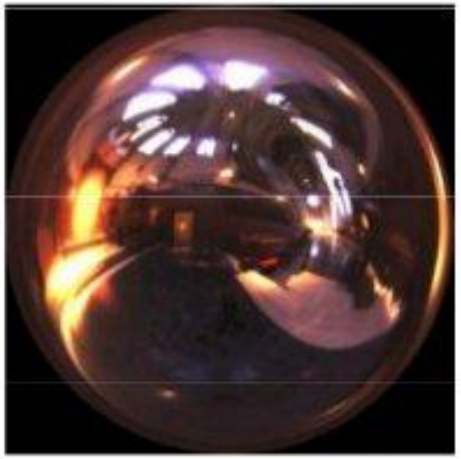
- (More) Physically based computation of light transport throughout the scene

- Efficiently approximate global illumination using precomputed texture maps!

Modeling lighting

- **Light maps**
  - Supply the lighting directly
  - Good for static environments
- **Projective textures**
  - Can be used to simulate a spot light
  - Shadow maps
- **Environment maps**
  - A representation of the scene around an object
  - Good for reflection





Light maps in Quake

- **Light maps are used to store pre-computed illumination**

	Texture Maps	Light Maps
Data	RGB	Intensity
Resolution	High	Low

Light map image by Nick Chirkov

Textures Only

Textures & Light Maps

Light maps

- precompute and “bake” into texture:
  static lighting of diffuse surfaces

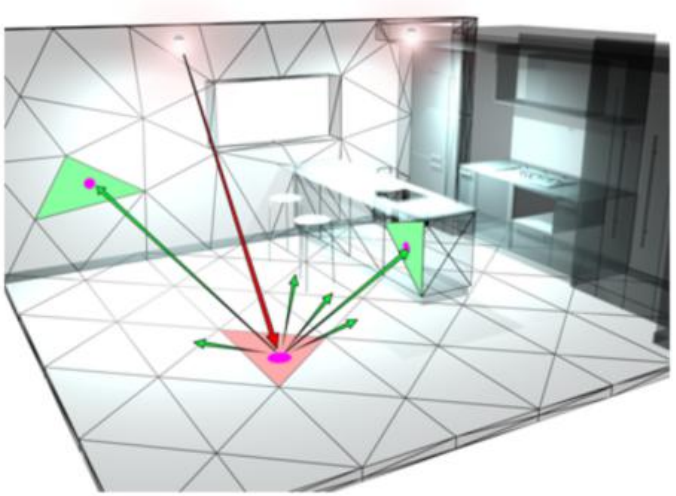
[https://blogs.unity3d.com/2017/03/31/]

“RADIOSITY” METHOD FOR GLOBAL ILLUMINATION

Form factor F_ij: Fraction of light leaving surface i and arriving at surface j. This depends on the shape, distance, orientation, and relative occlusions of the two surface patches.

Solve a set of simultaneous linear equations for the unknown energies.

Assumes all surfaces are Lambertian, i.e., diffuse.
Requires subdivision of scene into patches.
The n^2 form factors are expensive to compute.

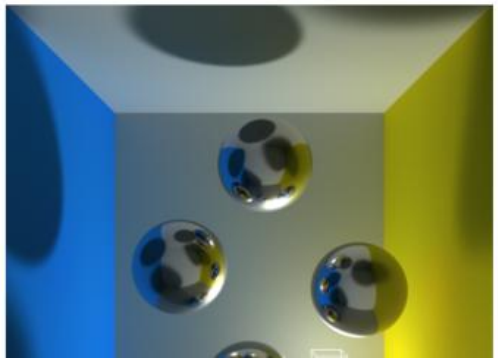


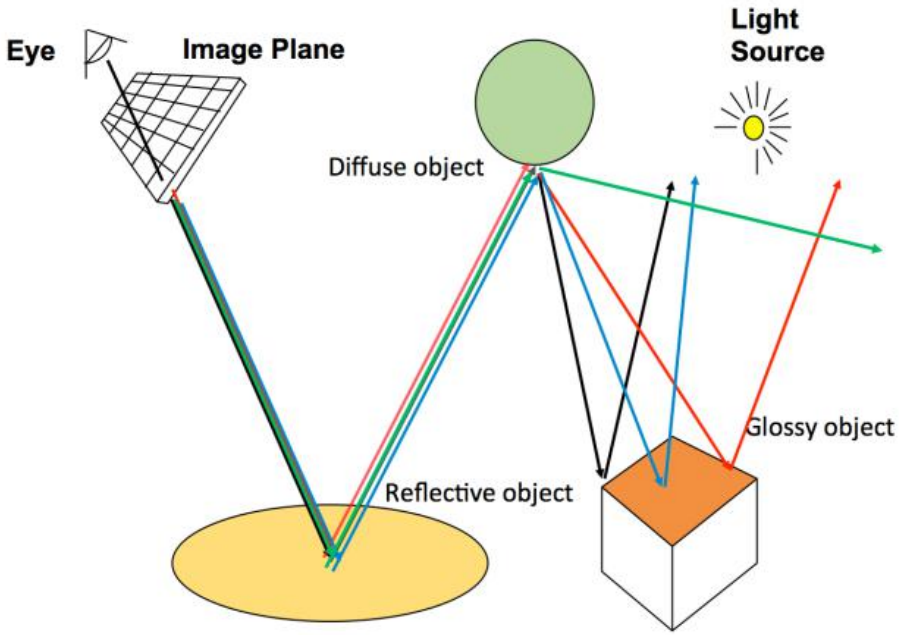
Radiosity: A ray of light that hits a surface is reflected by multiple diffuse rays, which can themselves illuminate other surfaces. Surfaces are subdivided to increase accuracy of the solution.

Michiel van de Panne,
UBC

PATHTRACING

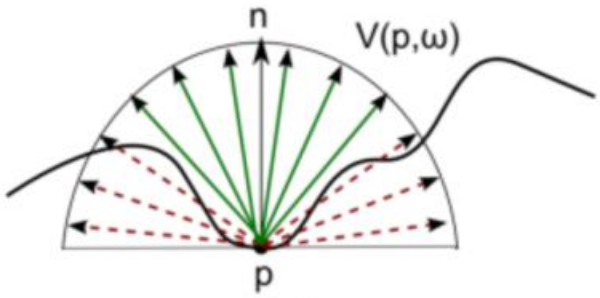
- global illumination
- trace paths from eye into scene
- Monte-Carlo sampling of directions
- multiple diffuse bounces
- average many sample rays per pixel
- "noisy" images with few samples
  - ML with deep nets to remove noise



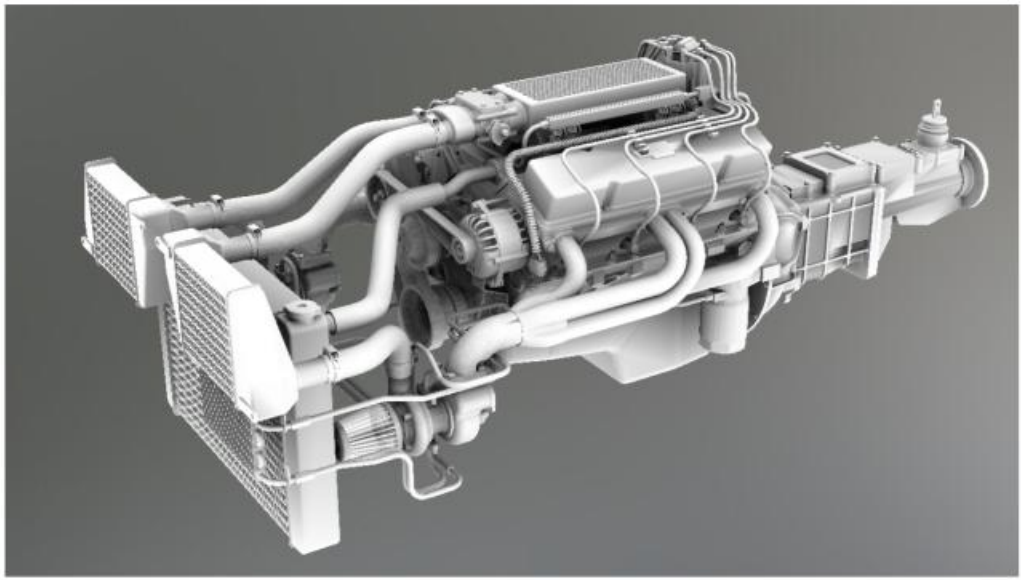


Michiel van de Panne, UBC

AMBIENT OCCLUSION



$$AO(\mathbf{p},\mathbf{n}) = \frac{1}{\pi}\int_{\Omega} V(\mathbf{p},\omega)\,\mathbf{n}\cdot\omega\,d\omega.$$



