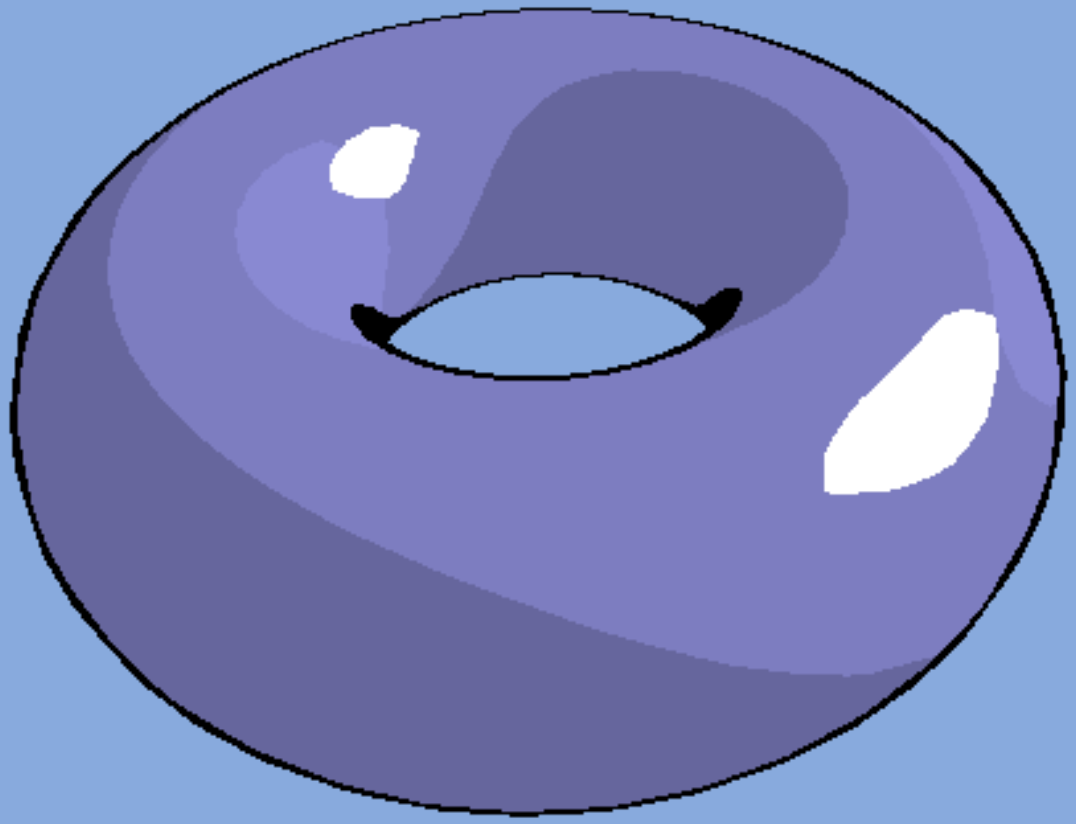


Shaders

also today: particle system review

CS 4300/5310

Computer Graphics

ANNOUNCEMENTS

Upcoming Deadlines

- HW4: Shaders
  - March 19th
- Reading Response
  - March 21st, **10:30am**
- Final Project Proposal
  - March 22nd

Northeastern Game Demo Day

On April 19 the Huskies will come out and play

Will you give them something to play with?

Go to **www.northeastern.edu/games/demo** and register now!

