CS4620/5620: Lecture 24

Texture Mapping

Announcements

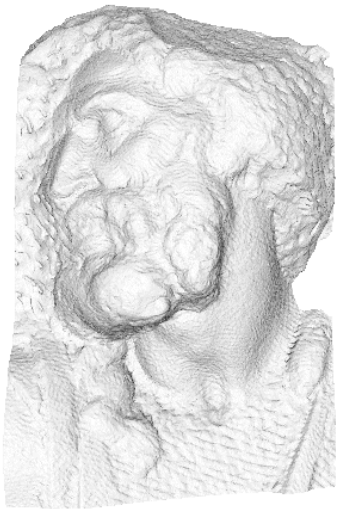
- PA2B out today
  - Material will be covered today and Wed
- Plan for next few lectures
  - Texturing
  - Wed: Perlin noise, Sampling and anti-aliasing
  - Friday onwards: Splines

Other uses of texture mapping

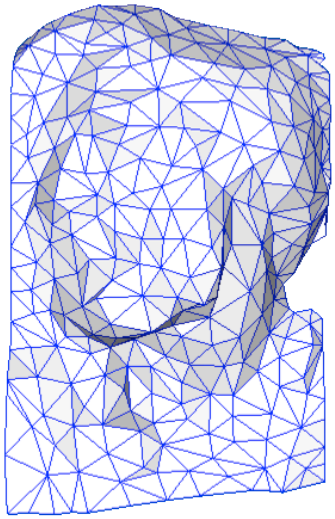
- Reflection maps
- Environment maps
- Normal maps
- Bump maps
- Displacement maps
- Shadow maps
- Irradiance maps
- ...

Normal mapping

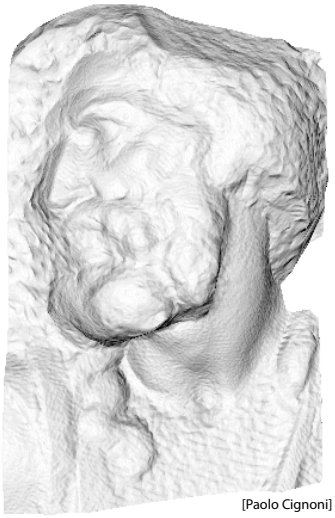
- Stores normals as texture map over coarse geometry



original mesh
4M triangles



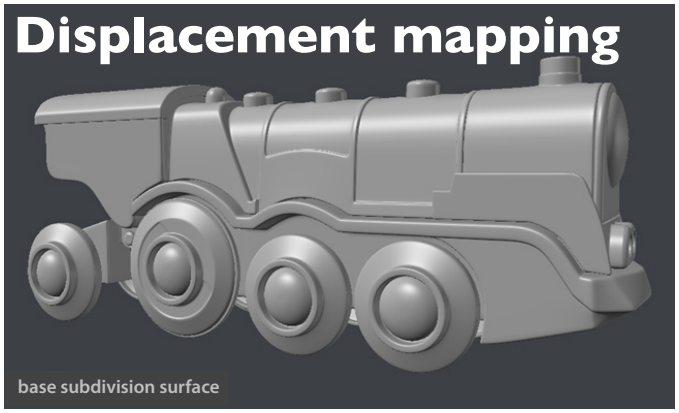
simplified mesh
500 triangles

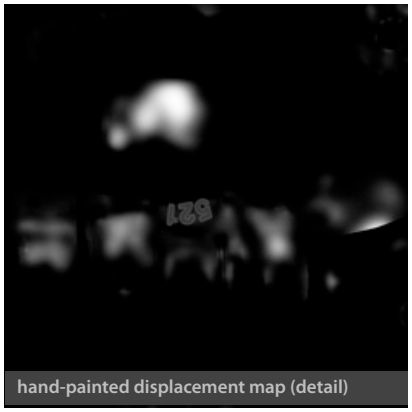


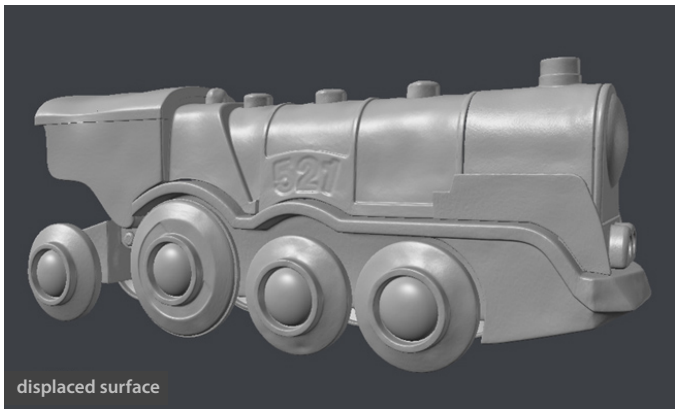
[Paolo Cignoni]