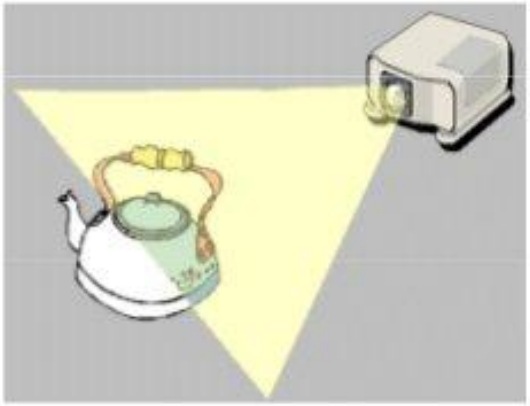
http://www.redway3d.com/

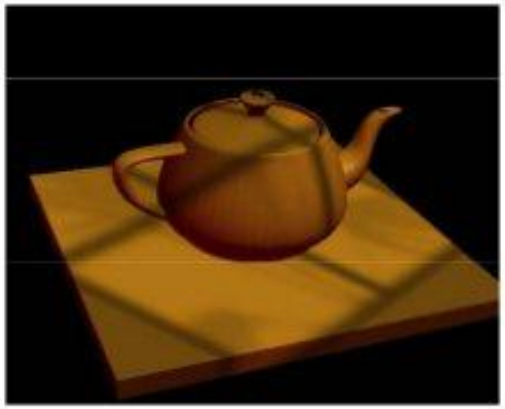
Assume that light is coming from all directions.
For a given point, compute the area of the surrounding hemisphere that is open.
Precompute and store this.

Michiel van de Panne,
UBC

Projective textures

- Treat the texture as a slide in a projector
  - A good model for shading variations due to illumination (cool spotlights)
- Projectors work like cameras in reverse
  - Camera: color of point in scene → color of corresponding pixel
  - Projector: color of pixel → color of corresponding point in the scene



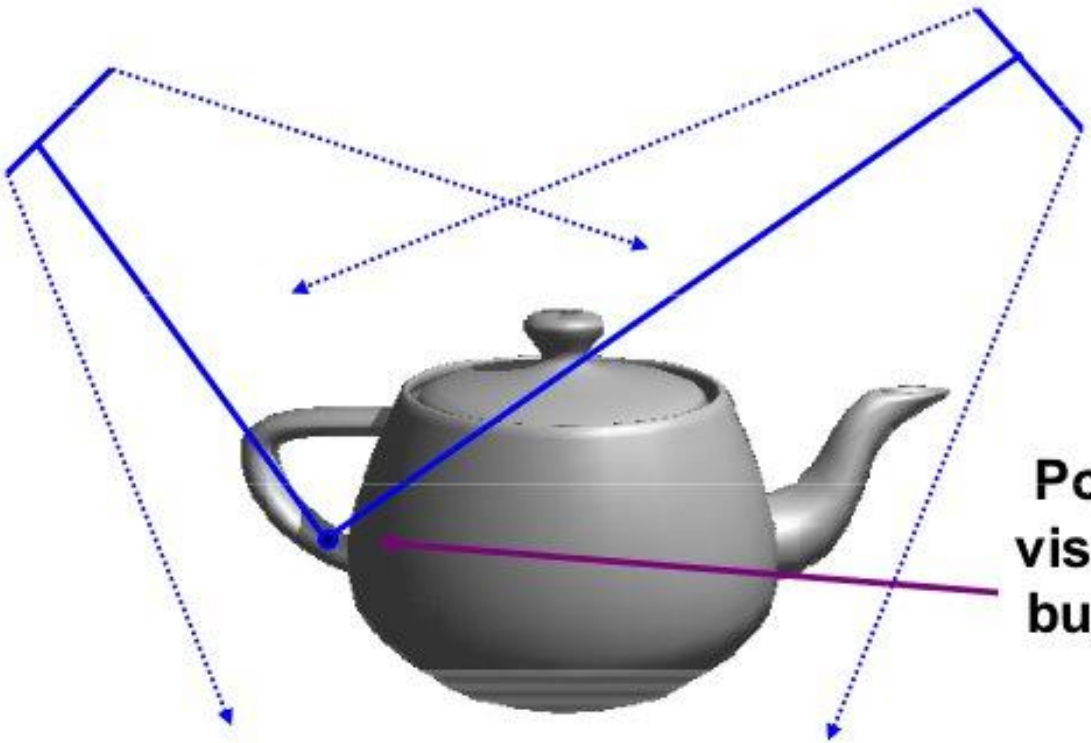


Shadow maps

Eye

Use the depth map in the light view to determine if sample point is visible

Light

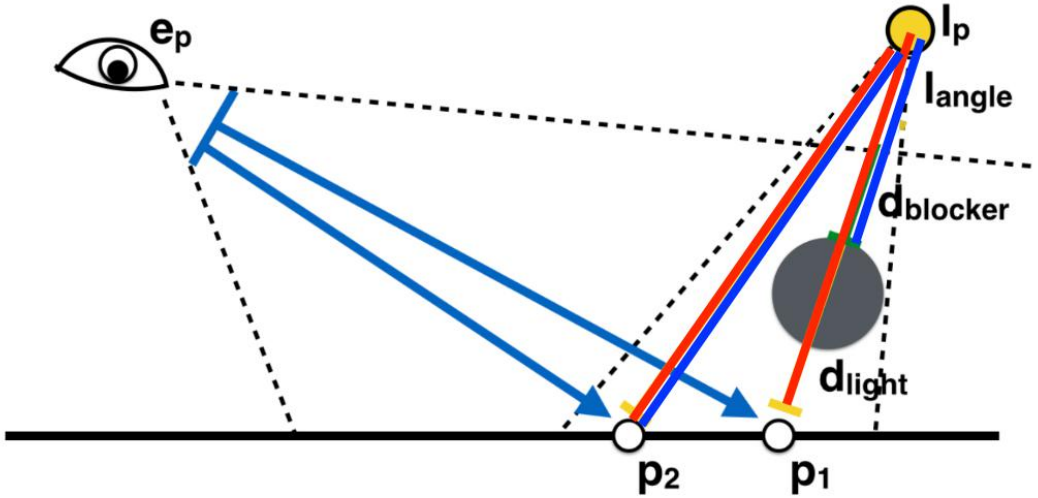


Point in shadow visible to the eye, but not visible to the light

KAIST

Shadow maps

Usually used with projective rendering



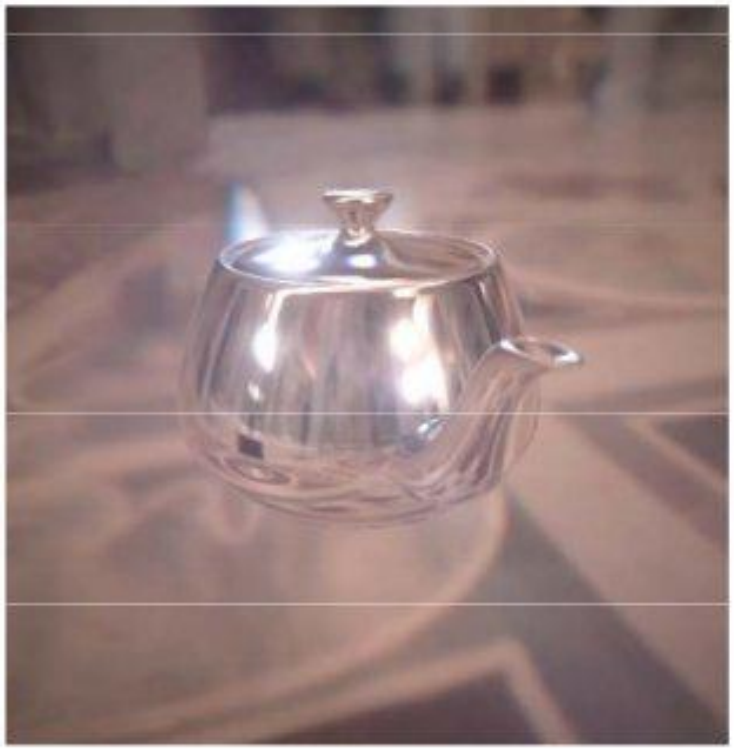
http://adrien.io/opengl-course/lesson-3-shadow-mapping/

