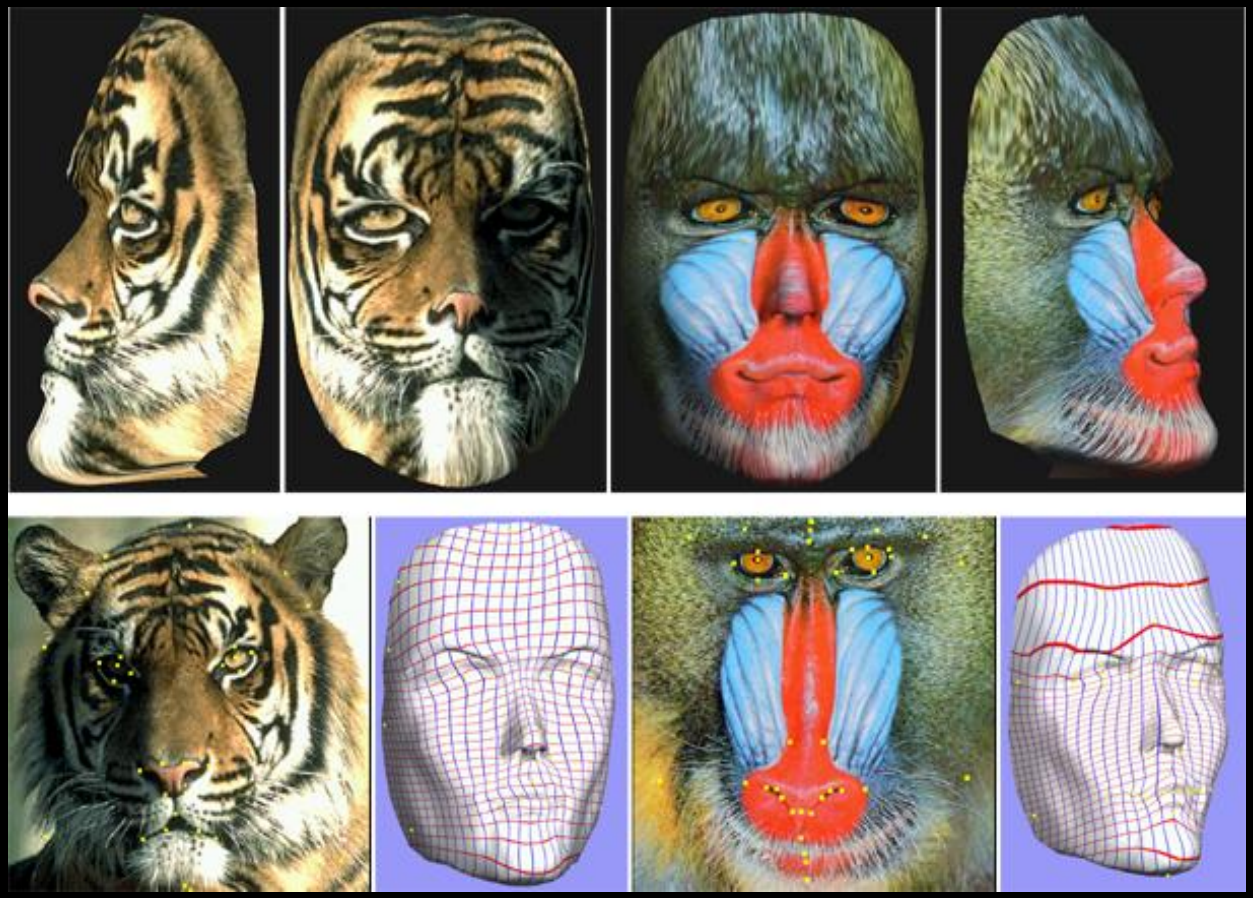
Levy: Constraint Texture Mapping, SIGGRAPH, 2001



Textures and Shaders

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon

Updated Staff/Office Hours

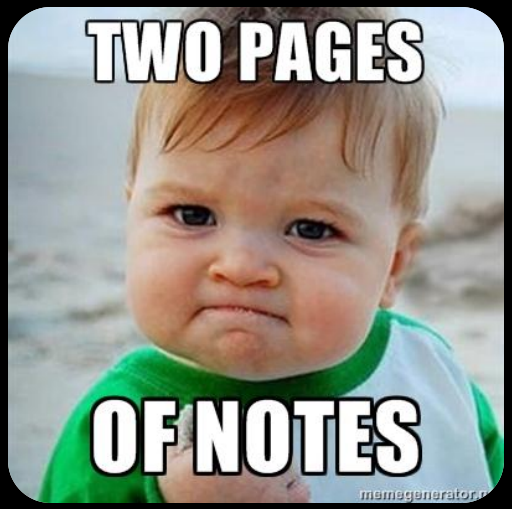
Instructor: Justin Solomon
Email: justin.solomon@stanford.edu
Office: Clark S297
Office hours: Tuesdays, 9am-11am;
Fridays, 2:15pm-4:15pm

but not this Friday!

CA: Blake Carpenter
Email: blakec@stanford.edu
Office hours: Wednesdays, 2pm-4pm;
Thursdays, 1pm-2pm (**Gates B24**)

CA: Taesung Park
Email: taesung@stanford.edu
Office hours: Mondays, 2:15pm-4:15pm (Gates B24)

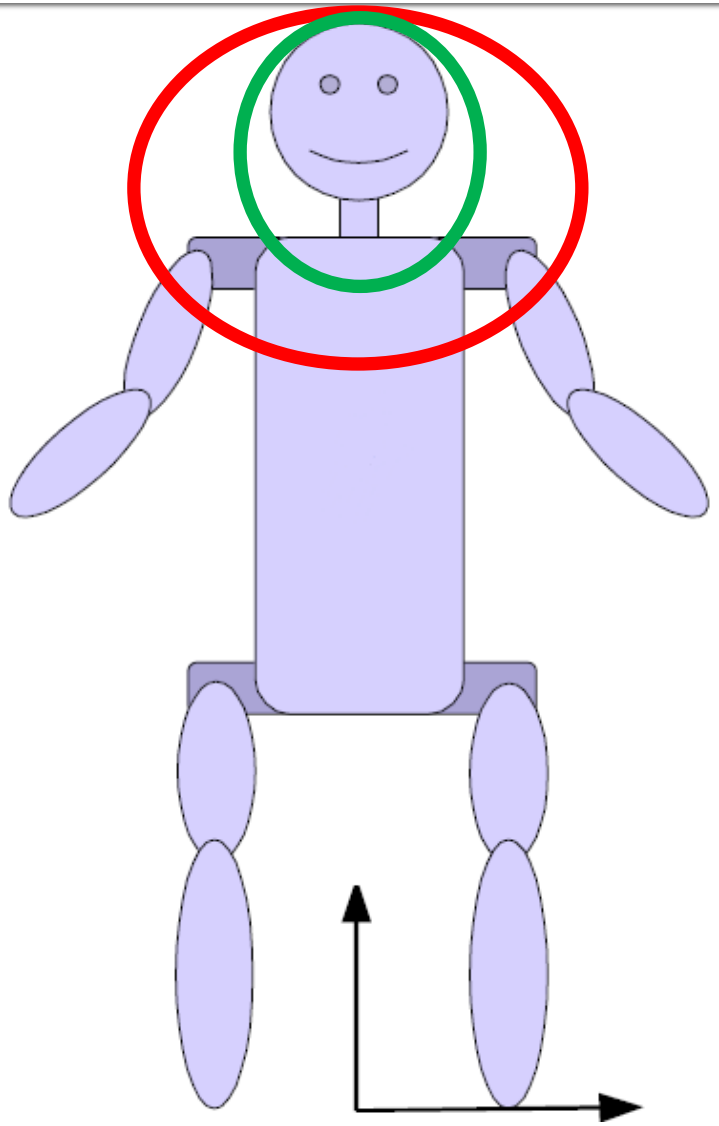
July 18, in class



July 18, in class

<leftover>

Hierarchical Modeling



```
glTranslatef(0,1.5,0);
drawTorso();
glPushMatrix();
 glTranslatef(0,5,0);
 drawShoulder();
 glPushMatrix();
  glRotatef(neck_y,0,1,0);
  glRotatef(neck_x,1,0,0);
  drawHead();
 glPopMatrix();
 glPushMatrix();
  glTranslatef(1.5,0,0);
   glRotatef(l_shoulder_x);
   drawUpperArm(); ...
```

http://graphics.stanford.edu/courses/cs148-10-summer/docs/07_rendering2.pdf

OpenGL Transformation Model

$$T = M_{projection} \cdot M_{camera} \cdot M_{model}$$

Camera coordinates to normalized device coordinates

GL_PROJECTION

Local coordinates to camera coordinates

GL_MODELVIEW

```
glMatrixMode(GL_PROJECTION);
glMatrixMode(GL_MODELVIEW);
```

Applied to all geometry

OpenGL Transformation Model

$$T = M_{projection} \cdot M_{camera} \cdot M_{model}$$

Camera coordinates to normalized device coordinates

Local coordinates to camera coordinates

GL_PROJECTION **GL_MODELVIEW**

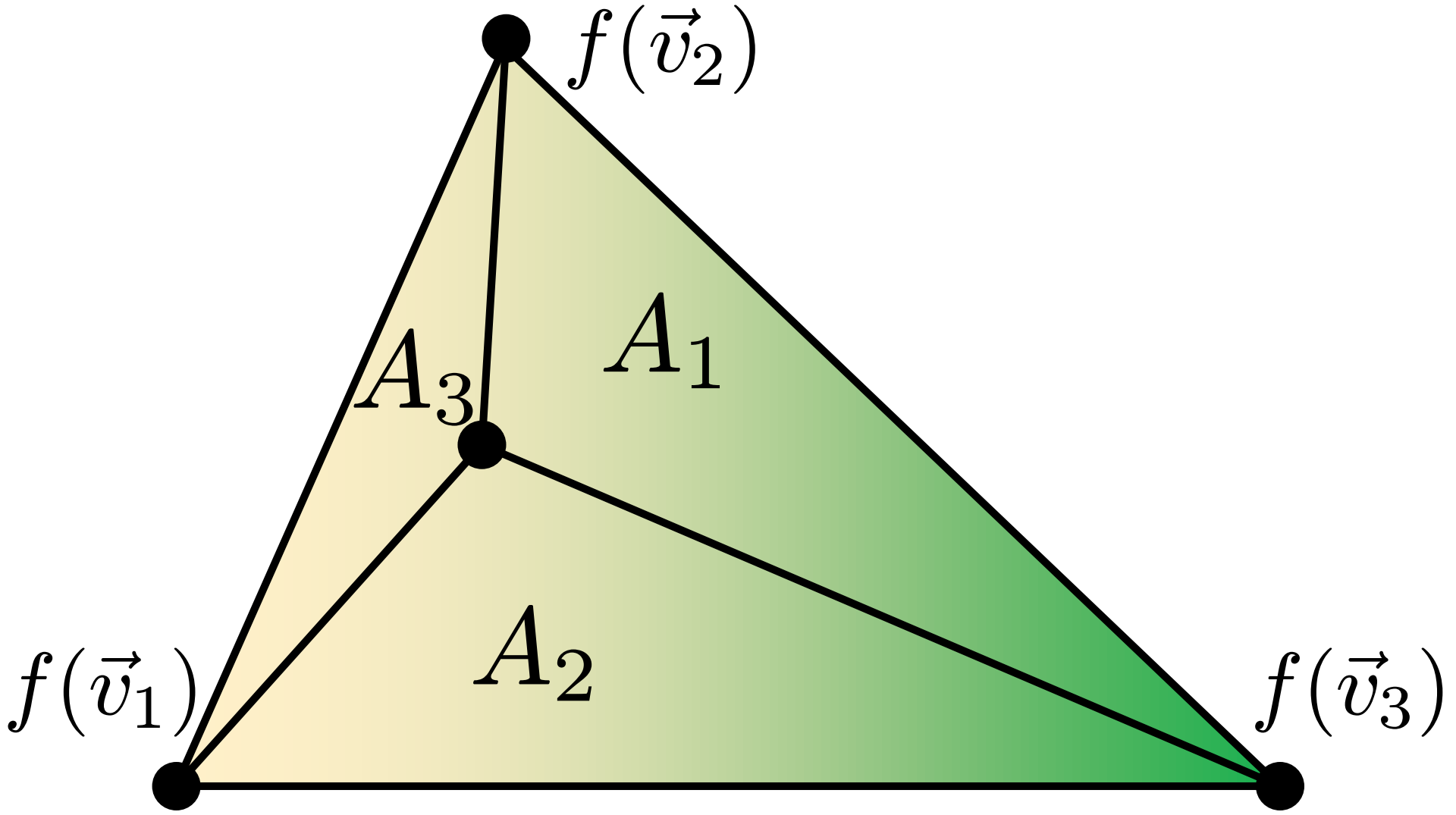
Can do whatever you want with these.

```
glMatrixMode(GL_PROJECTION);
glMatrixMode(GL_MODELVIEW);
```