simplified mesh
and normal mapping
500 triangles

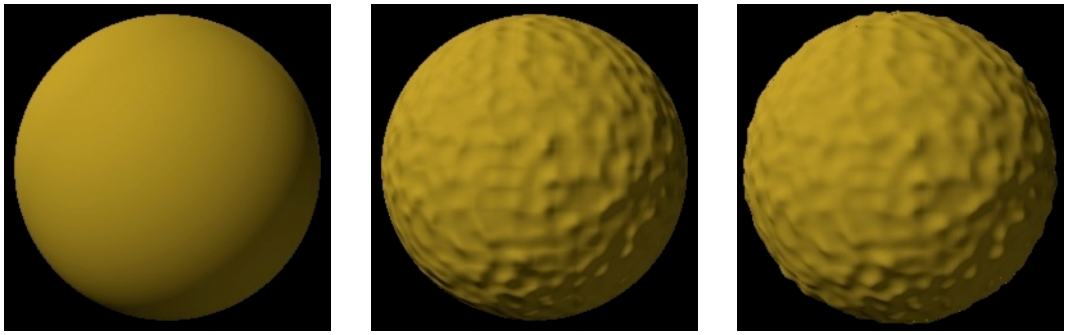
Displacement mapping







Paweł Filip
tolas.wordpress.com



no map normal map displacement map

Altogether: Gloss Maps/Normal Maps

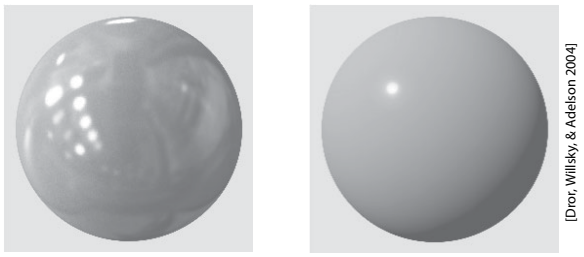
n.l d + (n.h)^m g

ATI Technologies, Inc.

- Normal maps, diffuse map, gloss map, shininess maps

Reflection mapping

- Early (earliest?) non-decal use of textures
- 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]

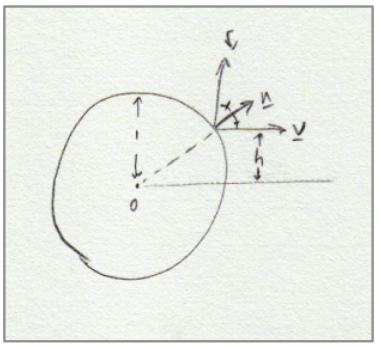
Figure 2. (a). A shiny sphere rendered under photographically acquired real-world illumination. (b). The same sphere rendered under illumination by a point light source.

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

Spherical environment maps

[Paul Debevec]

Types of Mappings: Cube Mapping

Cube mapping

Sphere Mapping Example

Other uses of texture mapping

- Reflection maps
- Environment maps
- Normal maps
- Bump maps
- Displacement maps
- Shadow maps
- Irradiance maps
- ...