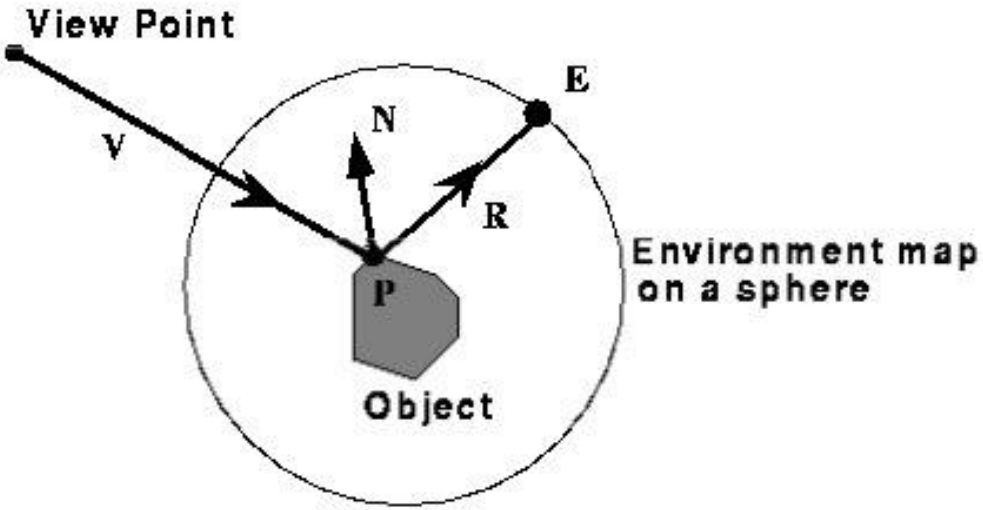
(1) render scene from light source; store the z-buffer
(2) render scene from camera view; p is in shadow if d_light > d_blocker

Issues: resolution of the shadow map image? No soft shadows.

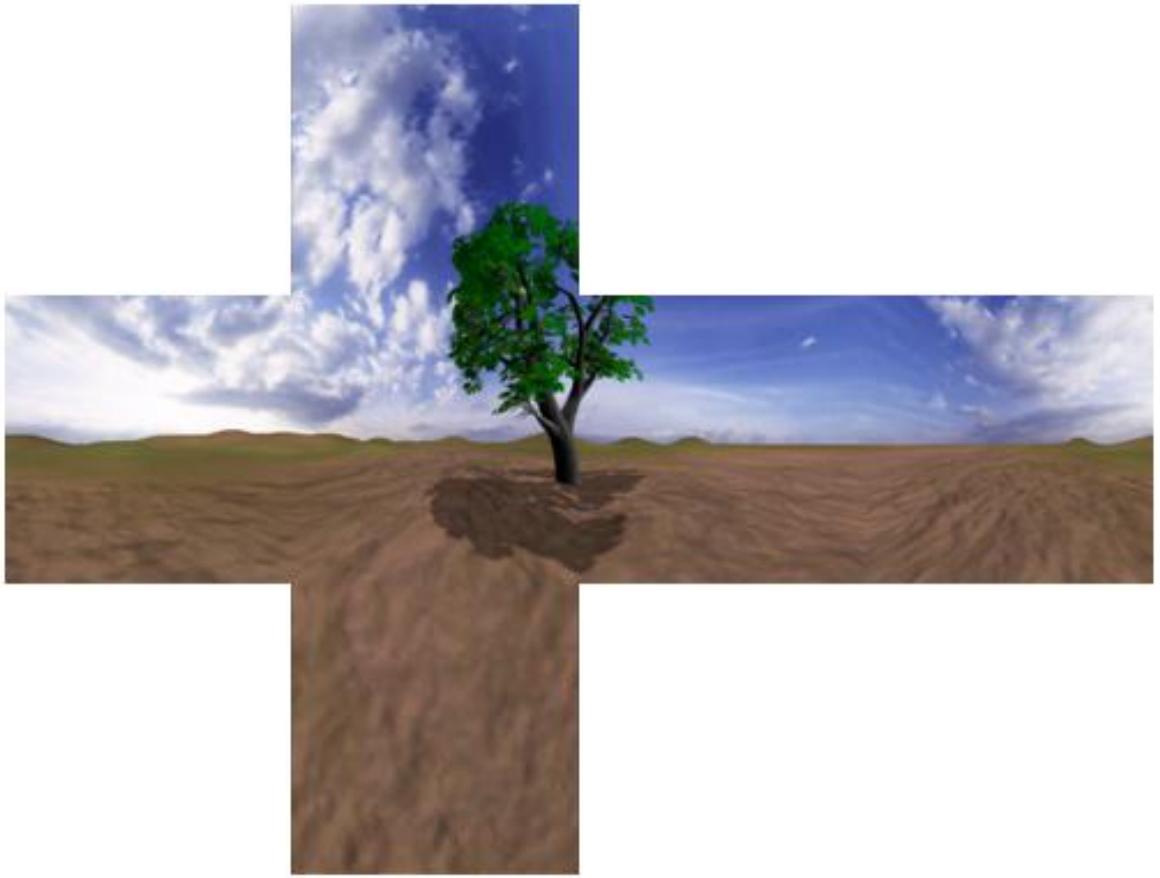
Environment maps

- **Simulate complex mirror-like objects**
  - **Use textures to capture environment of objects**
  - **Use surface normal to compute texture coordinates**





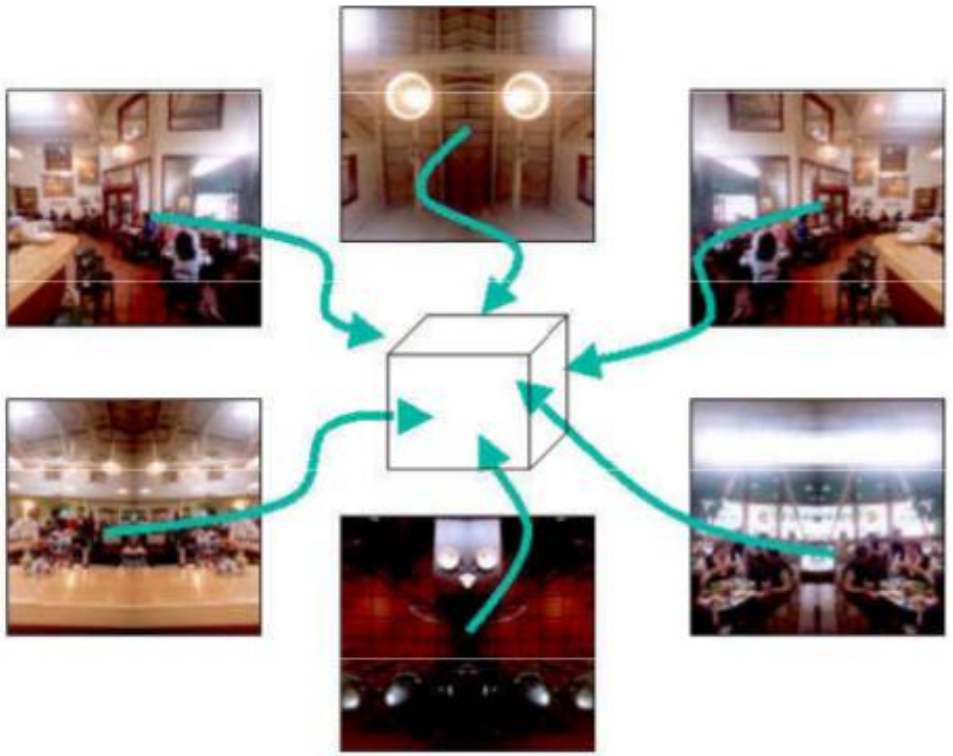
Cube environment map



•[Ned Greene]