Northeastern University
College of Arts, Media and Design

PLAIT

Northeastern Center for the Arts

Northeastern University
College of Computer and Information Science

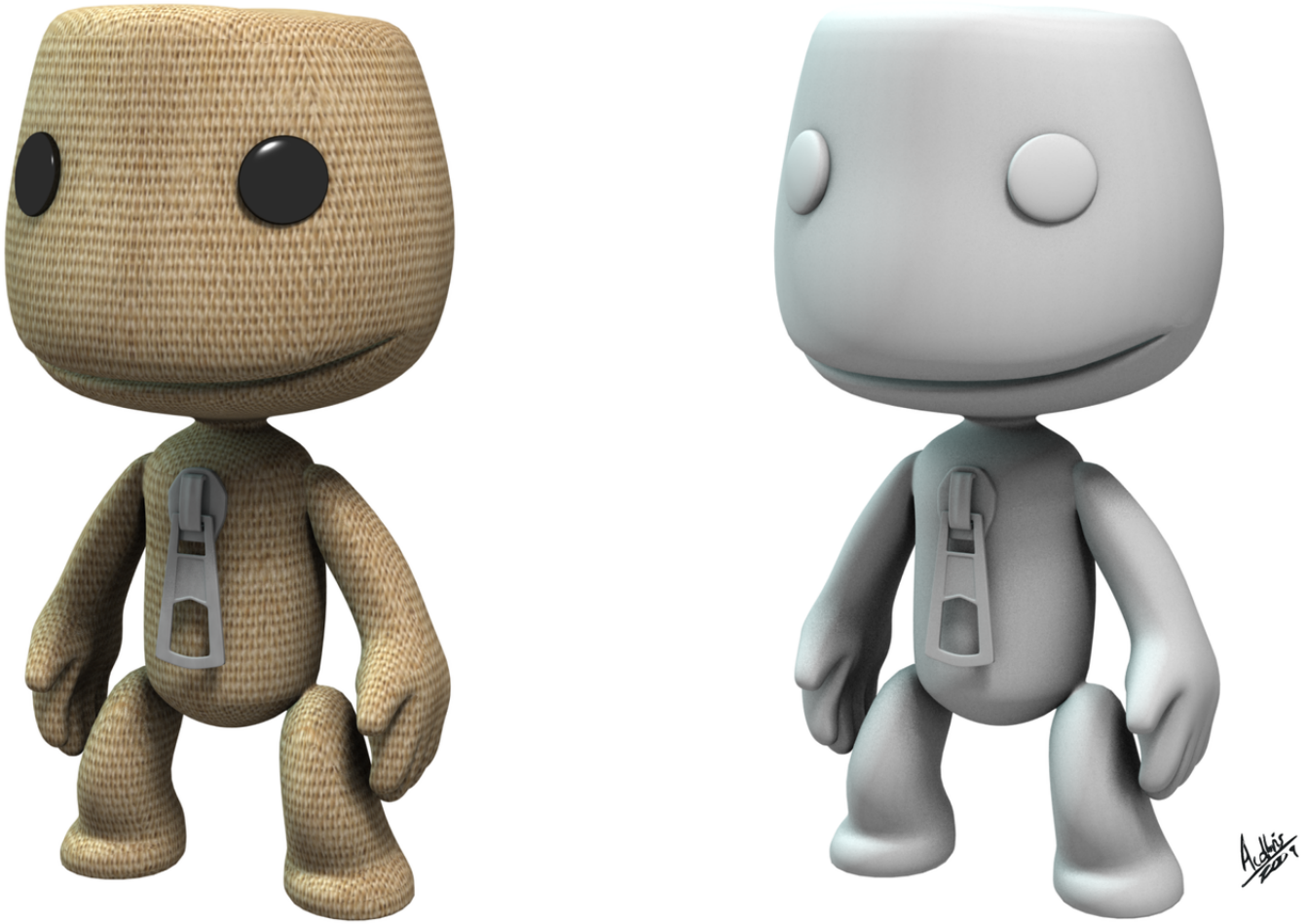
who wants to share?

PARTICLE SYSTEMS REVIEW

TEXTURING

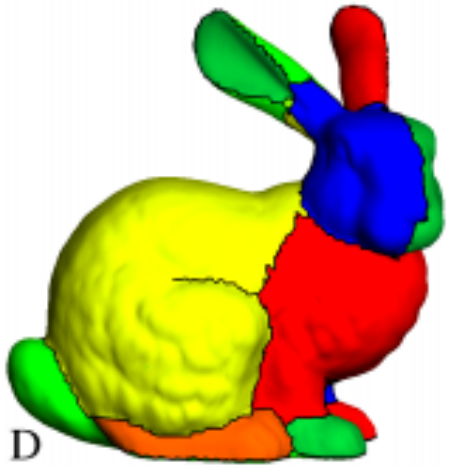
TEXTURING

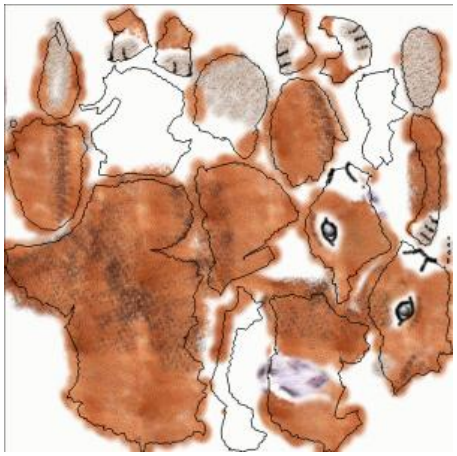
What is Texturing?



http://blenderartists.org/forum/showthread.php?174915-SackBoy-Model-(W-I-P)