Another definition

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

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

Texture mapping from 0 to infinity

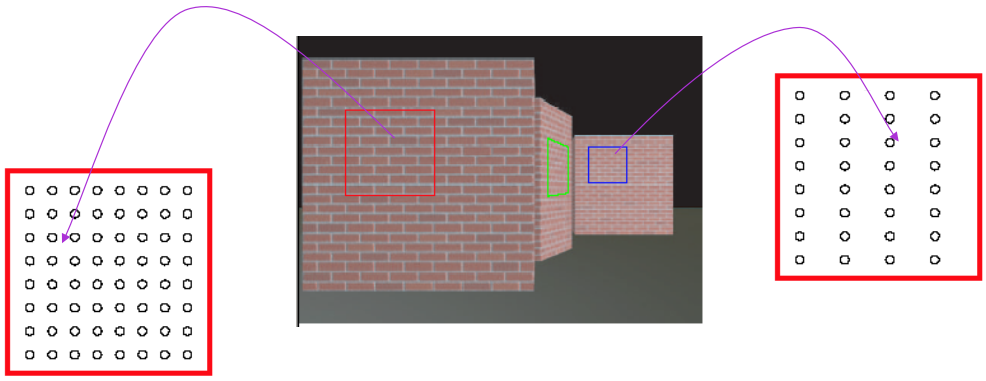
- When you go close...


When viewed from a distance

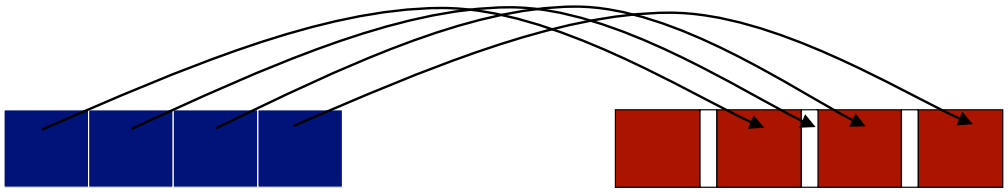
- Aliasing!
- Also, minification

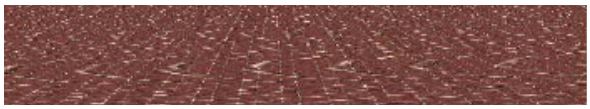
What is going on?

- Image-based texture mapping is resolution dependent



What is texture lookup doing? Sampling a single point

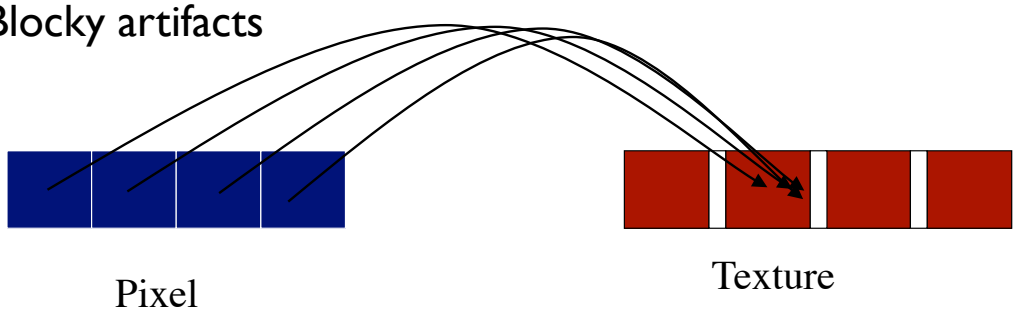




- From far away
- Scintillating artifacts over multiple frames

When viewed closer...

- Nearby pixels all lie in same texel
- Blocky artifacts

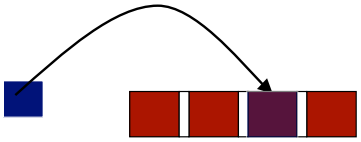


What is really the issue?

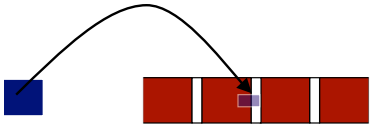
- A pixel is not a point
  - It is an area!
- Each pixel maps to some region of texture space
- Ideally, we want to integrate over mapped area

How does area map over distance?