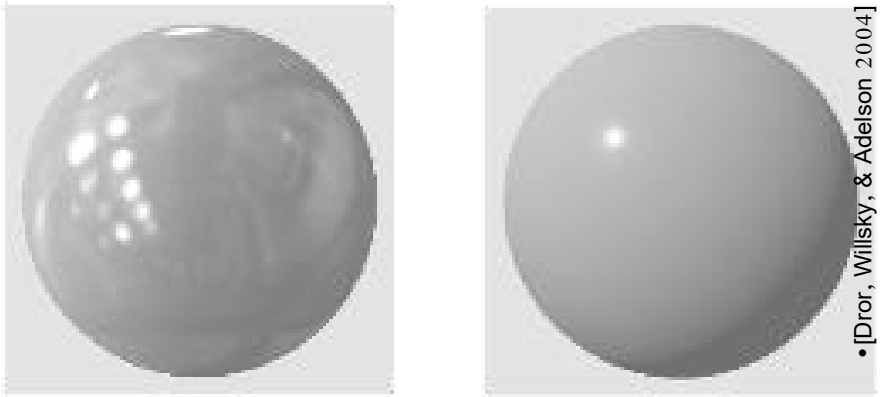
Cube environment map

- Maps a viewing direction b and returns an RGB color
  - Use stored texture maps



Environment mapping

- Appearance of shiny objects
  - Phong highlights produce blurry highlights for glossy surfaces.
  - A polished (shiny) object reflects a sharp image of its environment.
- The whole key to a shiny-looking material is providing something for it to reflect.



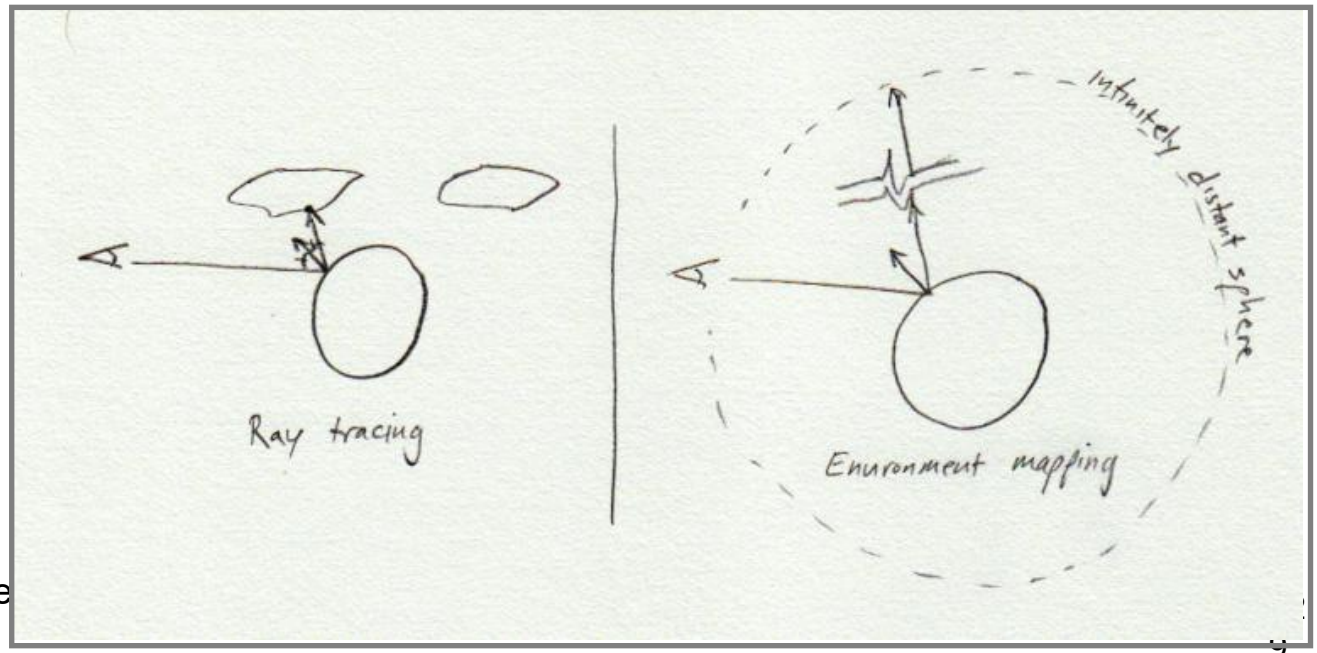
(a) (b)

•[Dror, Willsky, & Adelson 2004]

Figure 2. (a). A shiny sphere rendered under photographically acquired real-world illumination. (b). The same sphere rendered under illumination by a point light source.

Reflection mapping

- Ray tracing
  - trace a recursive ray into the scene—too expensive
- If scene is infinitely far away, depends only on direction
  - a two-dimensional
  - function