Texturing Methods: Atlas



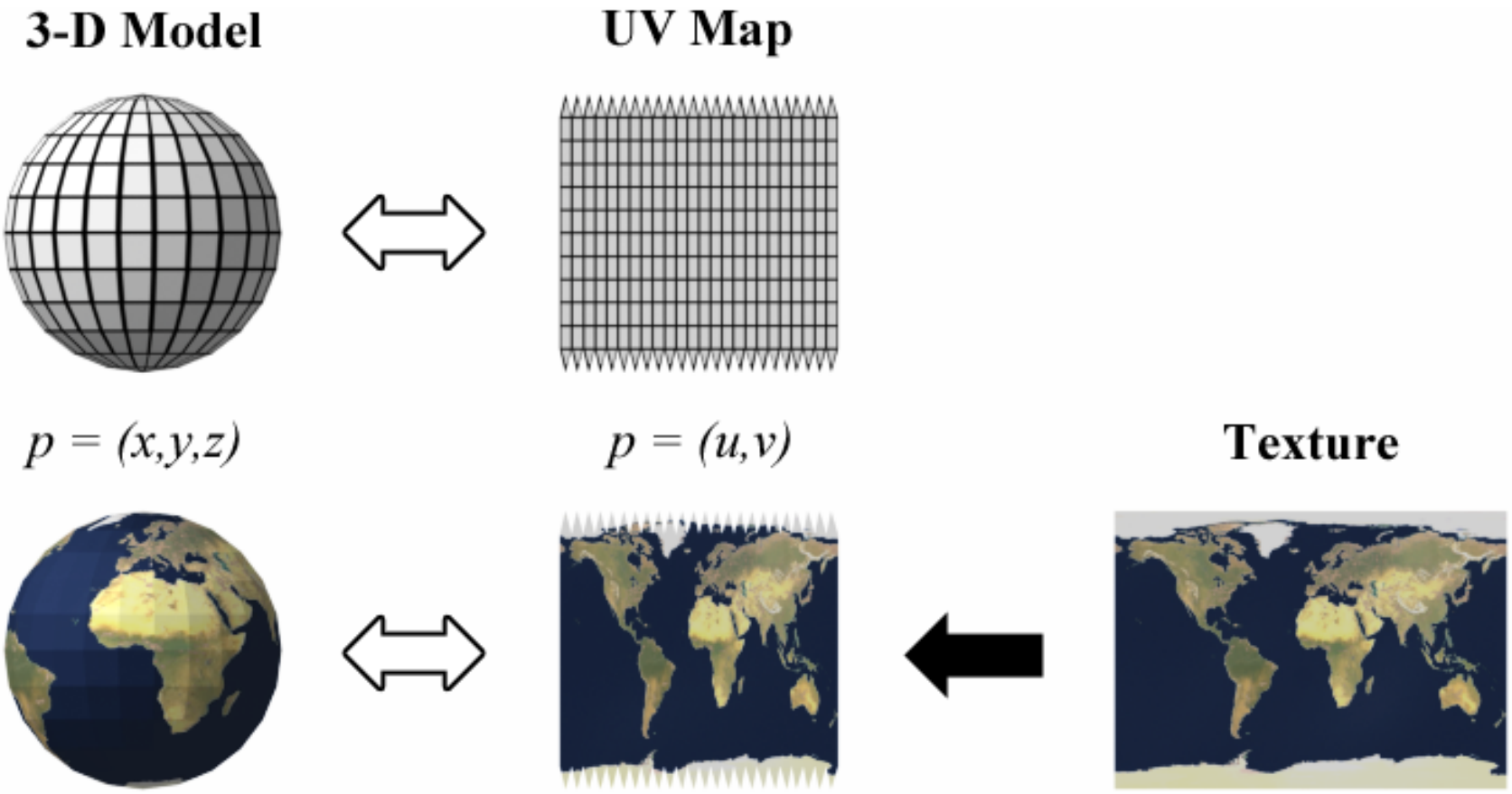


Levy, B; Petitjean, S.; Nicolas, R; Maillot, J. Least Squares Conformal Maps for Automatic Texture Atlas Generation. ACM SIGGRAPH 2002.

Texturing Methods: Unwrap the Model

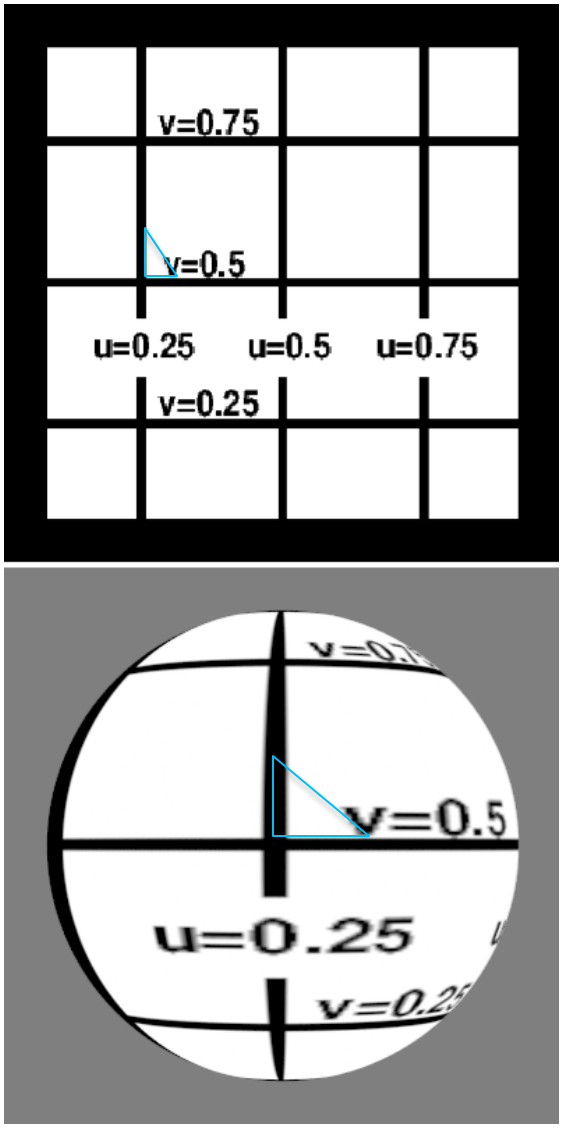
http://www.bob-marshall.co.uk/2011/02/horsing-around-in-zbrush-work-in.html