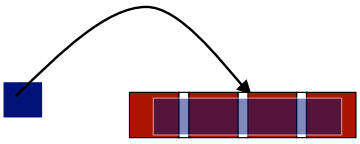
- At optimal viewing distance:
  - One-to-one mapping between pixel area and texel area



- When closer
  - Each pixel is a small part of the texel

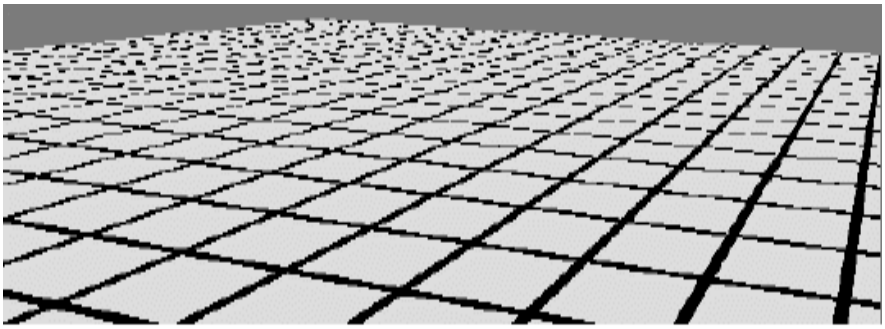


- When farther
  - Each pixel could include many texels



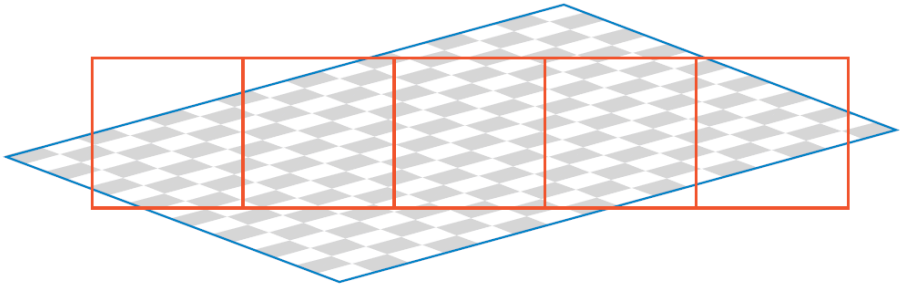
Solution: Antialiasing in textures

- Problem: Perspective produces very high image frequencies
- Solution
  - Would like to render textures with one (few) samples/pixel
  - Need to filter first!



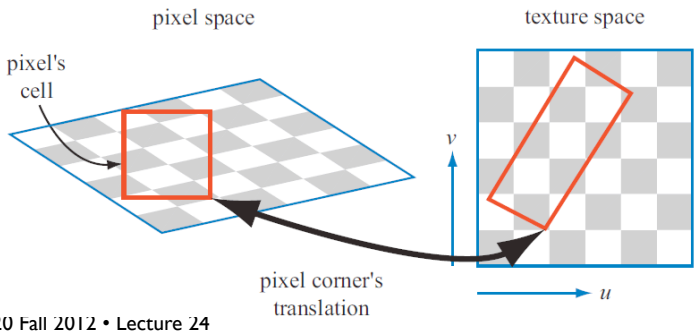
Theoretical Solution

- Find the area of pixel in texture space
- "Filter" the area to compute "average" texture color
  - Filtering eliminates high frequency artifacts



Theoretical Solution

- Find the area of pixel in texture space
- "Filter" the area to compute "average" texture color
  - Filtering eliminates high frequency artifacts
  - How to filter?
    - Analytically compute area
    - But too expensive



Solutions for Minification

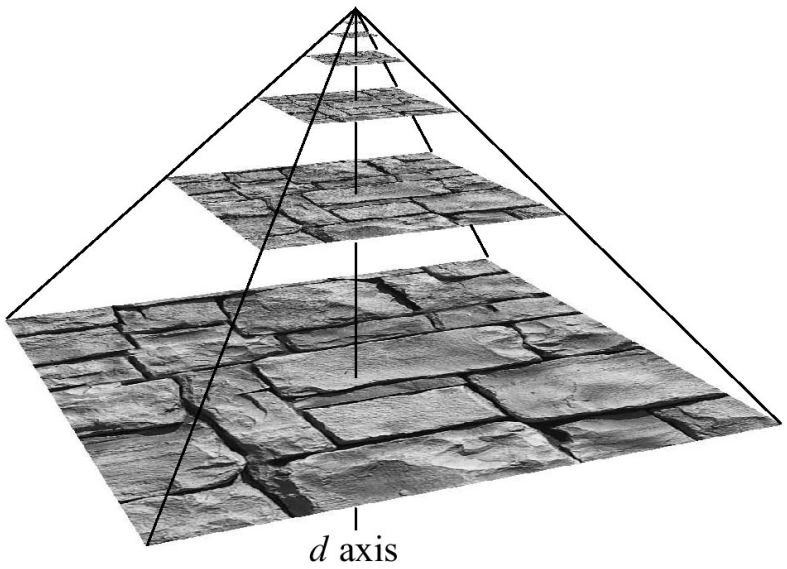
- Need some way to access pre-filtered (precomputed) regions of the texture