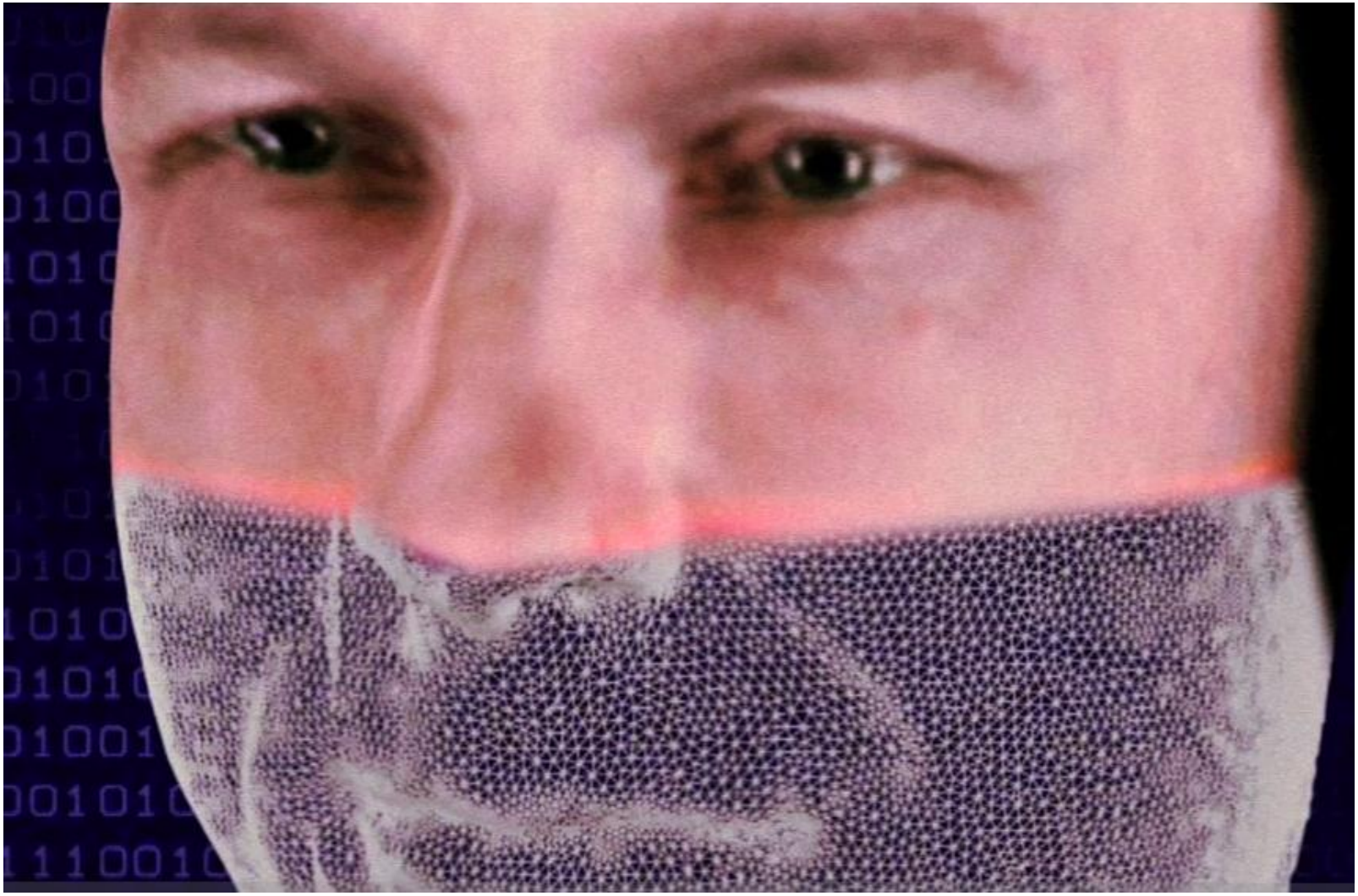
Applied to all geometry

</leftover>

Per-Vertex Coloring



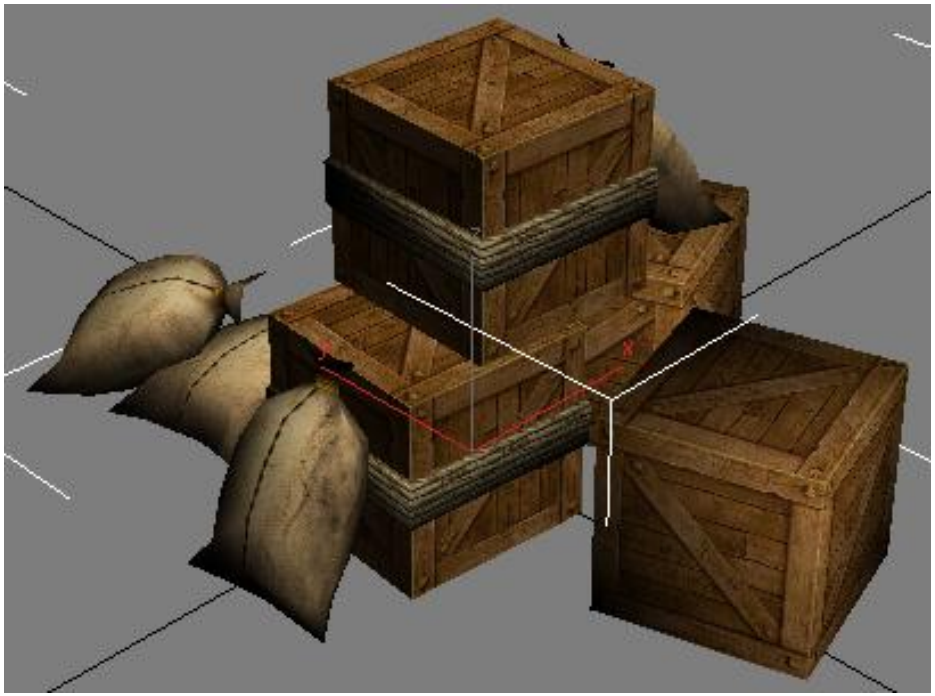
Per-Vertex Coloring

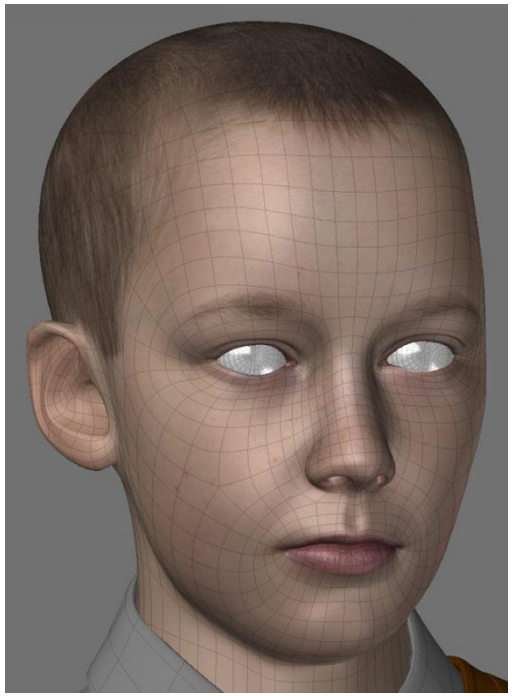


http://www.3dface.org/images/download/3D-Scan.png

Problem

http://dev.ryzom.com/projects/ryzom/wiki/ImportingMaxAssets
http://www.ten24.info/upload/pages/Head%20Texture.jpg



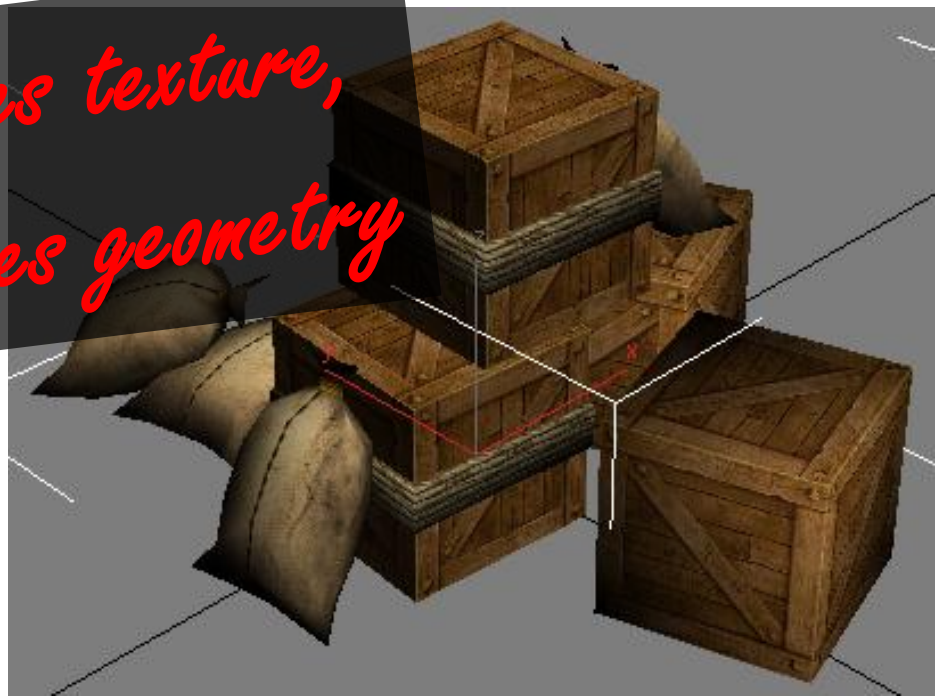


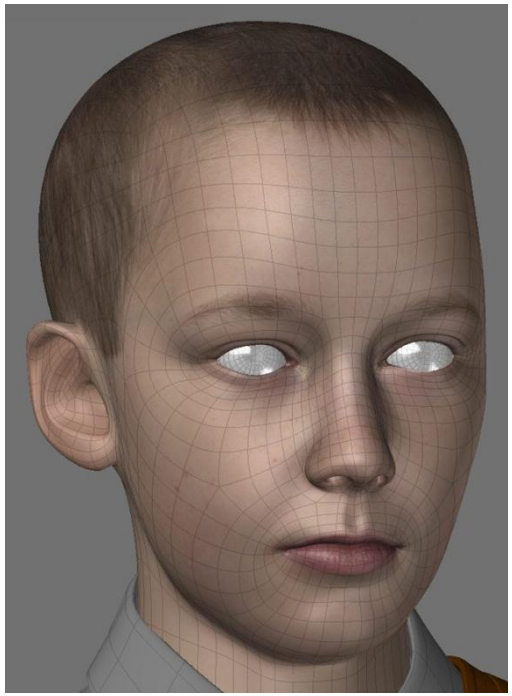
Texture and geometry resolution might not coincide.

Problem

http://dev.ryzom.com/projects/ryzom/wiki/ImportingMaxAssets
http://www.ten24.info/upload/pages/Head%20Texture.jpg