uv mapping

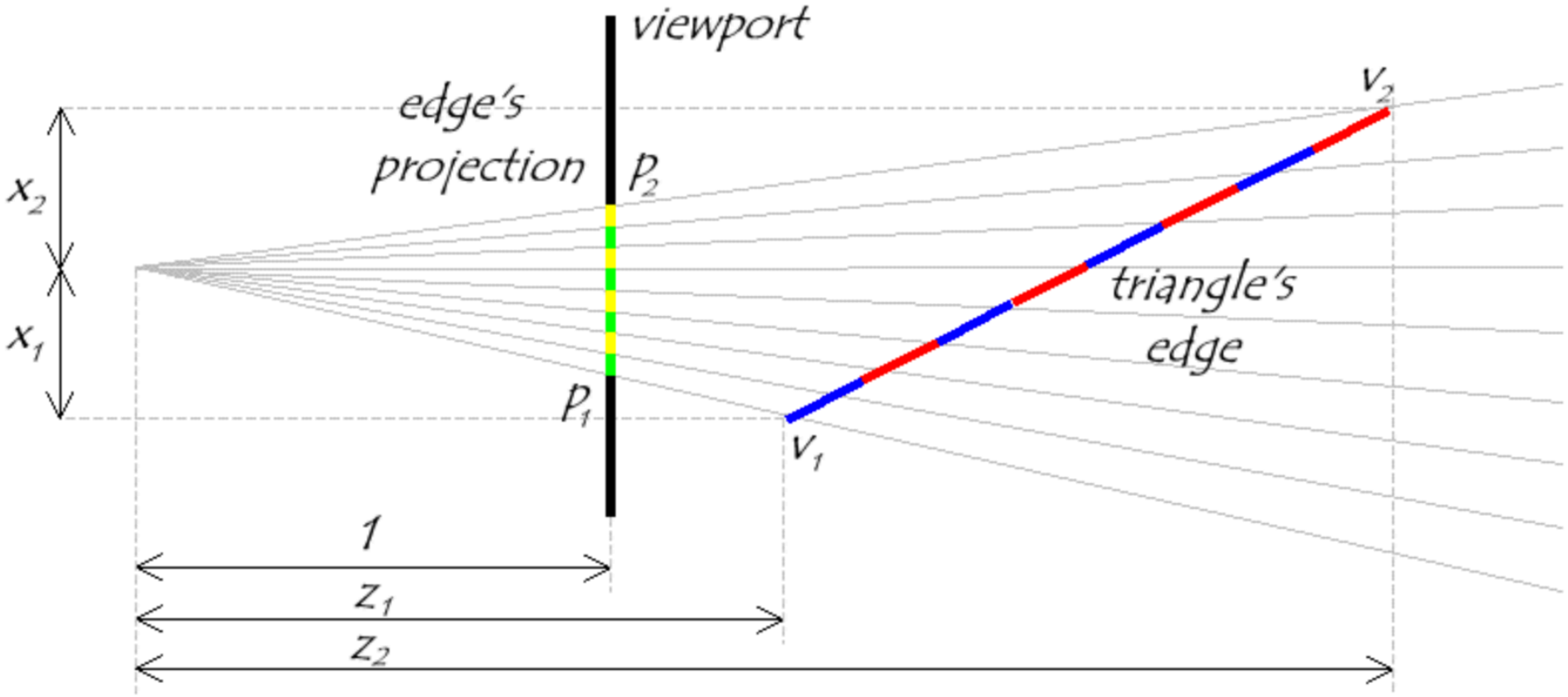


uv mapping

```
for all x:
    for all y:
        compute barycentric coords
        if (textured):
            diffuse color = texture(β, γ)
        else:
            diffuse color = interpolated
            from vertices
```