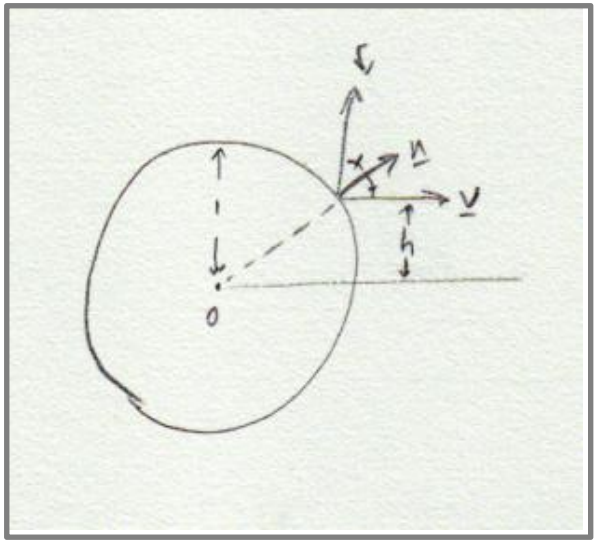


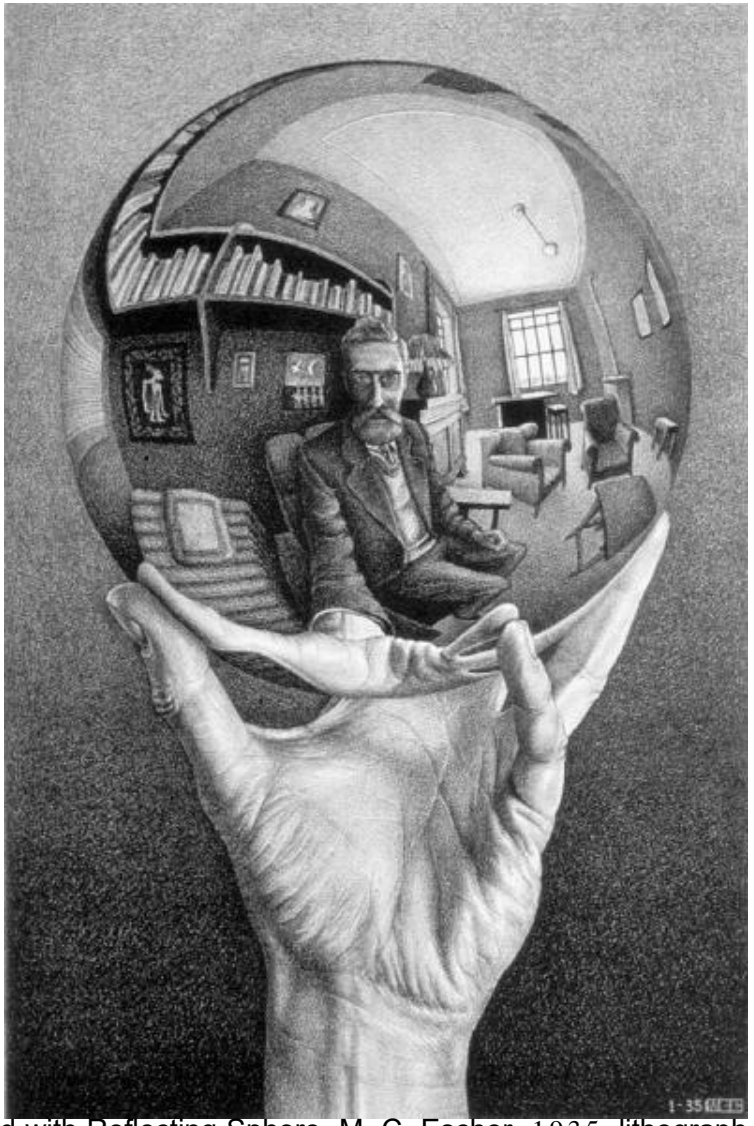
Environment map

- A function from the sphere to colors, stored as a texture.

•[Blinn & Newell 1976]

Spherical environment map





•Hand with Reflecting Sphere. M. C. Escher, 1935. lithograph

Environment Maps

- [Paul Debevec]

•[CS467 slides]

Environment maps – problems

- Expensive to update dynamically
- Not completely accurate

images from NVIDIA

Reflection of swimming pool is wrong

KAIST

Modeling geometries

- Store complex surface details in a texture rather than modeling them explicitly
- Bump maps
  - Modify the existing normal
- Normal maps
  - Replace the existing normal
- Displacement maps
  - Modify the geometry
- Opacity maps and billboards
  - Knock-out portions of a polygon using the alpha channel

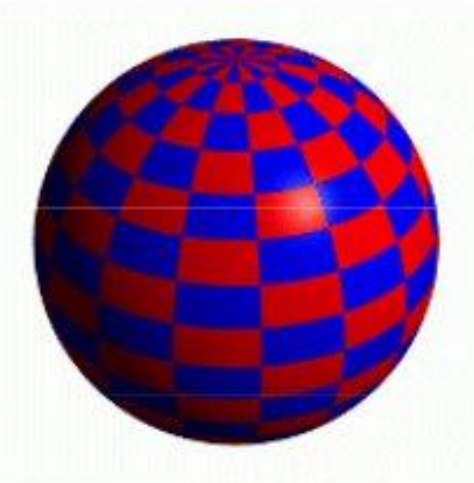
Bump mapping

2D texture maps that are used to model the appearance of surface bumps, by adding small perturbations to the surface normals. The rendered geometry does not actually have bumps, i.e., it is smooth !!

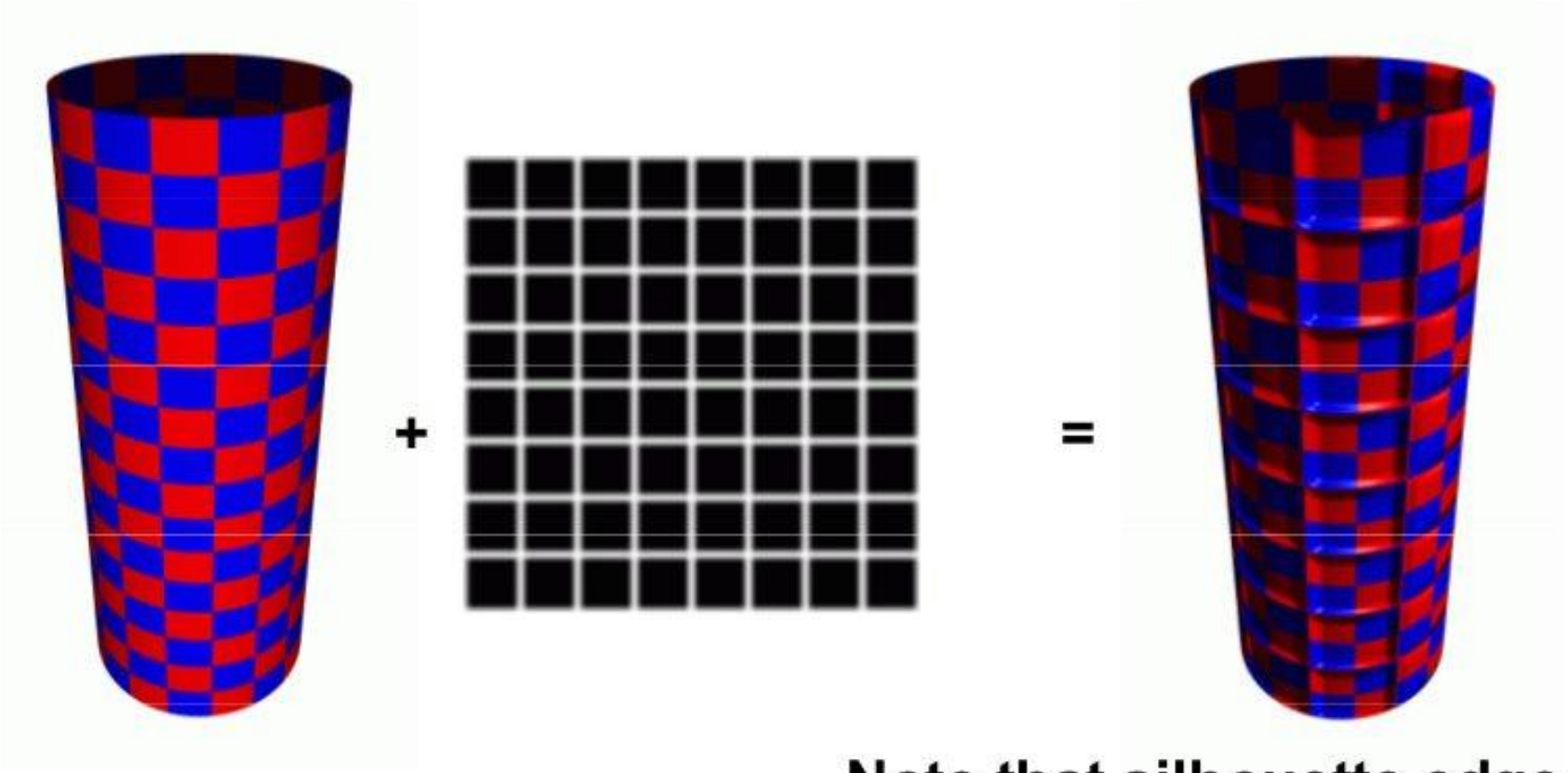
threejs.org: materials/bumpmap

Bump mapping

- **Modifies the normal not the actual geometry**
  - **Texture treated as a *h*eightfield**
  - **Partial derivatives used to change the normal**
  - **Causes surface to appear deformed by the heightfield**



Bump mapping

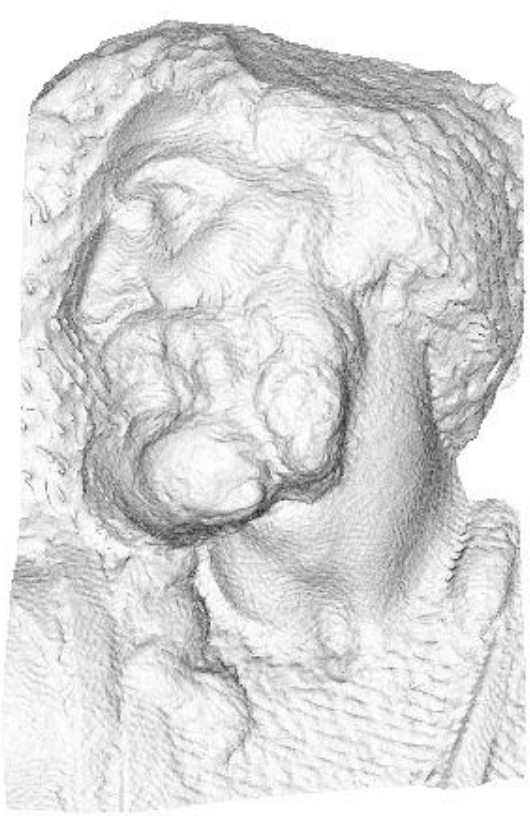


Note that silhouette edge of the object not affected!