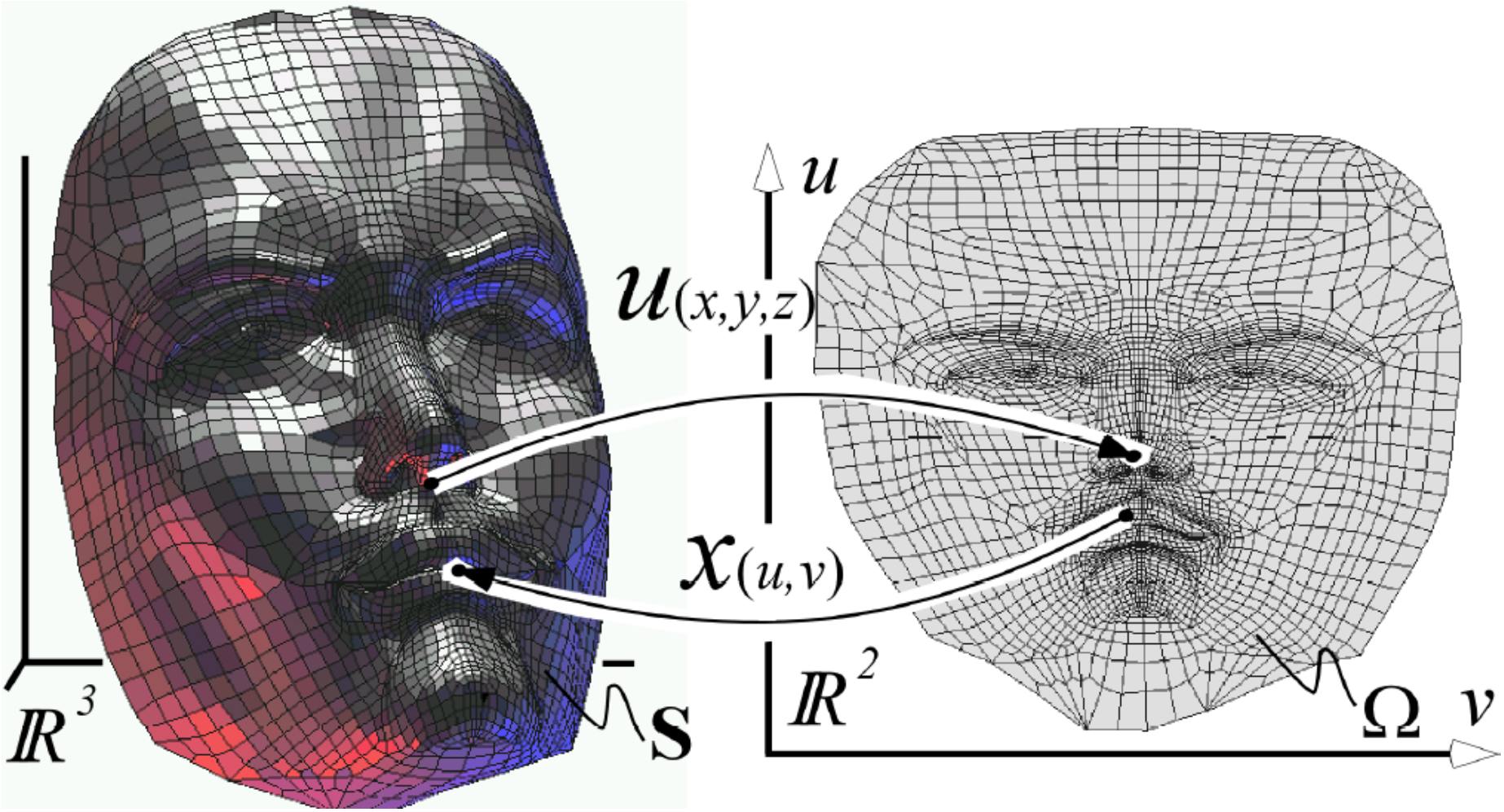
Hi-res texture,
Lo-res geometry





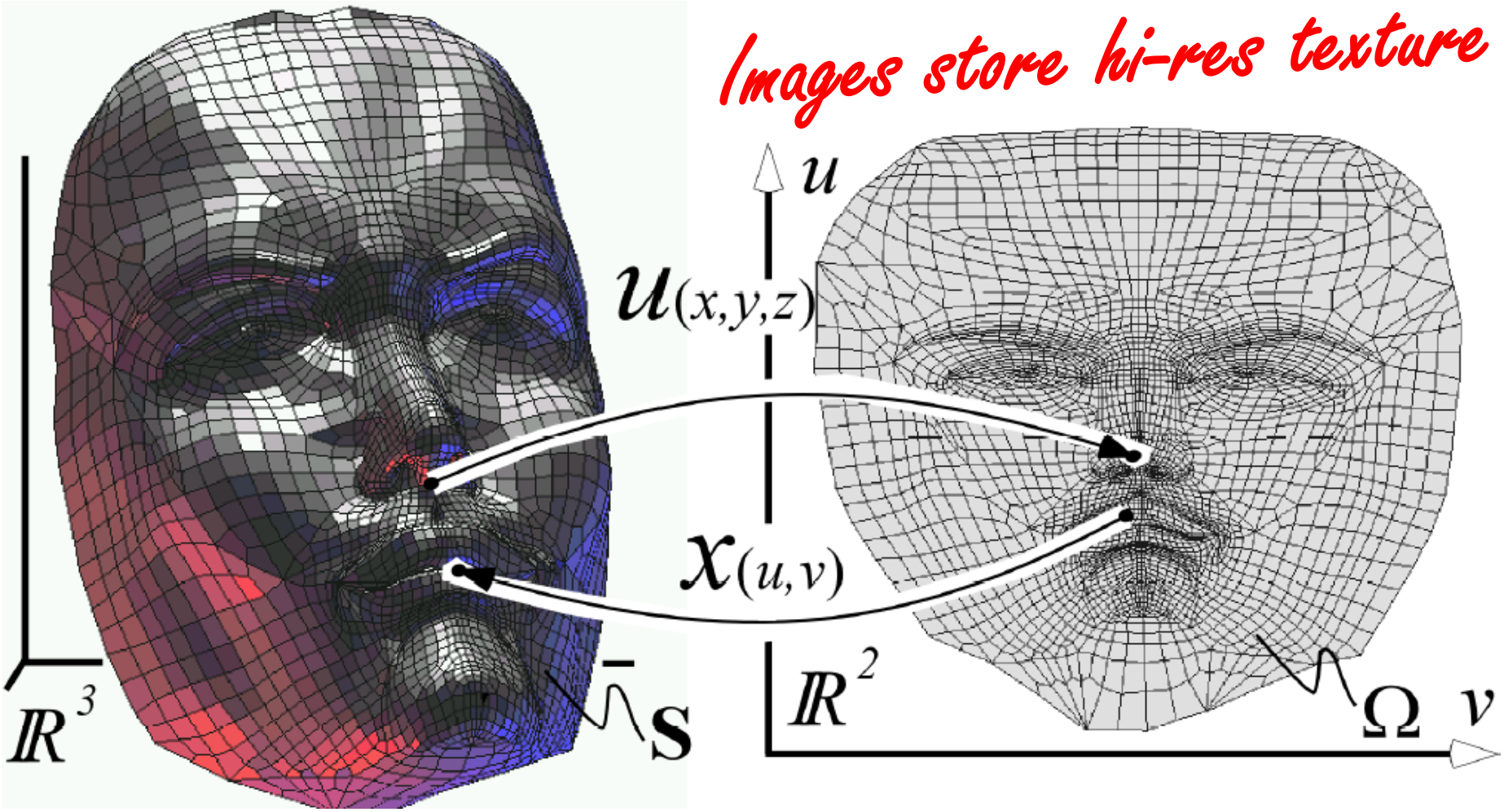
Texture and geometry resolution might not coincide.

Texture Coordinates



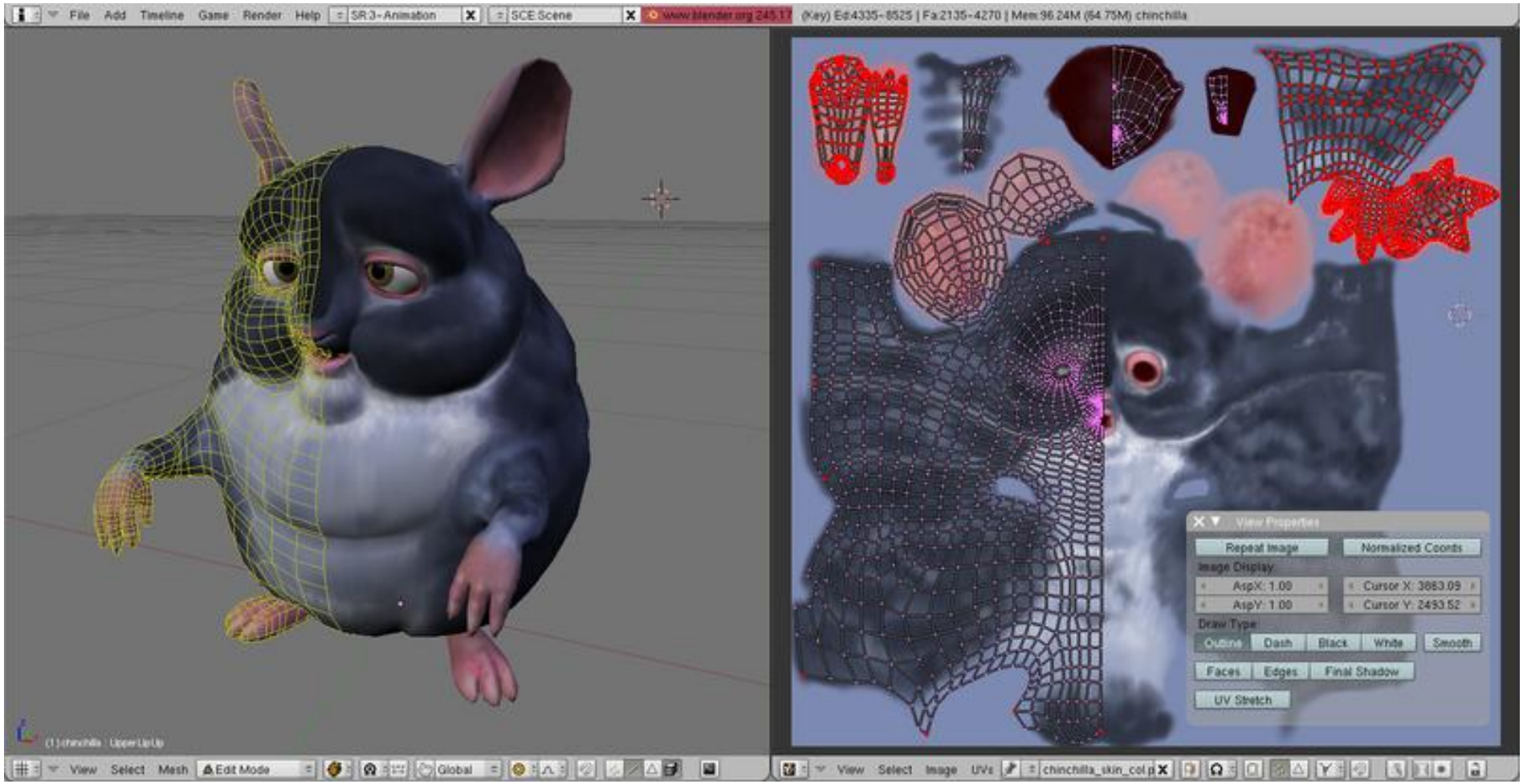
http://graphics.stanford.edu/courses/cs468-10-fall/LectureSlides/12_Parameterization1.pdf

Texture Coordinates



http://graphics.stanford.edu/courses/cs468-10-fall/LectureSlides/12_Parameterization1.pdf

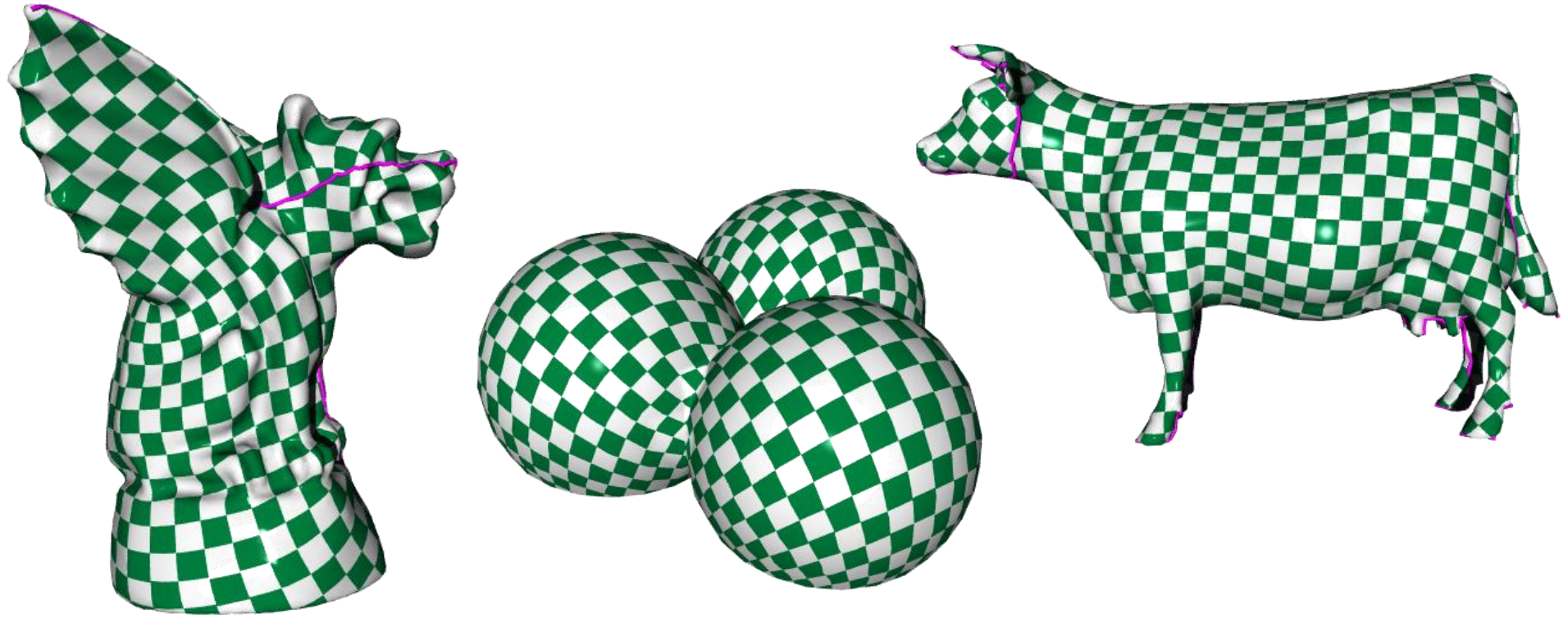
Topological Issues: Texture Atlas



http://www.blender.org/development/release-logs/blender-246/uv-editing/

(aside!)

Research in Parameterization

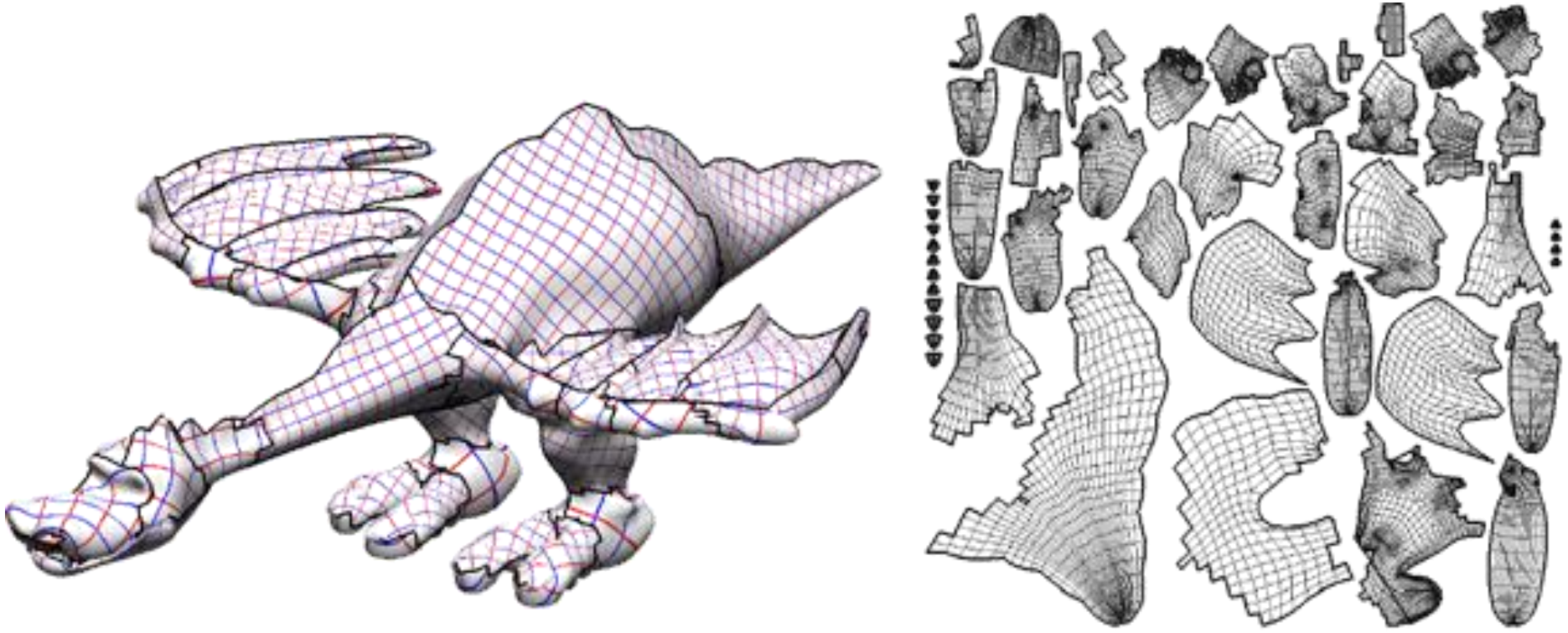


http://www.math.zju.edu.cn/cagd/Projects/ARAPPara/Paper/SGP08_ARAPPara.pdf

Trade off between area and angle preservation

(aside!)