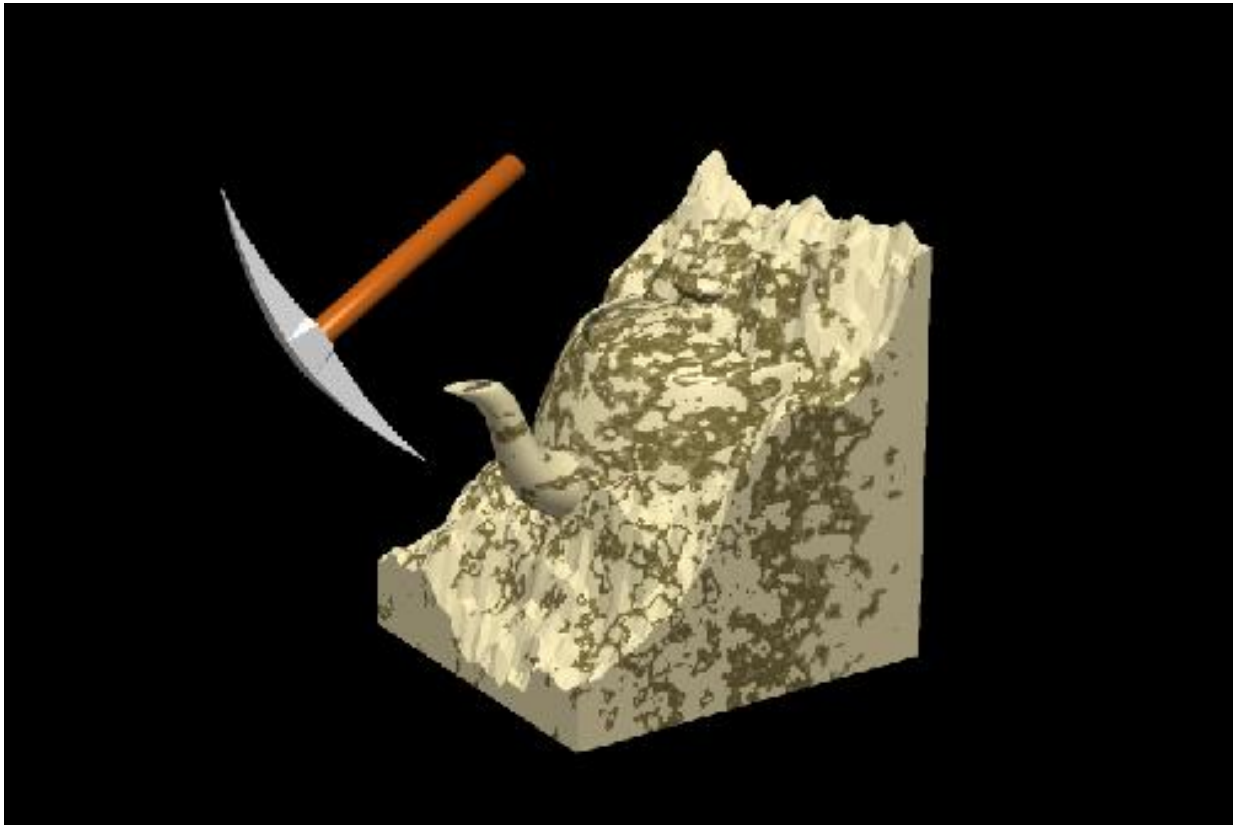
uv mapping: perspective correction



Uniform steps in screen space ≠ uniform steps in world space!

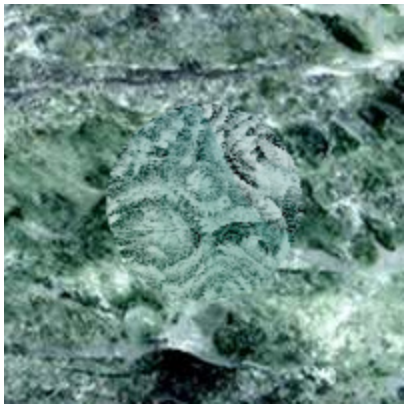
http://groups.csail.mit.edu/graphics/classes/6.837/F98/Lecture21/Slide08.html

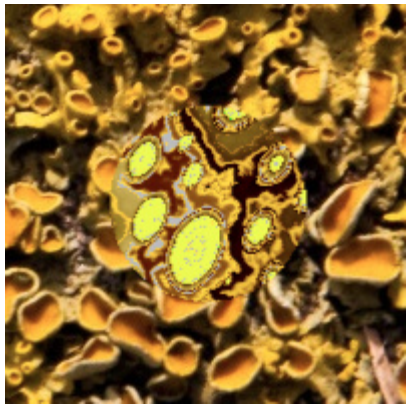
Texturing Methods: 3D Textures

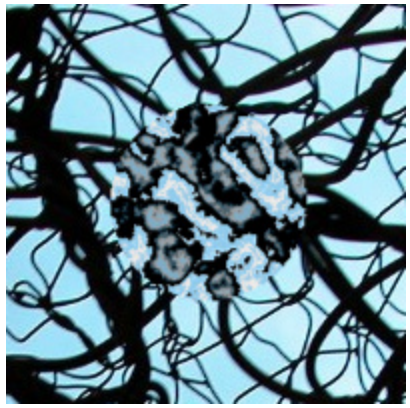


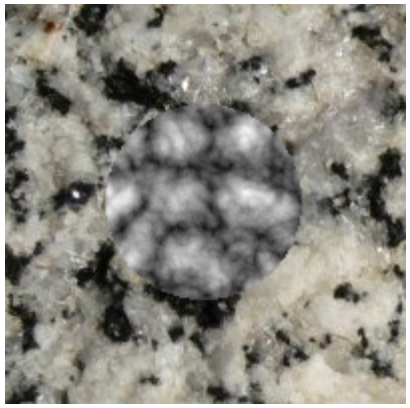
http://www.siggraph.org/education/materials/HyperGraph/mapping/r_wolfe/r_wolfe_mapping_6.htm

Texturing Methods: Procedural Textures







Craig Reynolds, Interactive Evolution of Camouflage: http://www.red3d.com/cwr/texsyn/diary.html

Texturing Approaches: What's In Common?