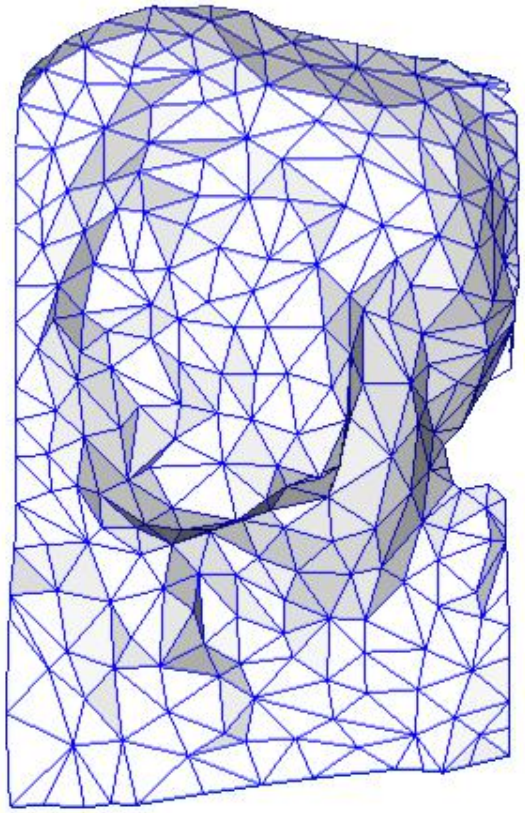
KAIST

Normal mapping

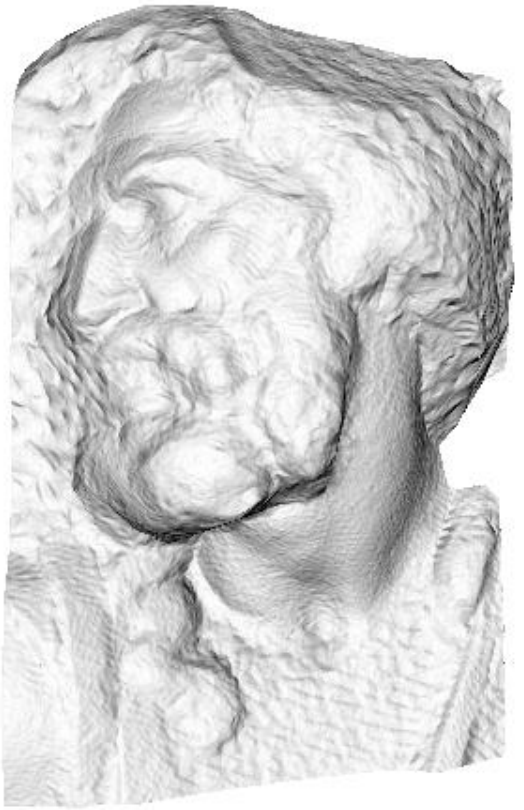
- Replace normals ignoring vertex normals



original mesh
4M triangles



simplified mesh
500 triangles

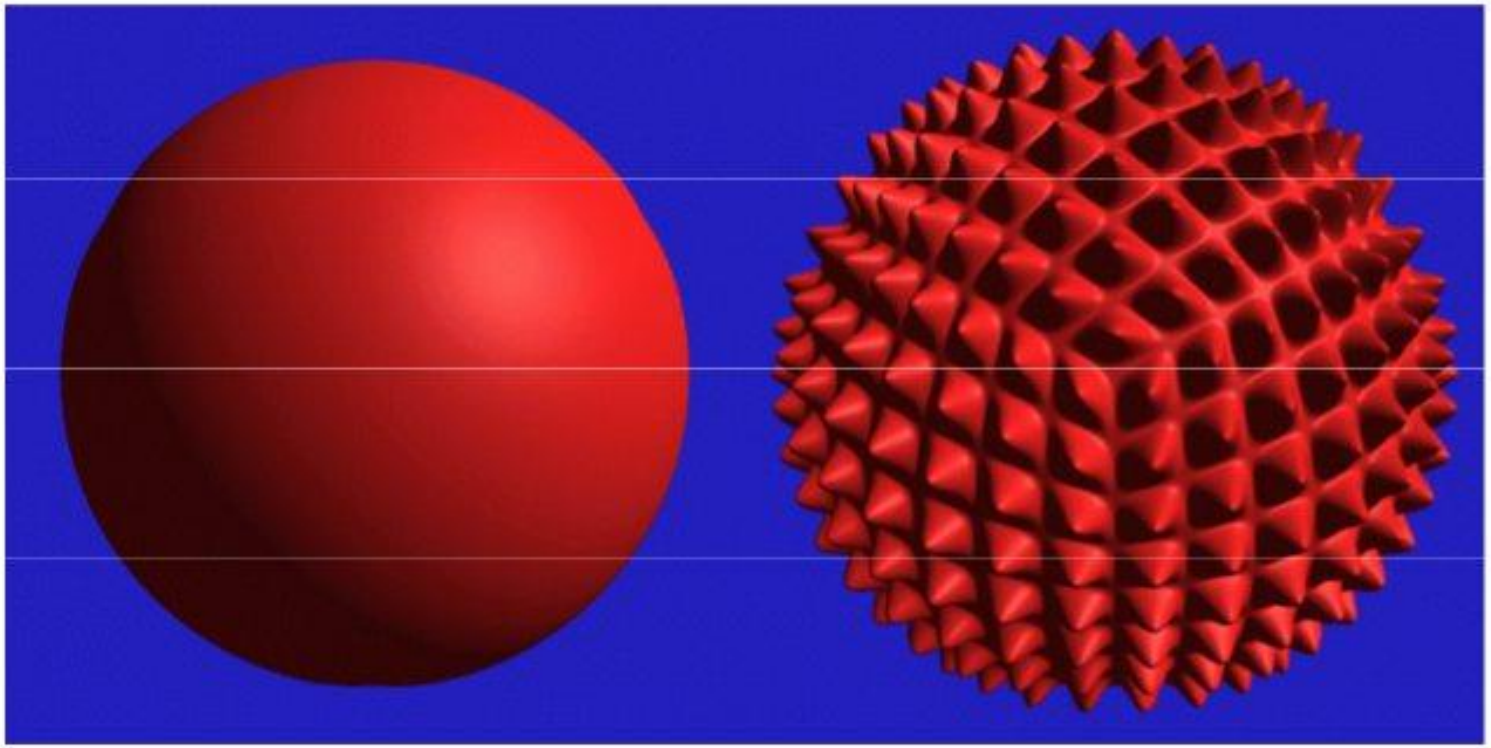


simplified mesh
and normal mapping
500 triangles

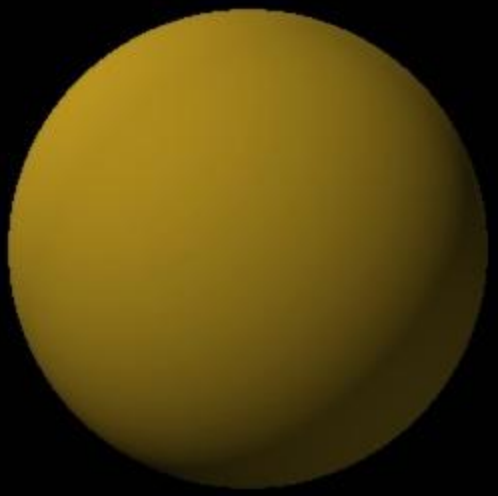
•[Paolo Cignoni]