Research in Parameterization

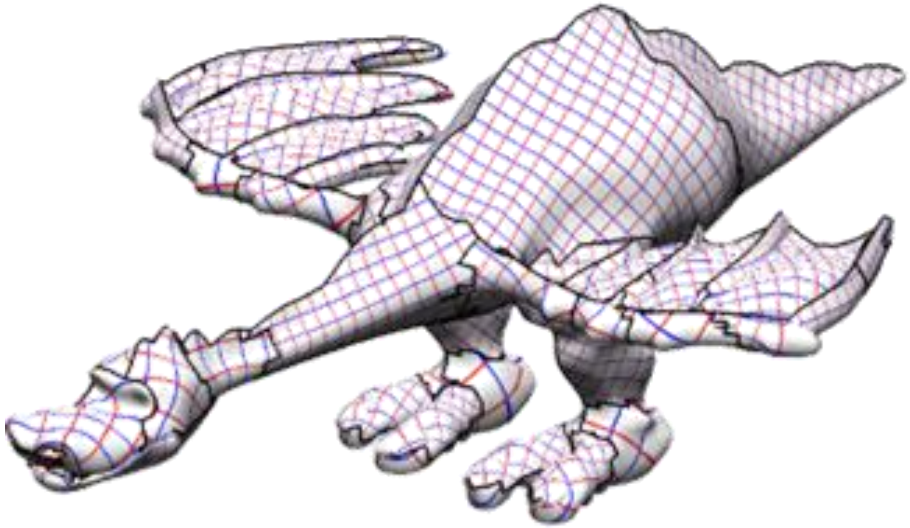


Levy, Petitjean, Ray, Maillot: Least Squares Conformal Maps for Automatic Texture Atlas Generation

Deal with high genus, distortion

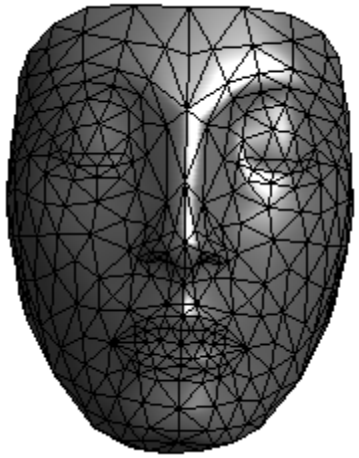
Representation

- Each vertex in each triangle stores **(*u*,*v*)** texture coordinates
- **Same vertex in adjacent triangles** might have different (*u*,*v*)



(aside!)

Barycentric Parameterization

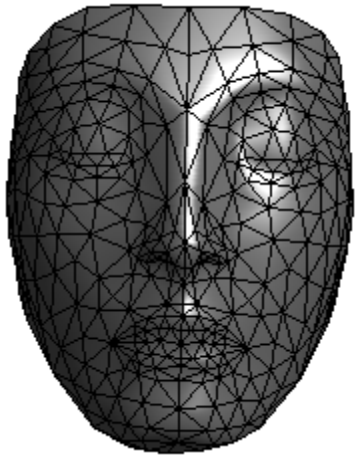


Goal: Assign (u,v) coordinate to each mesh vertex.

http://www.ceremade.dauphine.fr/~peyre/matlab/graph/content_09.png

(aside!)

Barycentric Parameterization



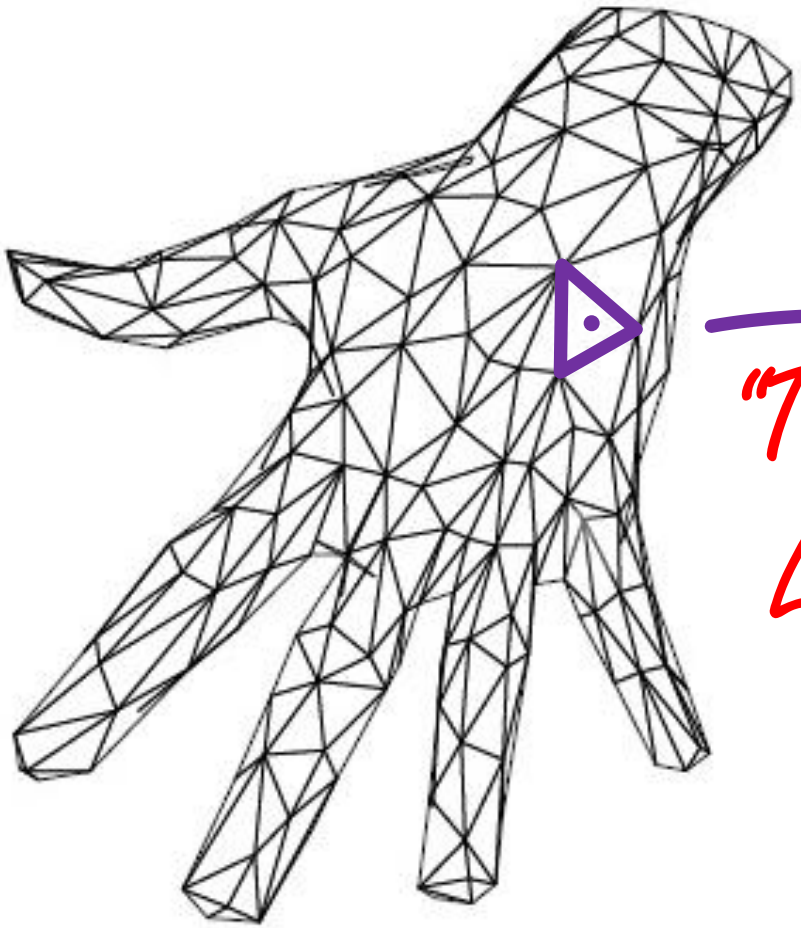
Goal: Assign (*u*,*v*) coordinate to each mesh vertex.

1. Fix (*u*,*v*) coordinates of boundary.
2. Want interior vertices to be at the (bary)center of their neighbors:

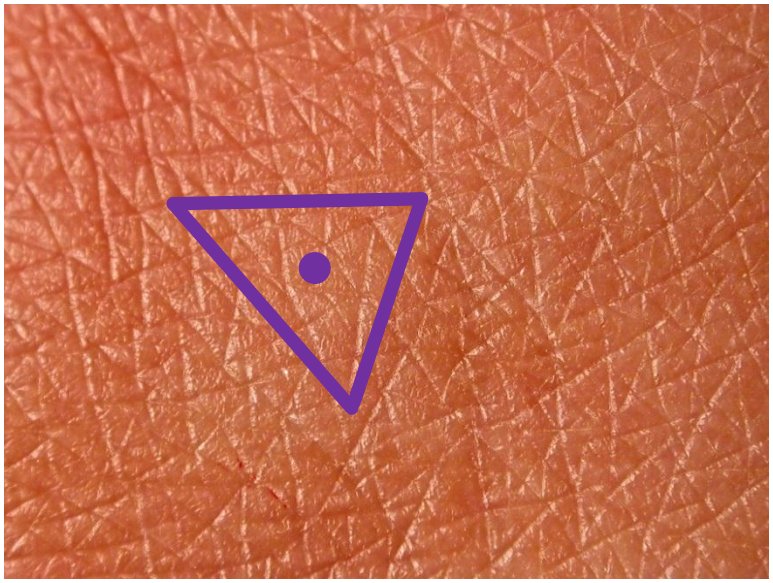
$$\mathbf{v}_i = \frac{1}{\text{valence}(i)} \sum_{(i,j)\ \text{neighbors}} \mathbf{v}_j$$

http://www.ceremade.dauphine.fr/~peyre/matlab/graph/content_09.png

Rasterizing Textured Triangles

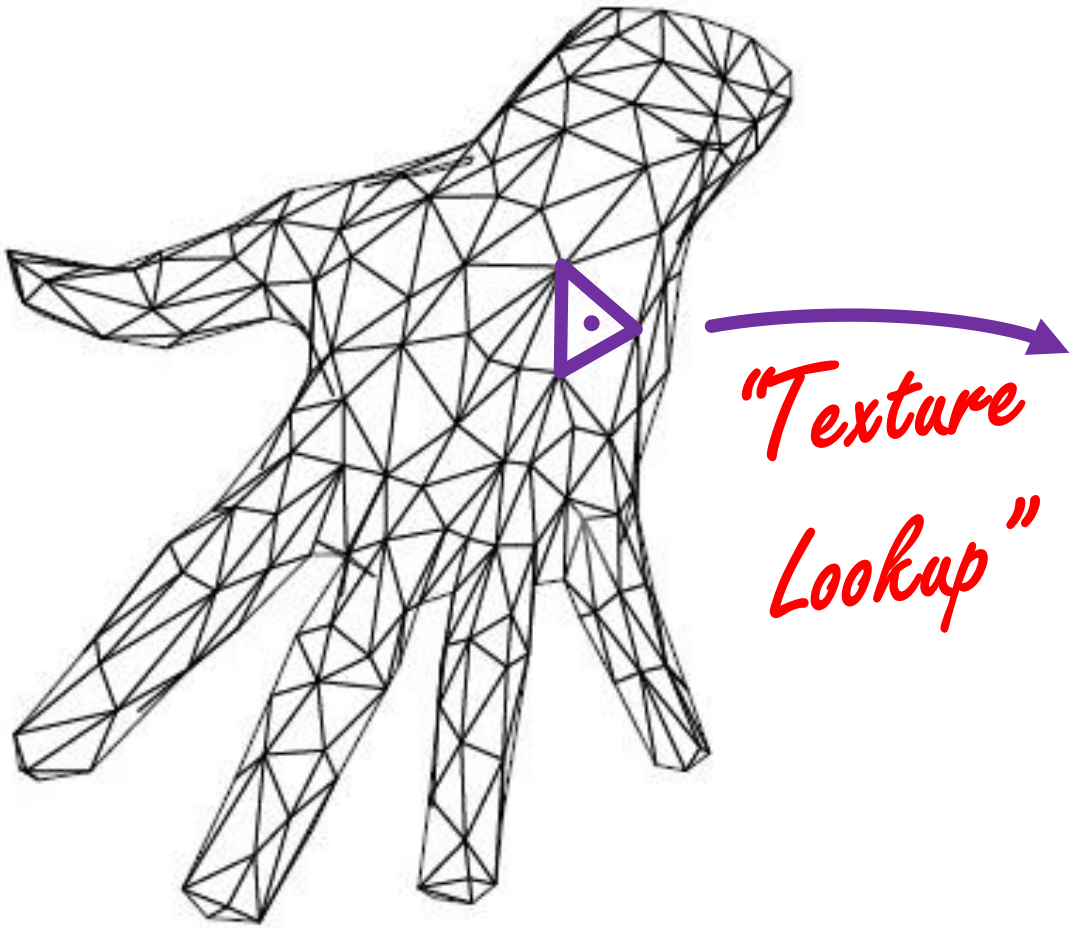


"Texture Lookup"



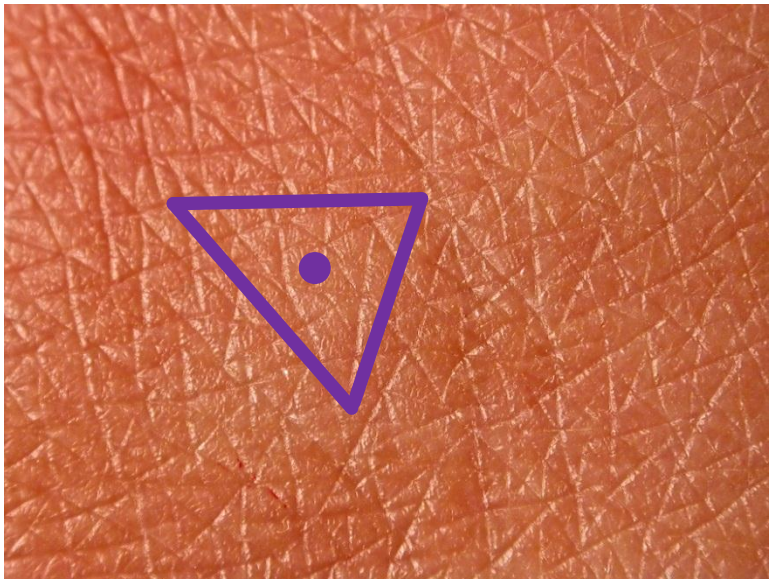
http://raweb.inria.fr/rapportsactivite/RA2006/iparla/uid33.html
http://runfatgirl.files.wordpress.com/2008/05/bigstockphoto_skin_texture_108750.jpg

Rasterizing Textured Triangles



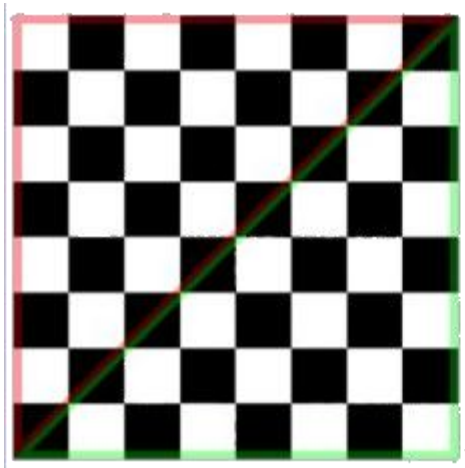
"Texture Lookup"

Texture Memory

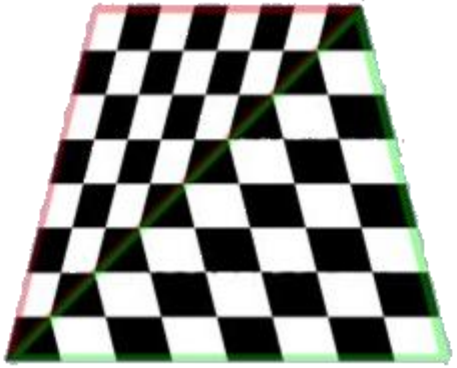


http://raweb.inria.fr/rapportsactivite/RA2006/iparla/uid33.html
http://runfatgirl.files.wordpress.com/2008/05/bigstockphoto_skin_texture_108750.jpg