- Data stored as an image (or set of images)
- Artist control?
- Programmer convenience?
- Mapping from XYZ space to UV(W) space

the programmable graphics pipeline

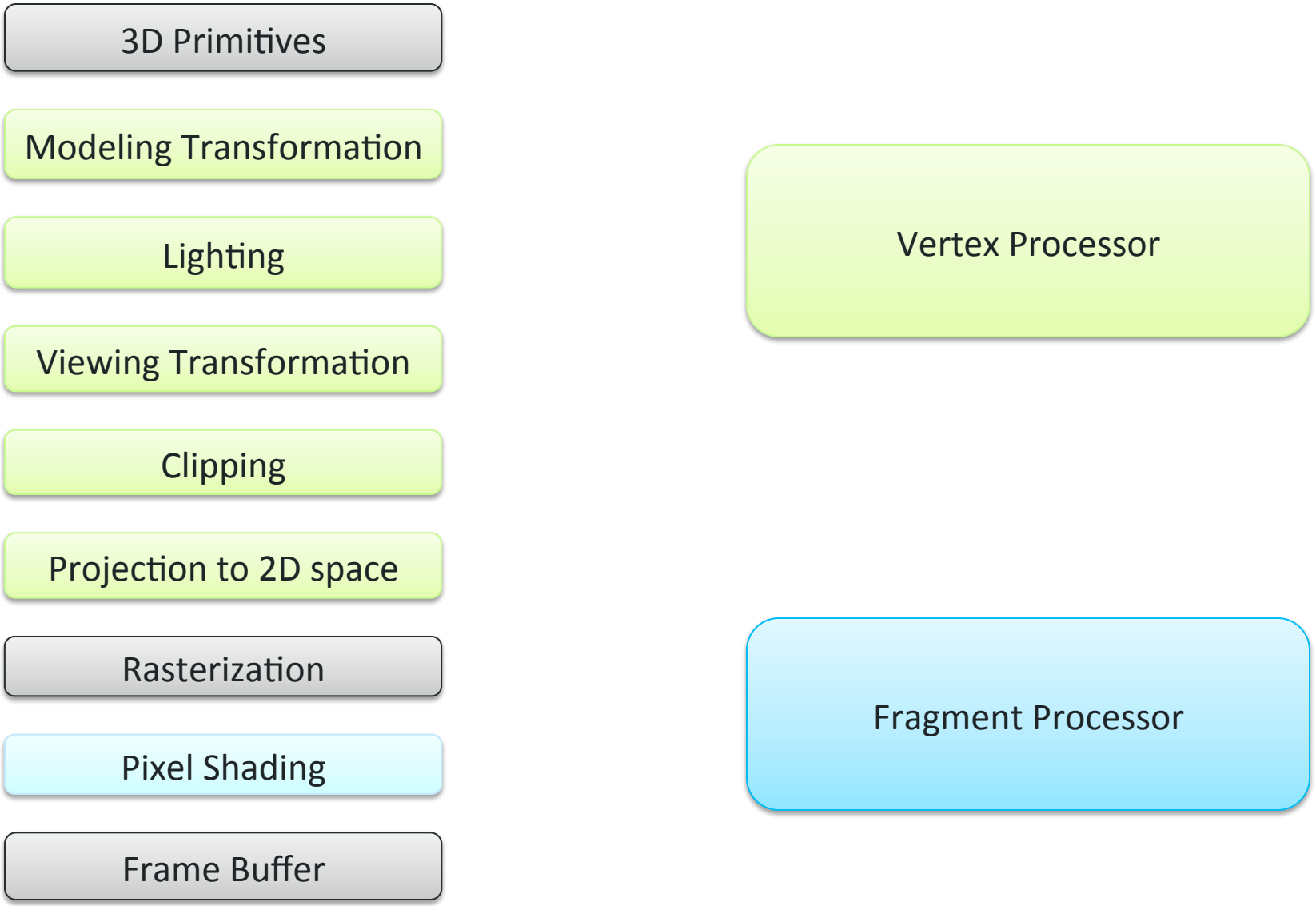
SHADERS

Review: The Graphics Pipeline

- 3D Primitives
- Modeling Transformation
- Lighting
- Viewing Transformation
- Clipping
- Projection to 2D space
- Rasterization
- Pixel Shading
- Frame Buffer

What happens in each of these stages?

The Programmable Graphics Pipeline



What does the graphics card look like?

- NVIDIA GTX 680
  - 1536 “CUDA cores”
  - 8 Streaming Multiprocessors (192 processors per)
  - 2048MB memory
  - 128 texture units

Vertex Shaders

- Operate on each vertex simultaneously(ish)
- What might be the inputs and outputs?

Vertex Shaders

- Operate on each vertex simultaneously(ish)

- Typical Inputs
  - Vertex positions (world space)
  - Transformation matrices
  - Vertex color
  - Vertex normal
  - User defined…
- Typical Outputs
  - Vertex position + depth (screen space)
  - Vertex color
  - Texture coordinate
  - Normal
  - User defined…

Tesselation Shaders

- Very new!
- Support subdividing surfaces into more triangles
- Why is this useful?

Geometry Shaders

- Slightly less new but still pretty new!

- Access to add new vertices
- Not limited to only triangles

Pixel Shaders

- Also commonly called “fragment” shaders
  - Why??
- Operate on each fragment simultaneously(ish)
- Inputs?
- Outputs?