- MIP Maps
- RIP Maps
- Summed Area Tables
- Clip Maps

MIP Maps

- MIP Maps
  - Multum in Parvo: Much in little, many in small places
  - Proposed by Lance Williams
- Stores pre-filtered versions of texture
- Supports very fast lookup

Mipmap image pyramid

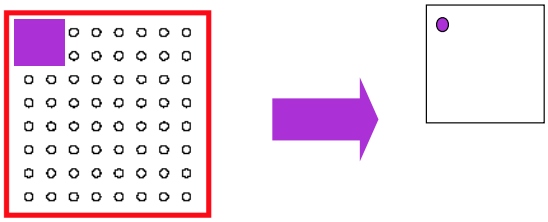


[Akenine-Möller & Haines 2002]

Some basic assumptions

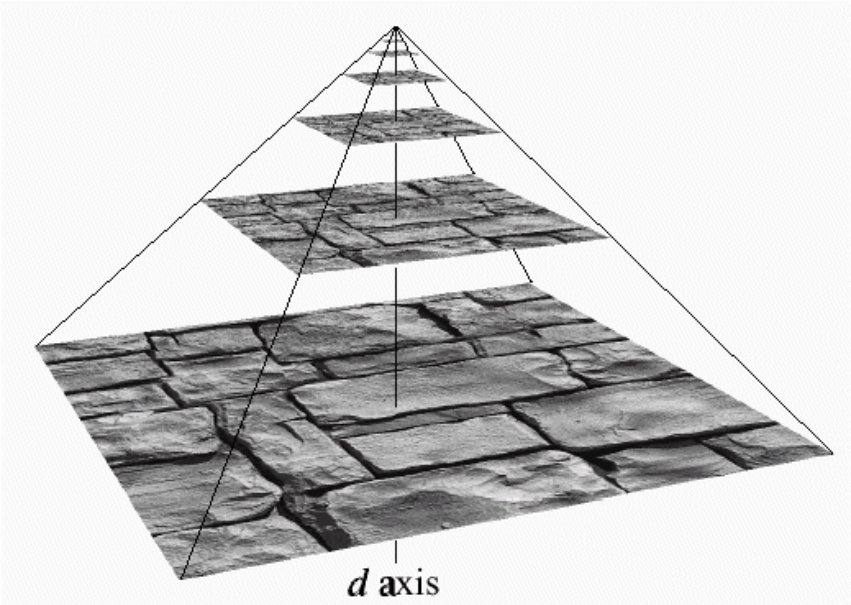
- Can't really precompute every possible required area
- But can precompute some areas

Filtering by Averaging



- Each pixel in a level corresponds to 4 pixels in lower level
  - Average
  - Gaussian filtering (more on this next lecture)

Image Pyramid



Using the MIP Map

- Find the MIP Map level where the pixel has a 1-to-1 mapping

- How?
  - Find largest side of pixel footprint in texture space
    - Pick level where that side corresponds to a texel
  - Compute derivatives to find pixel footprint
    - Intuition for derivatives

Using the MIP Map

- Find the MIP Map level where the pixel has a 1-to-1 mapping

- How?
  - Find largest side of pixel footprint in texture space
    - Pick level where that side corresponds to a texel
  - Compute derivatives to find pixel footprint
    - x derivative: $\frac{\partial u}{\partial x}$ $\frac{\partial v}{\partial x}$
    - y derivative: $\frac{\partial u}{\partial y}$ $\frac{\partial v}{\partial y}$

Given derivatives: what is level?

$$level = log[max(\frac{du}{dx}, \frac{dv}{dx}, \frac{du}{dy}, \frac{dv}{dy})]$$

$$level = log\sqrt{(\frac{du}{dx})^2 + (\frac{dv}{dx})^2 + (\frac{du}{dy})^2 + (\frac{dv}{dy})^2}$$

- Gradients
  - Available in pixel shader (except where there is dynamic branching)