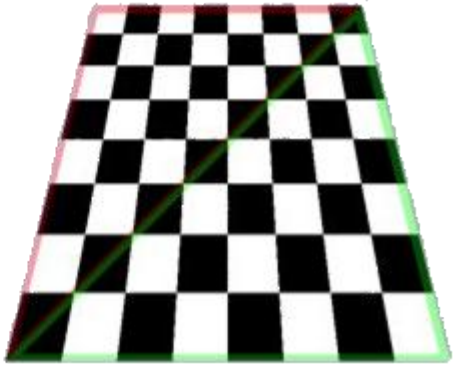
Perspective-Correct Texturing



Flat



Affine

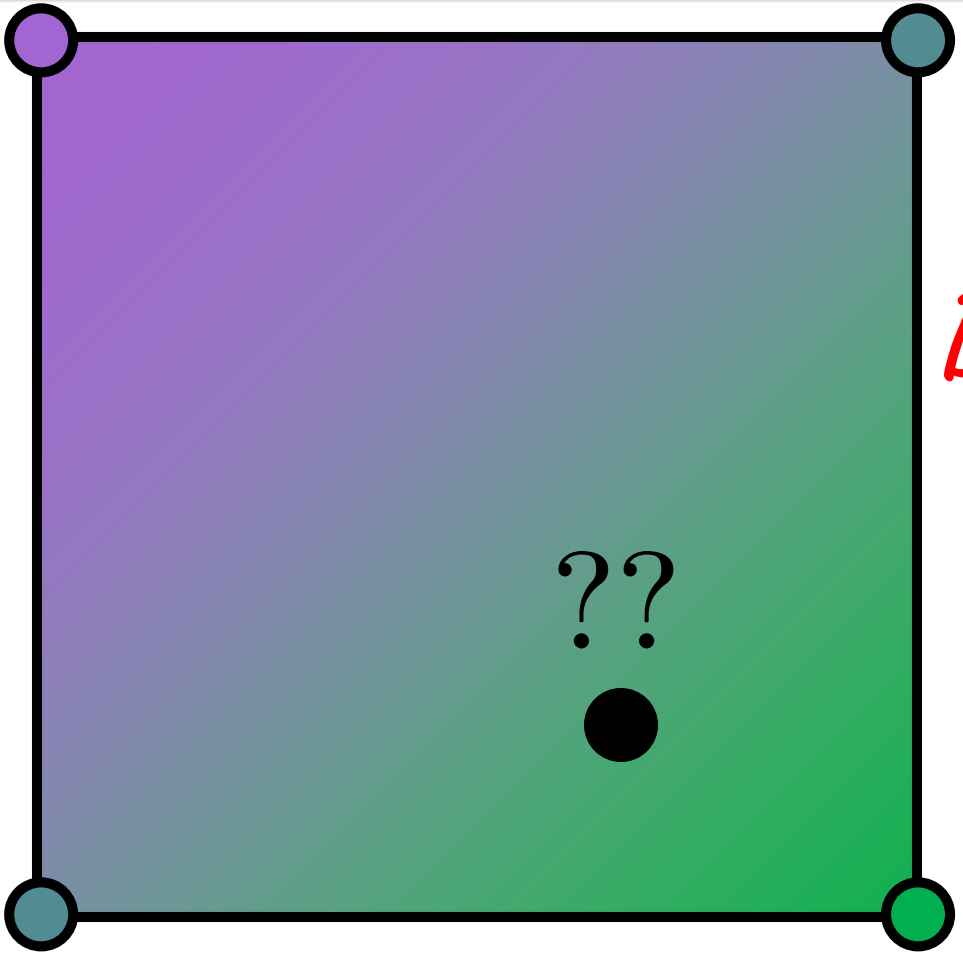


Correct

http://upload.wikimedia.org/wikipedia/commons/5/57/Perspective_correct_texture_mapping.jpg

Can't use image plane barycentric coordinates!

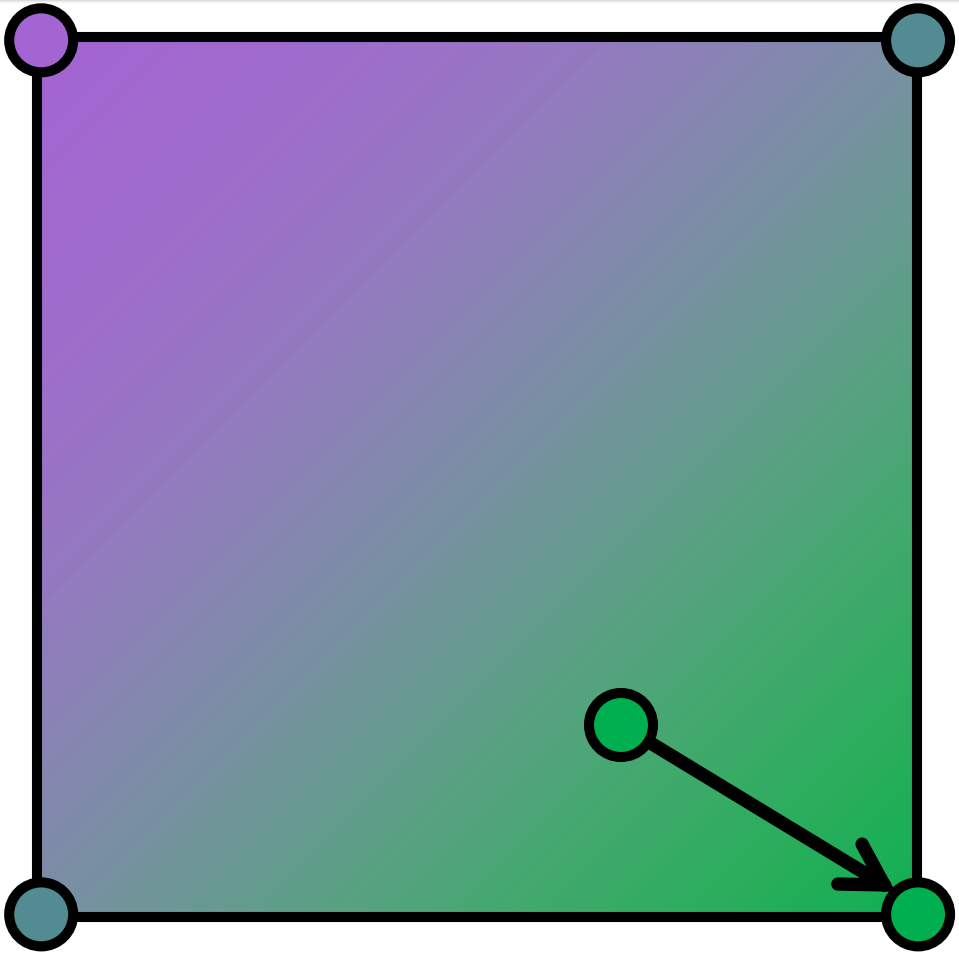
Texture Interpolation



Between four pixels!