Displacement mapping

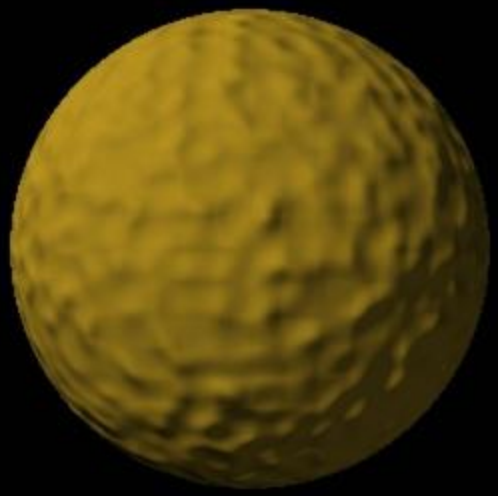
- **Texture maps can be used to actually move surface points**



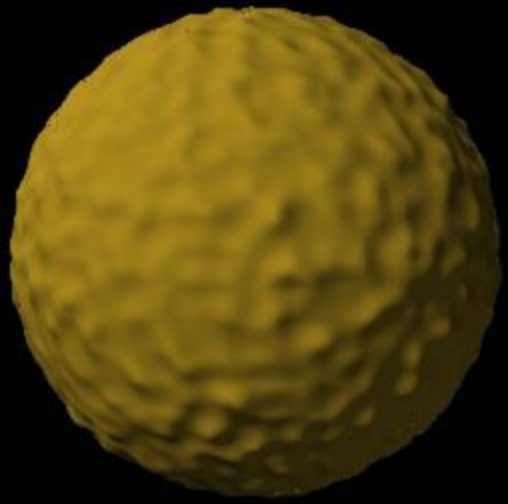
Displacement mapping



- Geometry

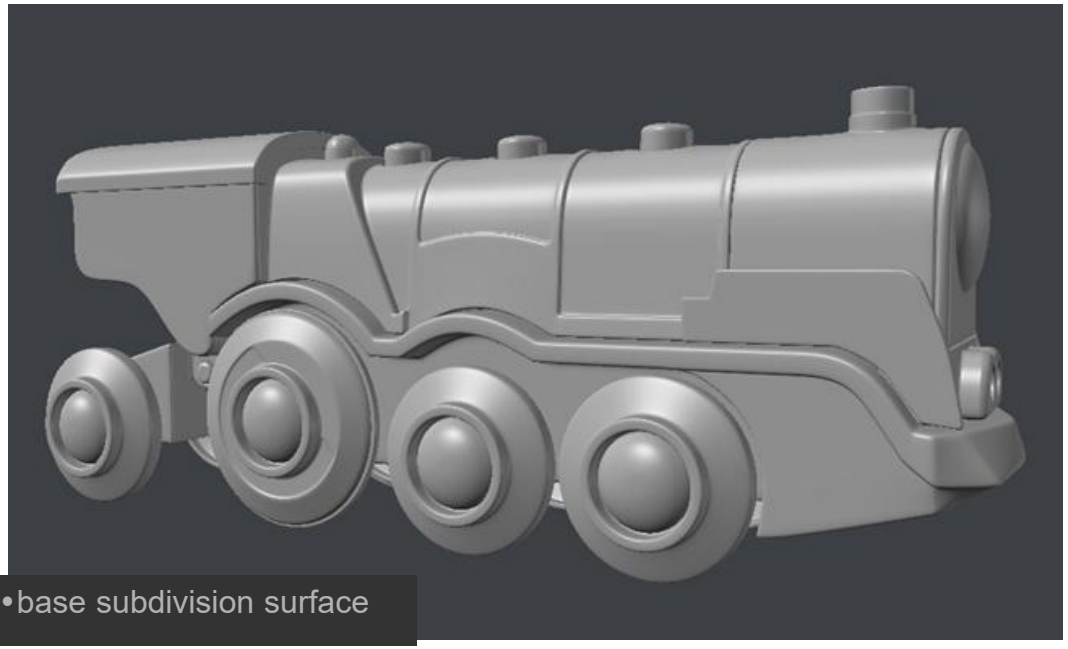


- Bump
- mapping



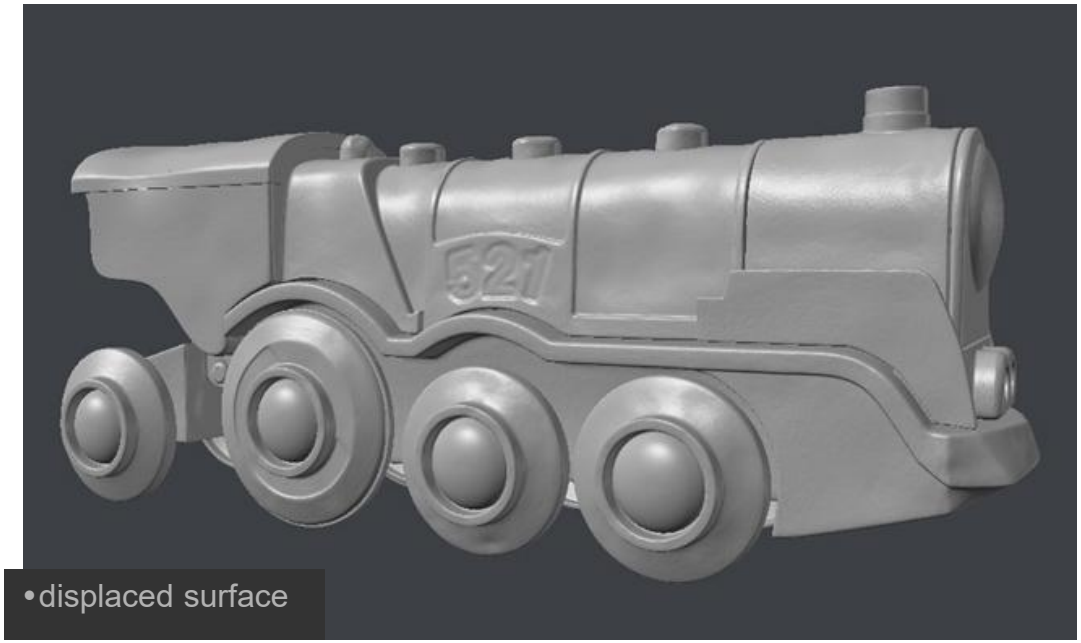
- Displacement
- mapping

- [CS467 slides]



•base subdivision surface

•hand-painted displacement map (detail)