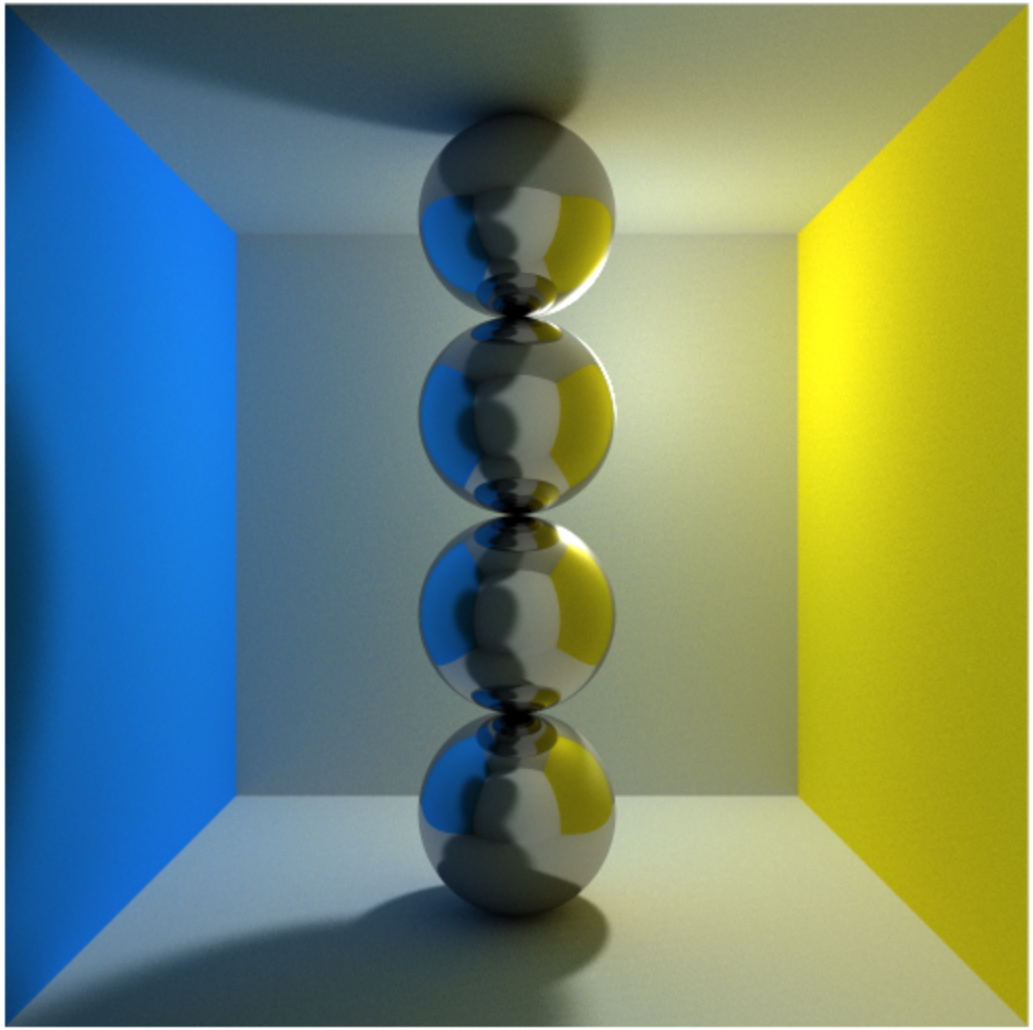
Focus: Vertex and Fragment Shaders

- We can still do a lot with these!
- Using GLSL: OpenGL Shading Language
- Vertex: Manipulate existing geometry
- Fragment: Manipulate individual pixel colors

GLSL Language Features

- Swizzle
  - new_vec.zyxw = vec.xyzw
- Variable types
  - Uniform
  - Attribute
  - Varying
- Lots of built-in and optimized math
  - sin, cos, dot, cross…
  - vec2, vec3, vec4, mat3, mat4…