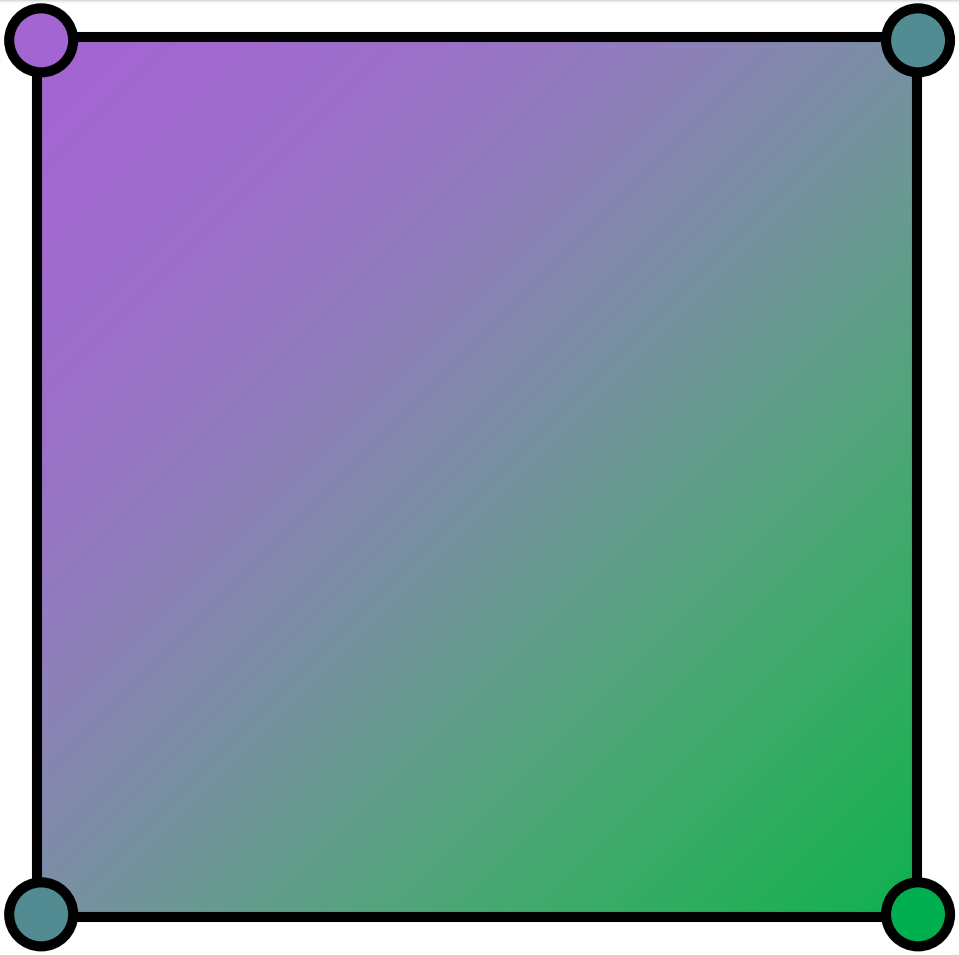
Unlikely to land at pixel center

Texture Interpolation



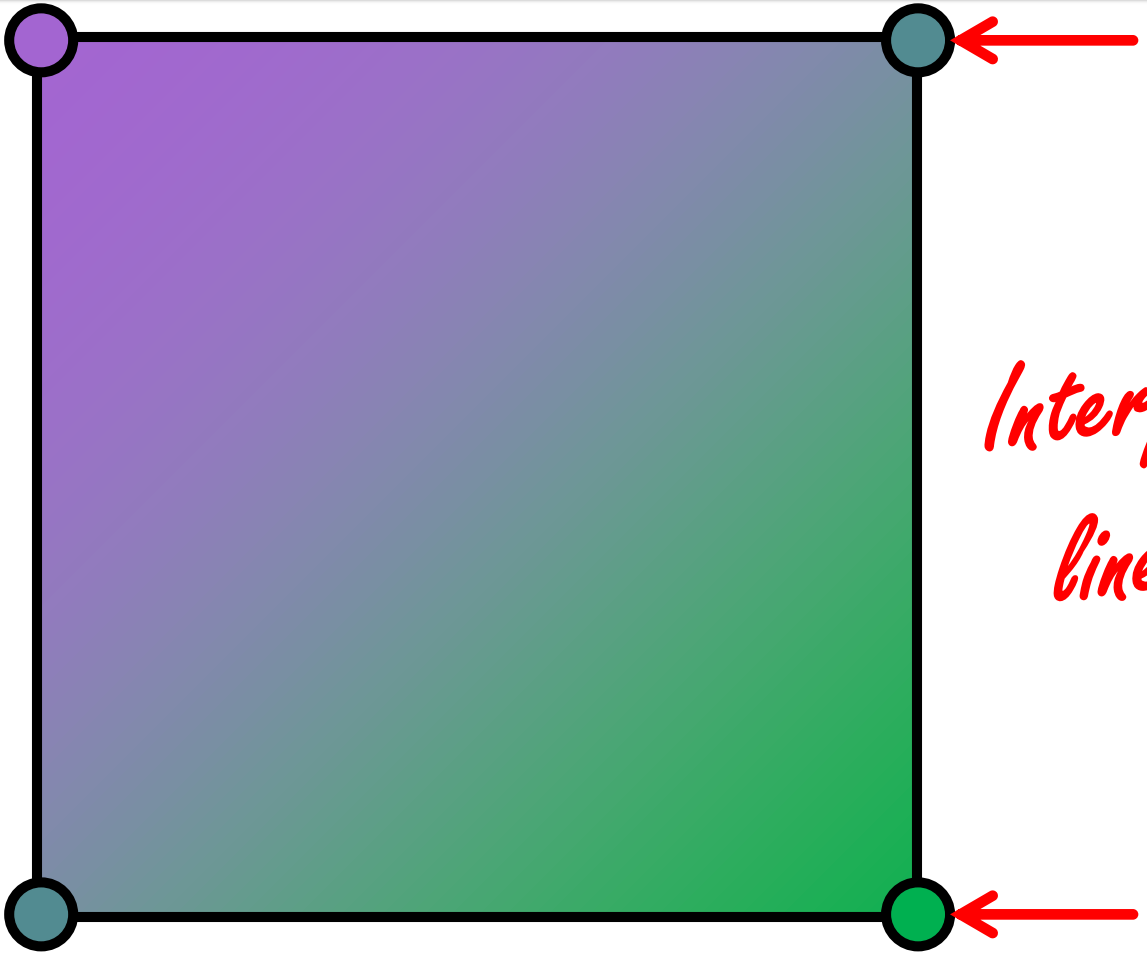
Most efficient: Rounding

Texture Interpolation



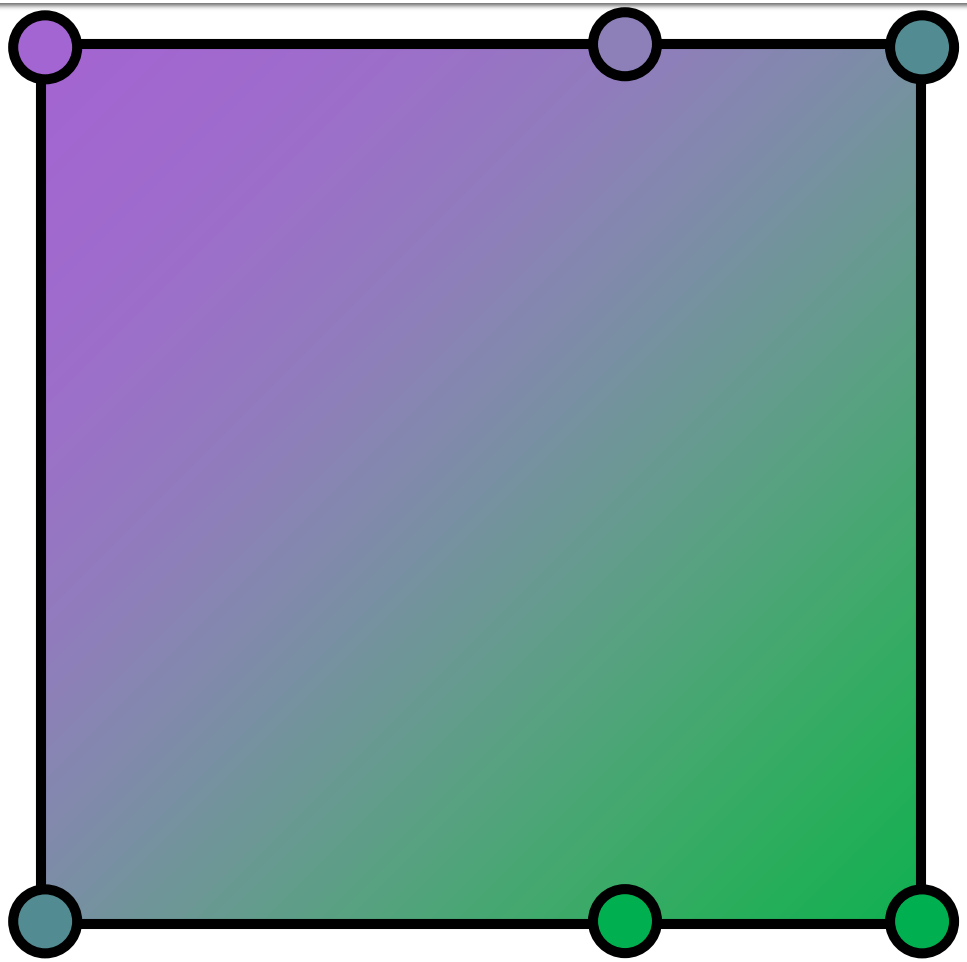
Smoother: Bilinear Interpolation

Texture Interpolation



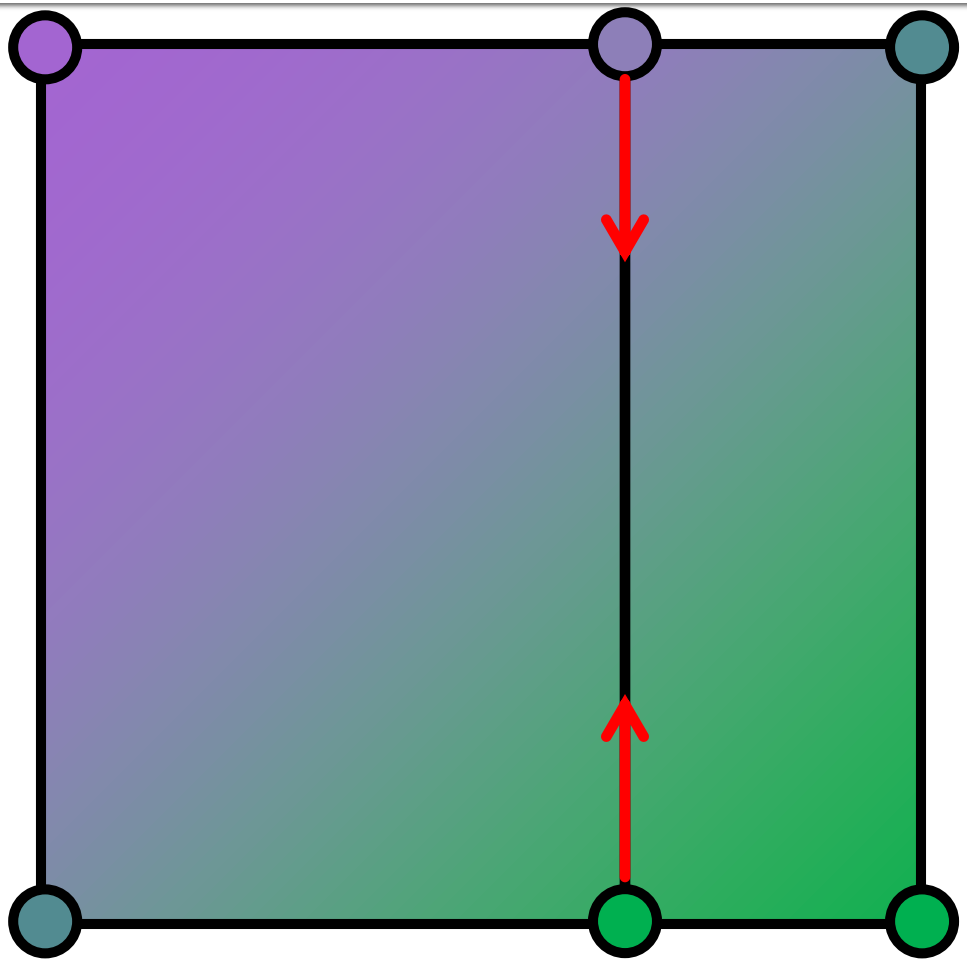
Smoother: Bilinear Interpolation

Texture Interpolation



Smoother: Bilinear Interpolation

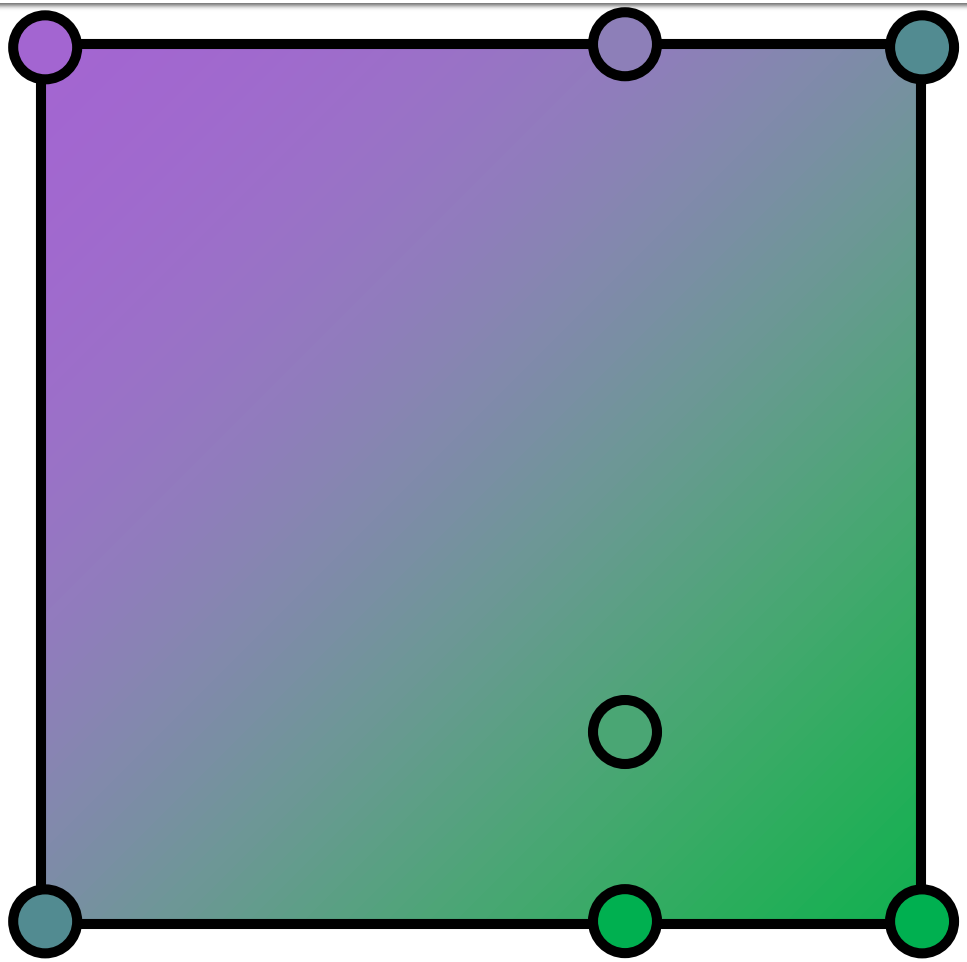
Texture Interpolation



Interpolate again

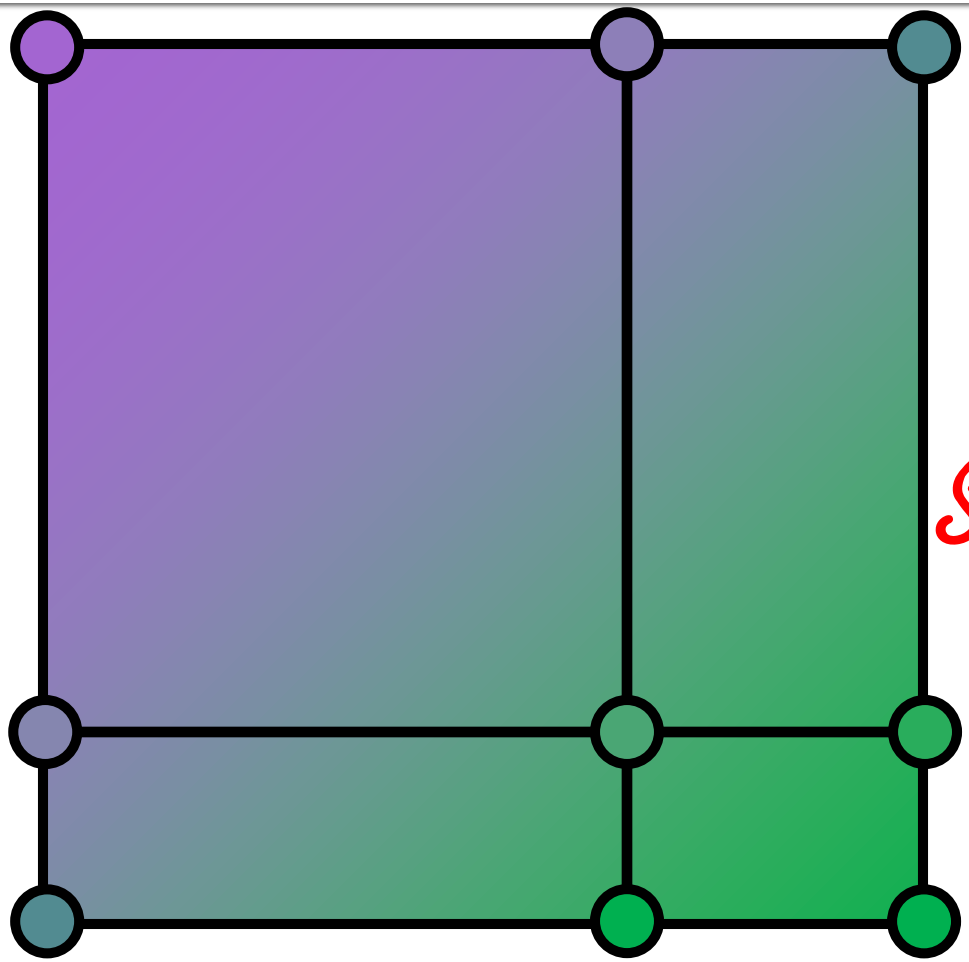
Smoother: Bilinear Interpolation

Texture Interpolation



Smoother: Bilinear Interpolation

Texture Interpolation

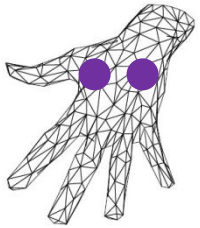


Same formula!