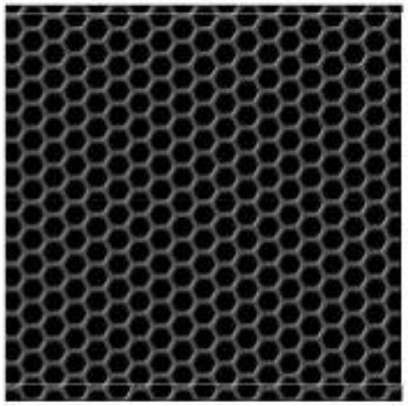


•displaced surface

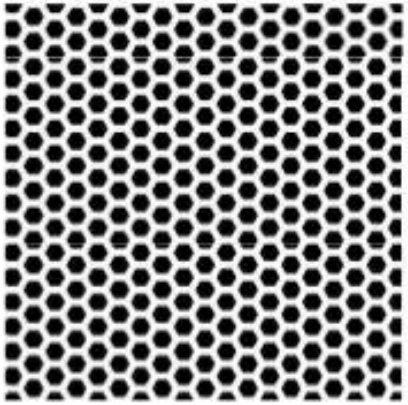
•Paweł Filip
•tolas.wordpress.com


521
fryrender
physically-based render engine
©2007 Paweł Filip

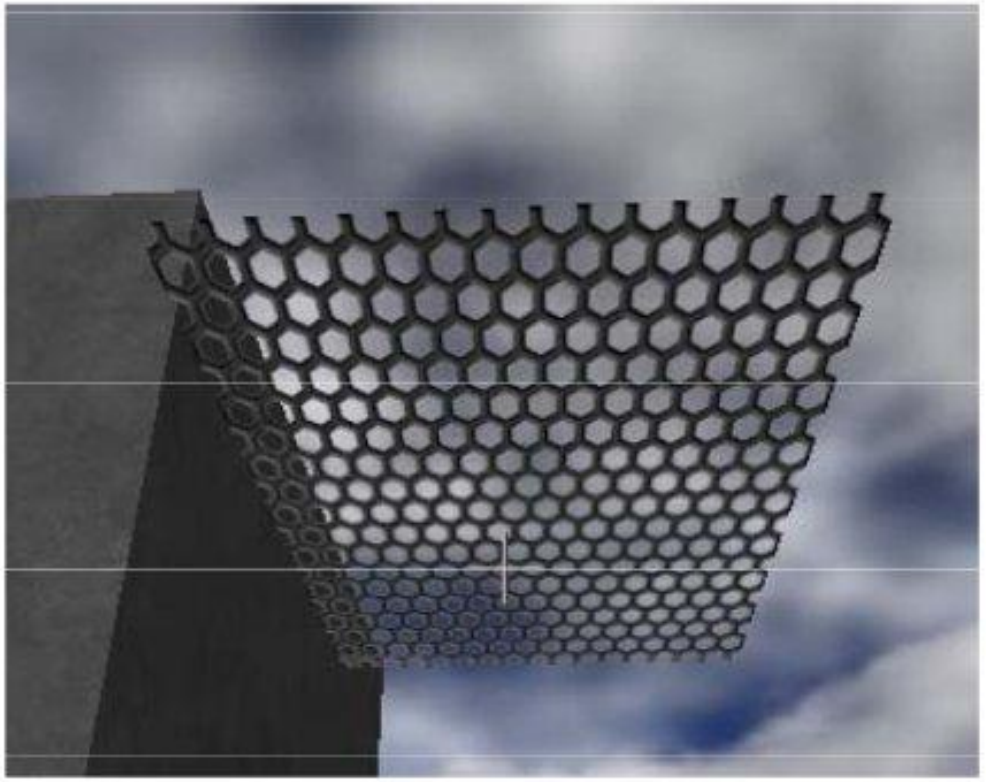
Opacity maps



RGB channels

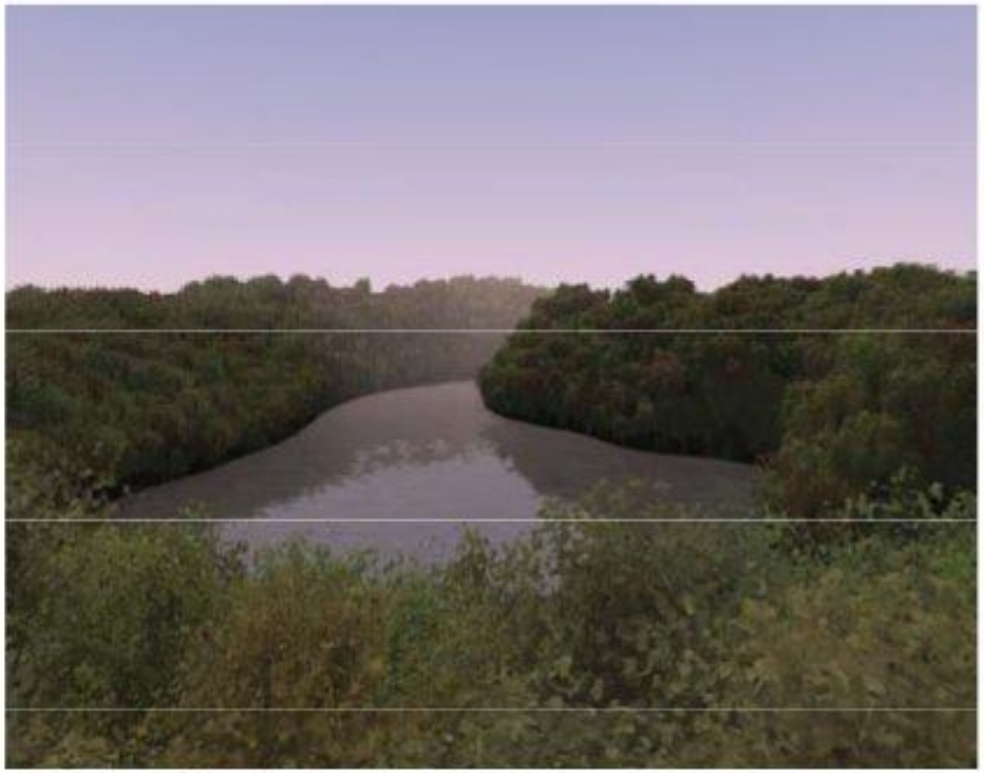


alpha channel



Use the alpha channel to make portions of the texture transparent

Billboards

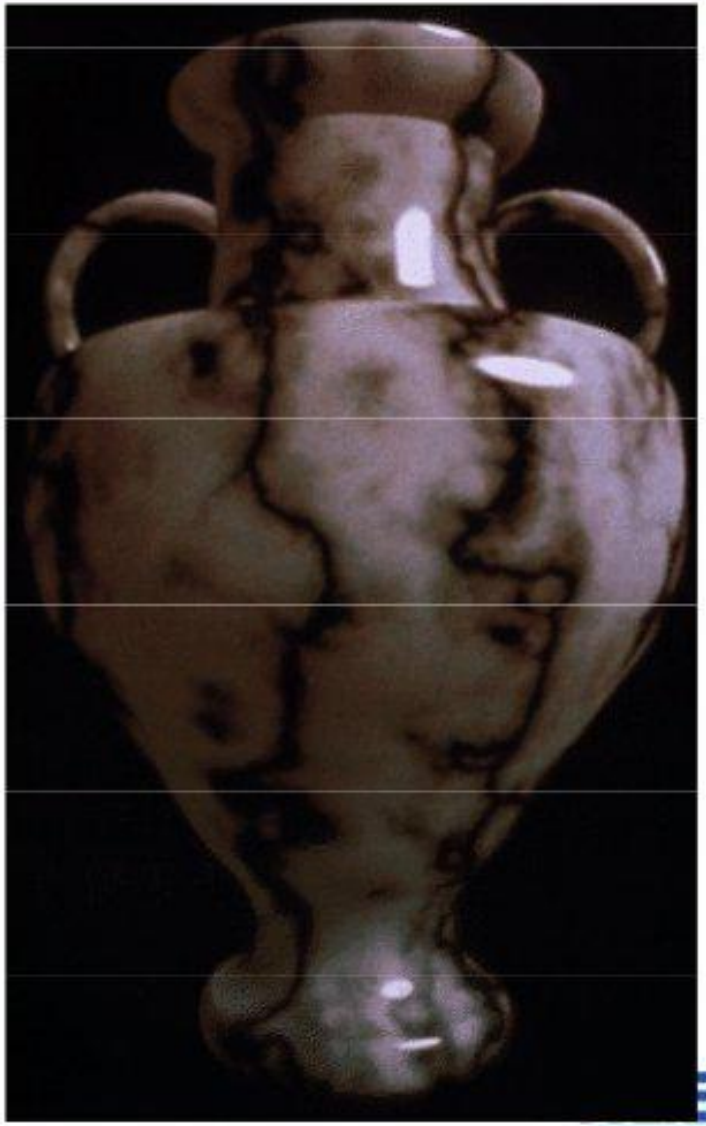


Replace complex geometry with polygons
texture mapped with transparent textures

KAIST

3D or solid textures

- **Solid textures are three dimensional assigning values to points in 3 space**
  - **Very effective at representing some types of materials such as marble and wood**
  - **The object is "carved" out of the solid texture**
- **Generally, solid textures are defined procedural functions rather than tabularized or sampled functions as used in 2D**



Another definition

- **Texture mapping**: a general technique for storing and evaluating functions.

- They're not just for shading parameters any more!