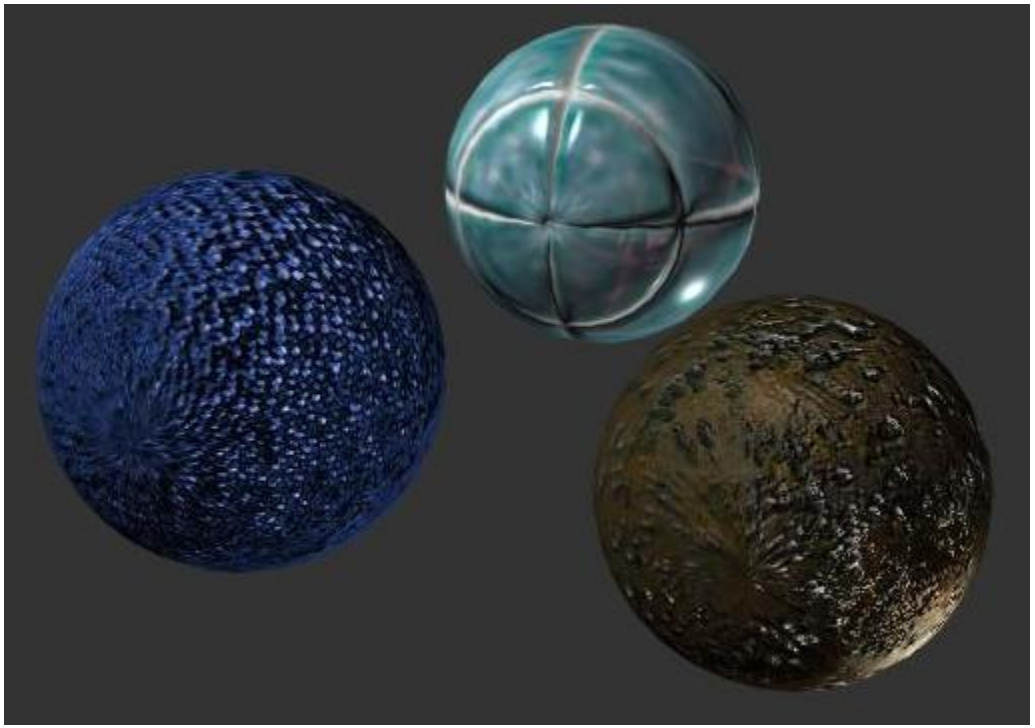
Shader Examples: Path Tracing

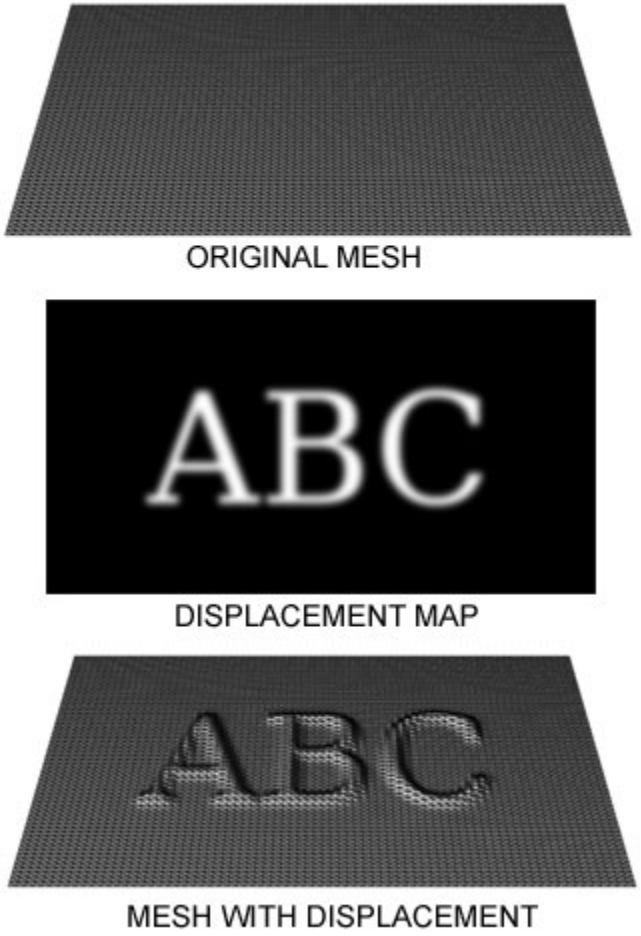


http://madebyevan.com/webgl-path-tracing/

Shader Examples: Better Texturing

- Bump Mapping vs. Displacement Mapping





GLSL in Processing

- Conway's Game of Life: Cellular Automata
  - Cell with less than two neighbors will die
  - Cell with two-three neighbors will live
  - Cell with more than three neighbors dies
  - A dead cell with exactly three neighbors comes alive

GPGPU

- GPU is basically just a stream processor, which is useful for lots of things outside graphics!

- Breaking encryption
- Biomedical applications [human genome sequencing!]
- Physics simulations
- Audio processing
- …other ideas?

writing simple shaders

ASSIGNMENT 4

requirements; topic brainstorming

FINAL PROJECT

Project Structure

- 2-3 person groups
  - Prefer not mixing grad/undergrad
- Proposal: March 22nd
- Checkpoints: In class March 28th, April 4th
- Presentations: In class April 9th, 11th, 16th
- Report and Video: April 19th

Suggested Topics

- Extension to Raytracer
  - More complex geometry
  - Experiment with different camera types
  - Texturing and/or bump mapping
- Interactive Applications
  - Make a game
  - 3D Data Visualization
  - Interactive Art

Project Goals

- More focused exploration of 3D topics
  - Longer term project than 2D
  - Time for deeper exploration and reflection
- Technical writing and writing critique
- Project critique and peer code review

Writing Requirements

- Project proposal: 3-5 pages
  - What are you going to do?
  - How are you going to do it?
  - Why is it interesting? What is the problem you are trying to solve?
  - What are some related projects?
- Project report: 6-8 pages
  - What did you do?
  - How was it related to what other people have done?
  - What were the results?

Project Brainstorming Activity

- Assignments form a “sampler” of 3D graphics
  - Raytracing
  - Interactive 3D
  - Shaders

- What do you find interesting?
- What do you wish your assignment could do that it can’t?
- What do you wish you knew more about?