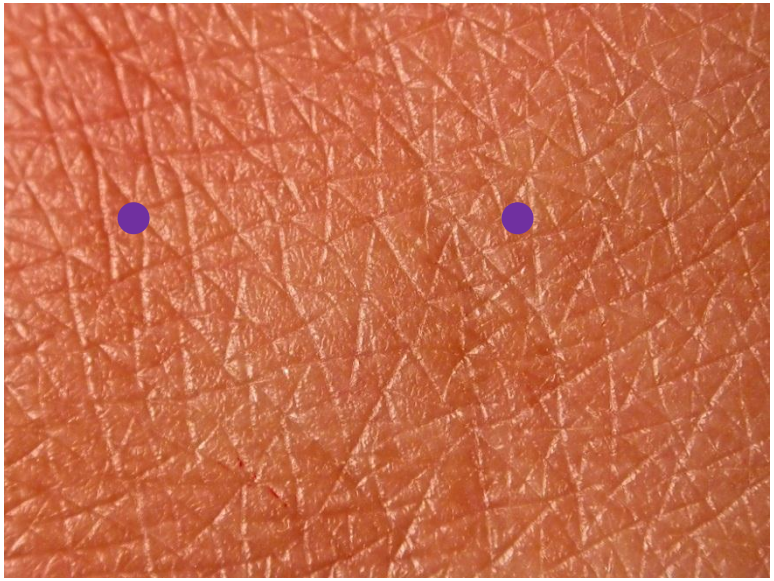
Smoother: Bilinear Interpolation

Texture Can Be *Too* Detailed

Small image

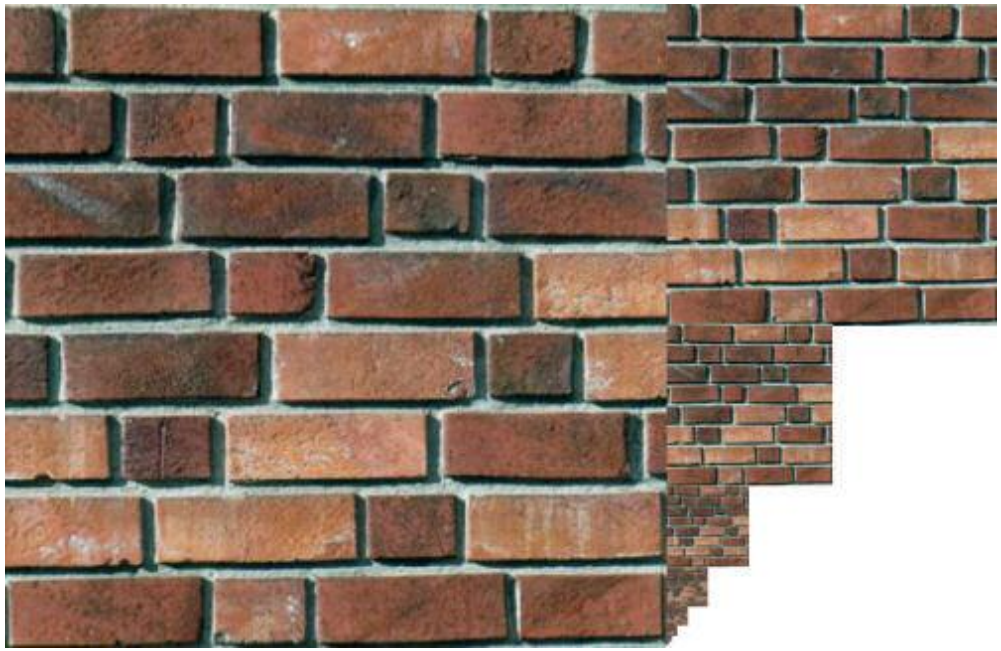


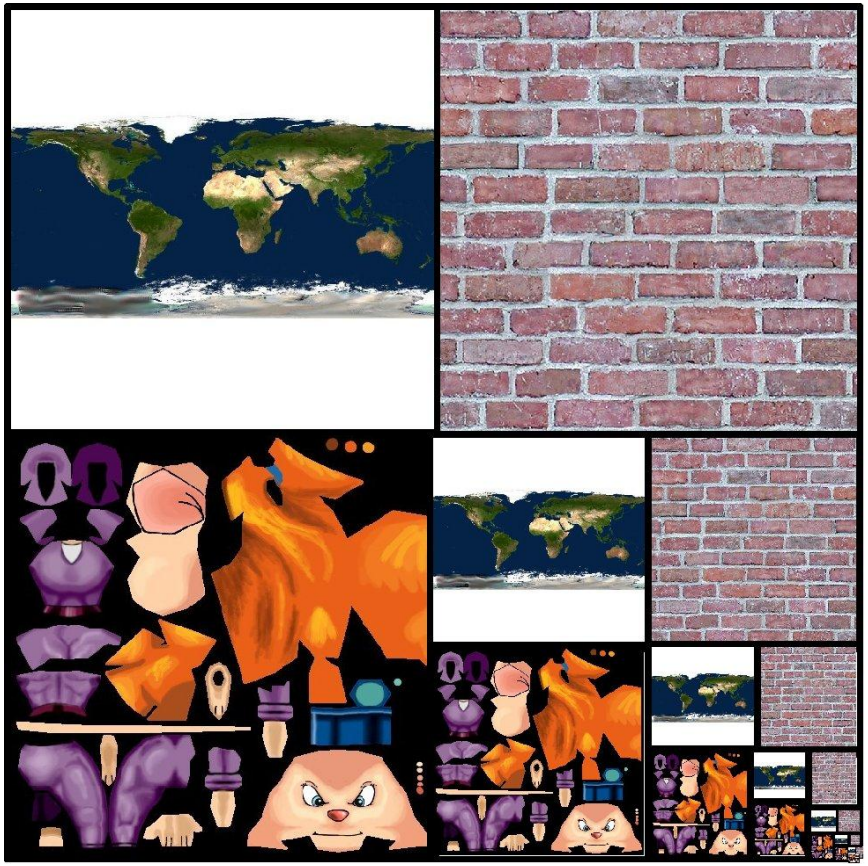
Large texture



Adjacent rendered pixels are far apart in texture

MIP Maps

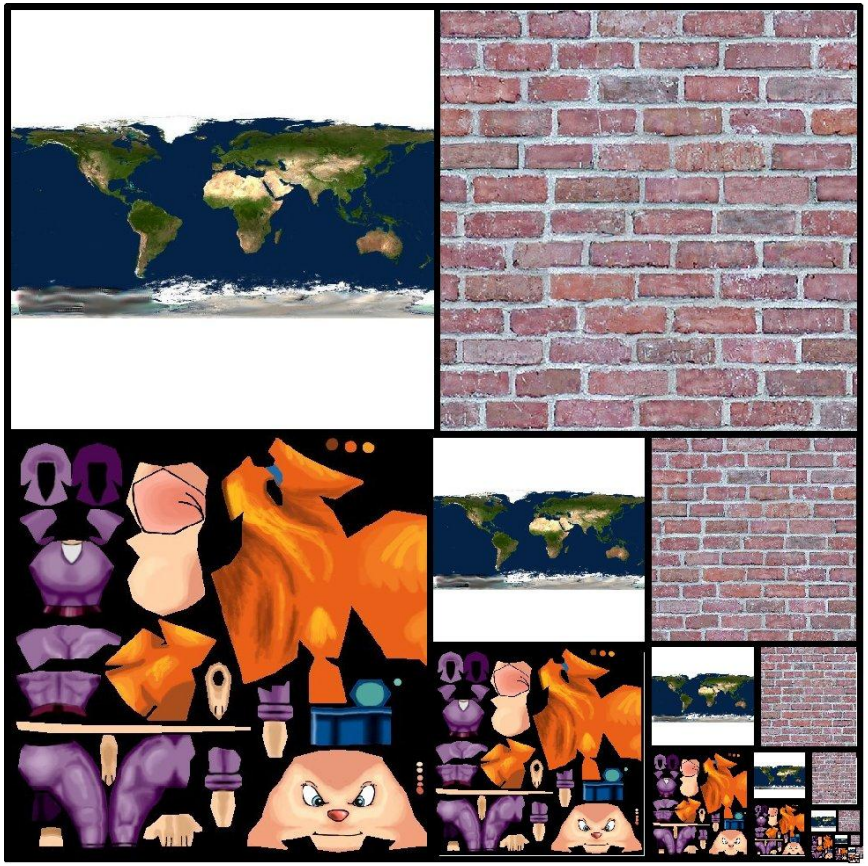




http://glasnost.itcarlow.ie/~powerk/GeneralGraphicsNotes/texturemapping/texturemapping.htm
http://www.ithinkibrokeit.co.uk/articles/graphicso6/page6.php

Precompute small images

MIP Maps



http://glasnost.itcarlow.ie/~powerk/GeneralGraphicsNotes/texturemapping/texturemapping.htm
http://www.ithinkibrokeit.co.uk/articles/graphicso6/page6.php

Precompute small images

MIP Maps

http://glasnost.itcarlow.ie/~powerk/GeneralGraphicsNotes/texturemapping/texturemapping.htm
http://www.ithinkibrokeit.co.uk/articles/graphicso6/page6.php

Precompute small images

Observation

Texture lookups

Barycentric interpolation

MIP mapping

Scan conversion

Lighting

Applying transformations

Depth buffering

We're asking a lot of the GPU!

What's in common?

Apply transformations …	to **millions** of triangles
Look up texture…	for **millions** of pixels
Compute lighting…	for **millions** of vertices
Pick closest…	of **millions** of fragments

Repetitive operations

What's in common?

Apply transformations … to **millions** of triangles

Look up texture… for **millions** of pixels

Compute lighting… for **millions** of vertices

Pick closest… of **millions** of fragments

Single instruction, Multiple data

Repetitive operations

What's in common?

Apply transformations … to **millions** of triangles

Look up texture… for **millions** of pixels

Compute lighting… for **millions** of vertices

Pick closest… of **millions** of fragments

SIMD

Repetitive operations

Modern GPU Throughput

$$\begin{aligned}\text{fragment rate} &= |\text{Image}| \times \text{frames/sec} \times \text{depth complexity} \\ &= (1440 \times 900) \times 60 \times 4 \\ &\approx 32 \text{ gigafragments/sec}\end{aligned}$$

$$\begin{aligned}\text{bandwidth} &= \text{gigafragments/sec} \times \text{bytes/fragment} \\ &= 32 \times 4 \\ &= 96 \text{ GB/sec}\end{aligned}$$

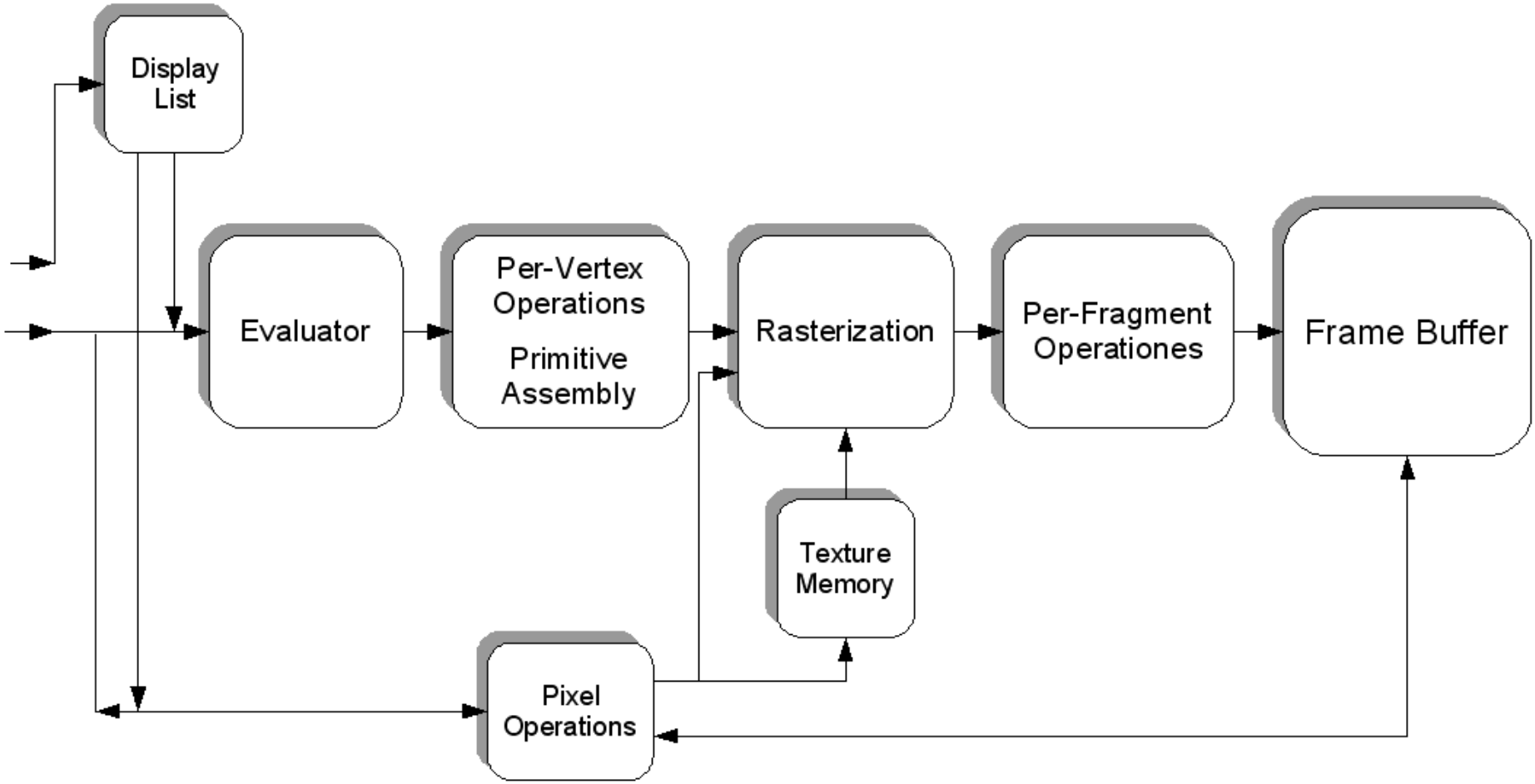
$$\begin{aligned}\text{triangle rate} &= (\text{gigafragments/sec})/(\text{fragments/triangle}) \\ &= 32/16 \\ &= 2 \text{ gigatriangles/sec}\end{aligned}$$

http://graphics.stanford.edu/courses/cs148-11-fall/lectures/drawing.pdf

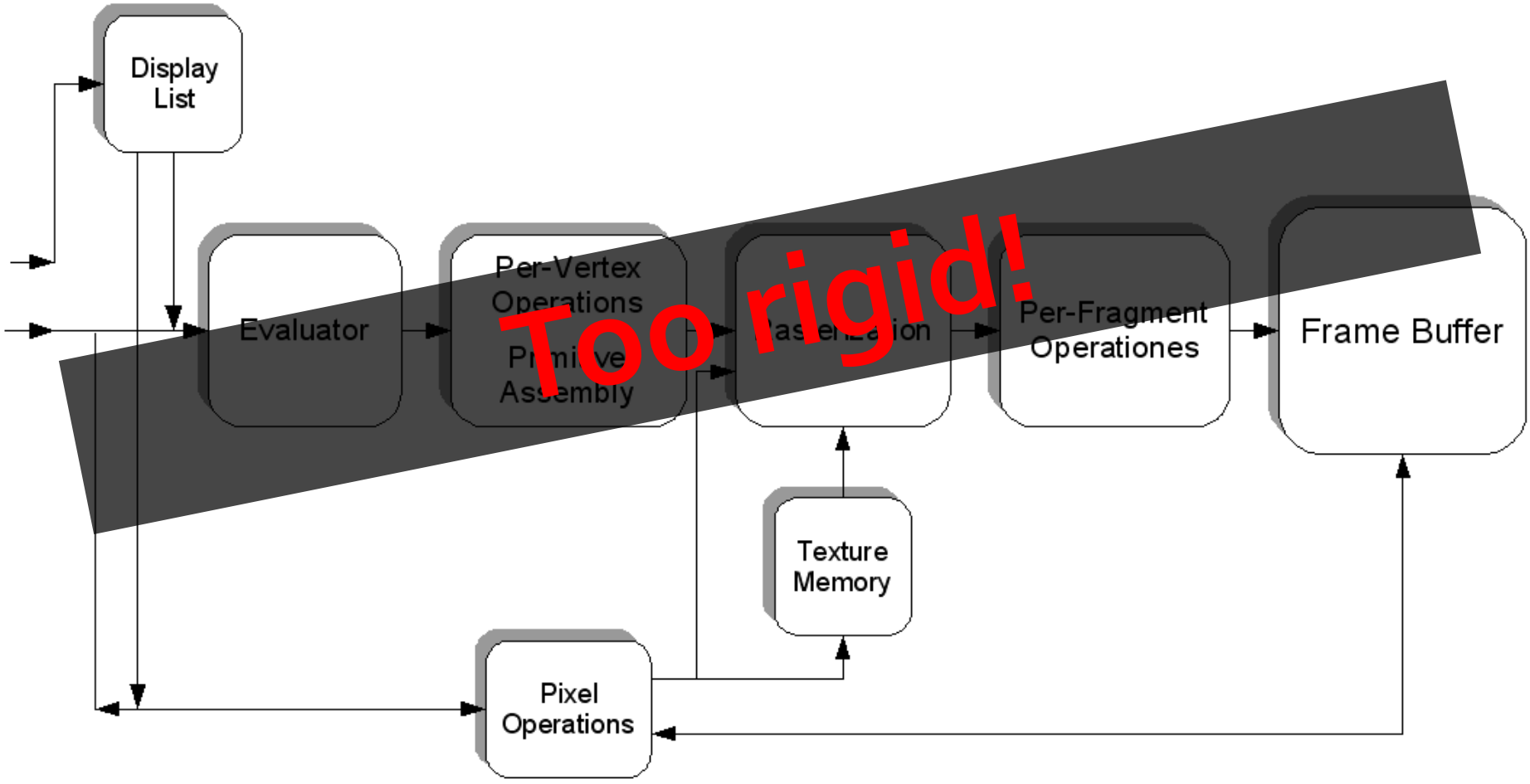
NVidia GPU Performance

Year	Product	Triangles/sec	Fragments/sec
1998	Riva ZX	3m	100m
1999	Riva TNT2	9m	350m
2000	GeForce2 GTS	25m	664m
2001	GeForce3	30m	800m
2002	GeForce Ti 4600	60m	1200m
2003	GeForce FX	167m	2000m
2004	GeForce 6800 Ultra	170m	6800m
2005	GeForce 7800 GTX	940m	10300m
2006	GeForce 7900 GTX	1400	15600m
2007	GeForce 8800 GTX	1800m	36800m
2008	GeForce GTX 280		48160m
2010	GeForce 480		42000m
2011	GeForce GTX 580		49400m

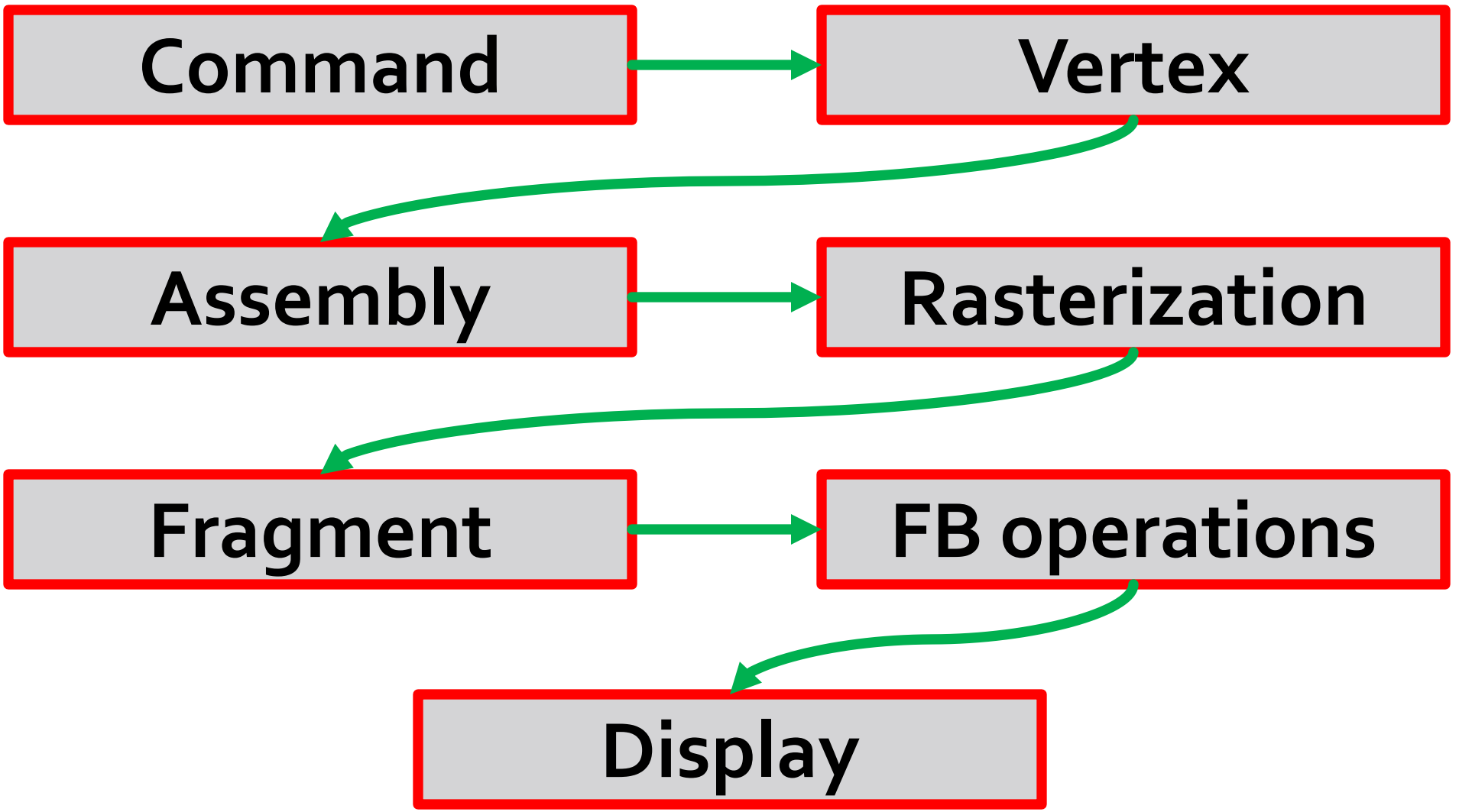
Fixed-Function Pipeline



Fixed-Function Pipeline

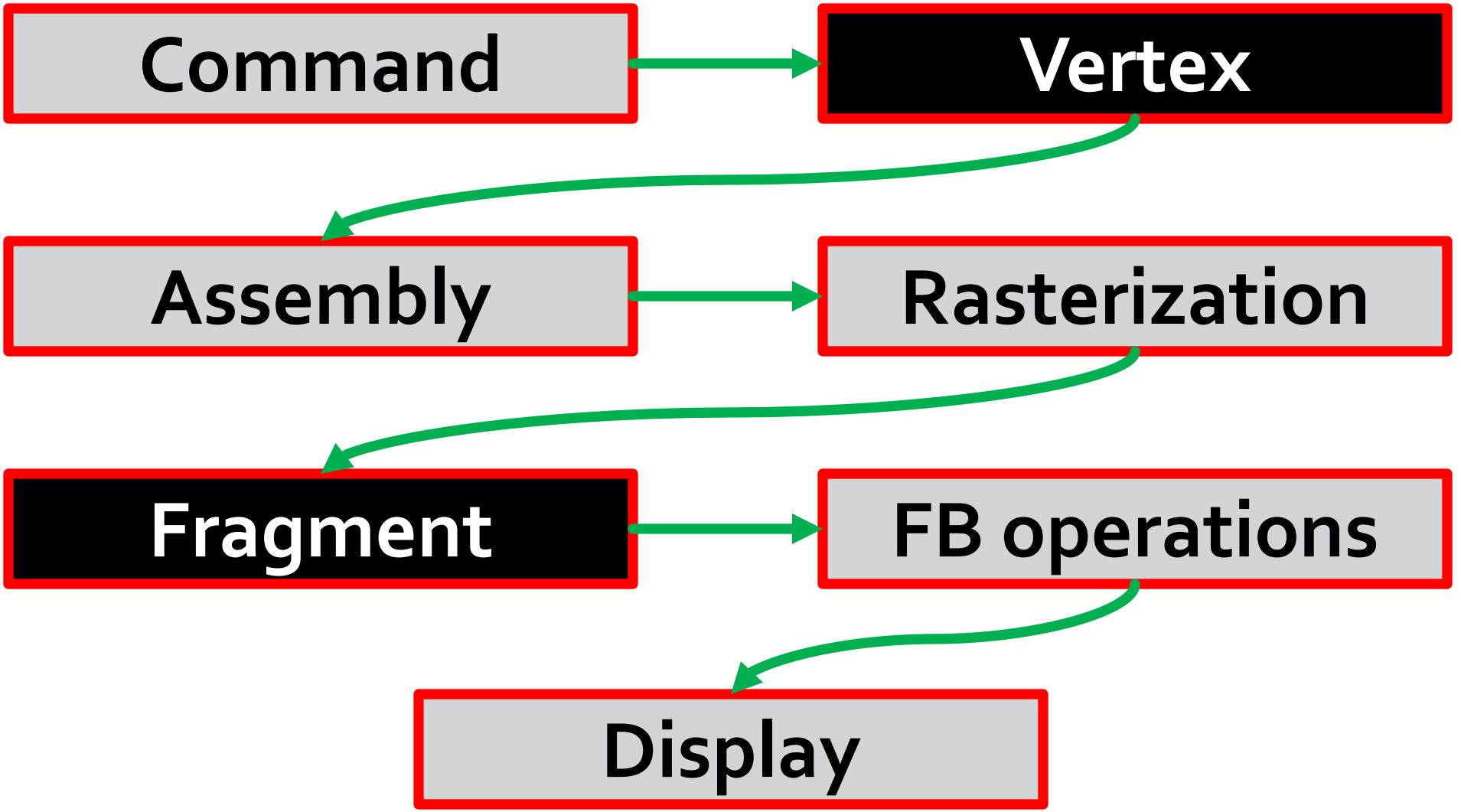


More Generic Pipeline



http://graphics.stanford.edu/courses/cs148-11-fall/lectures/programmable.pdf

Programmable Aspects



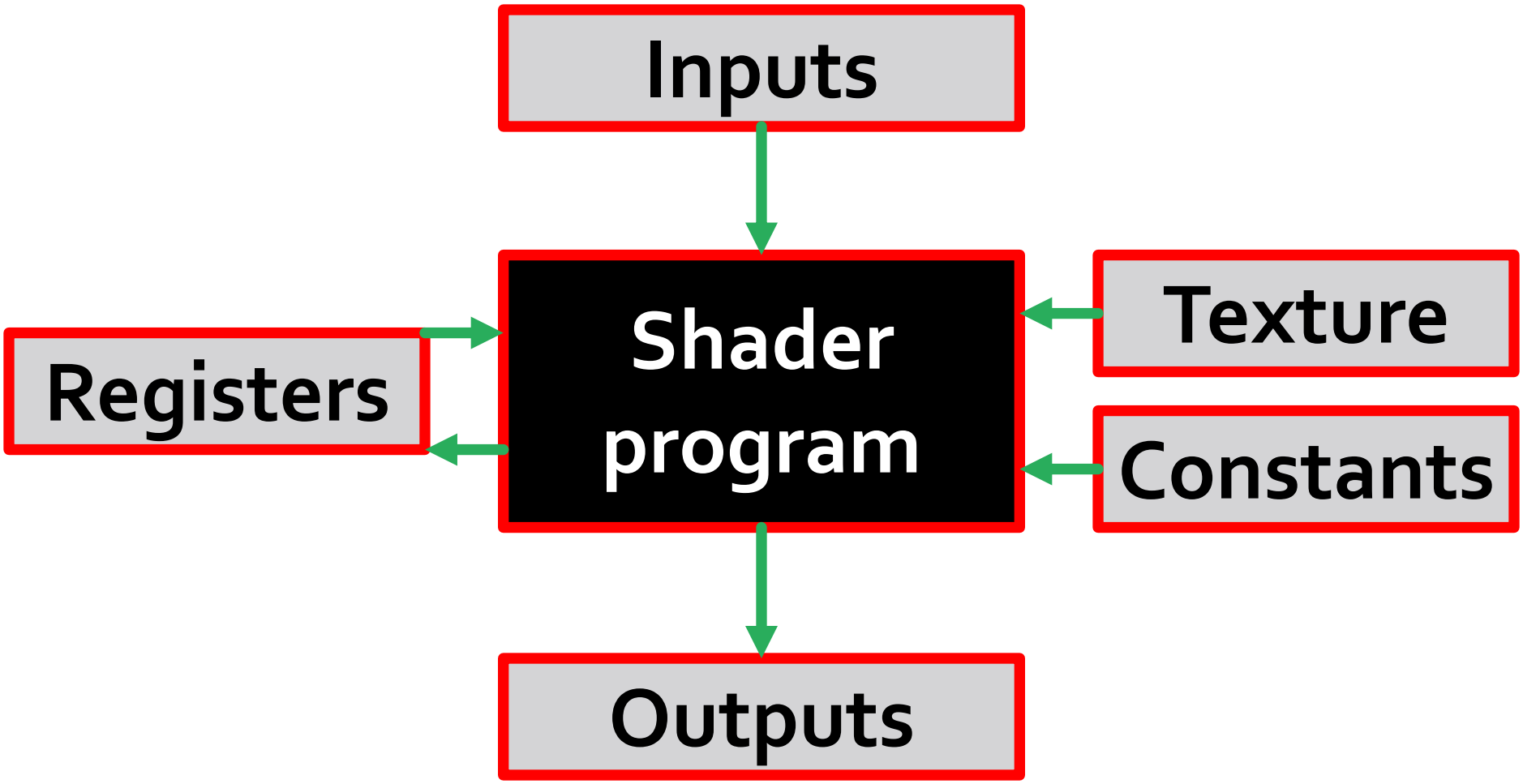
http://graphics.stanford.edu/courses/cs148-11-fall/lectures/programmable.pdf

Introducing "Shaders"

Short program customizing a part of the graphics pipeline.

Misnomer: They do more than shade!

Shader Architecture



http://graphics.stanford.edu/courses/cs148-11-fall/lectures/programmable.pdf

Types of Shaders

Customizes:

- Color and lighting
- Texture coordinate
- Position and transformation
- Normal

Vertex shader *or "Vertex program"*

Types of Shaders

http://www.youtube.com/watch?v=56kt1E1Qioo

Vertex shader *or "Vertex program"*

Types of Shaders

http://www.youtube.com/watch?v=tYesJtyVzFU

Vertex shader *or "Vertex program"*

Types of Shaders

Customizes:

- Visibility and transparency
  - Color
- Texturing and shading
  - Composition

Fragment shader

Types of Shaders

http://www.youtube.com/watch?v=7LvjWGolzQk

Fragment shader

Types of Shaders

http://www.youtube.com/watch?v=9ETfgTD6L2I

Fragment shader

Shader Considerations

- Single-precision arithmetic
- Branching, loops expensive
- No access to neighboring fragments/vertices
- Cannot bind output buffer as input

Shader Considerations

- Limited stack/instruction count
- Timeout possibility
- Support, debugging not consistent

Shader Language Options

- **Assembly**
  Issue commands directly to GPU
- **GLSL**
  OpenGL
- **Cg**
  OpenGL/Direct3D
- **HLSL**
  Direct3D

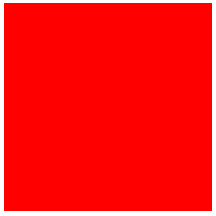
Shader Language Options

- **Assembly**
  Issue commands directly to GPU
- **GLSL**
  OpenGL
- **Cg**
  OpenGL/Direct3D
- **HLSL**
  Direct3D

Simple GLSL Vertex Shader

```
void main() {
    gl_Position = gl_ProjectionMatrix
                * gl_ModelViewMatrix
                * gl_Vertex;
    gl_FrontColor = gl_Color;
    gl_BackColor = gl_Color;
}
```

Input



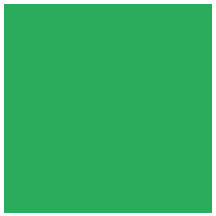
Output

http://graphics.stanford.edu/courses/cs148-10-summer/docs/08_rendering3.pdf

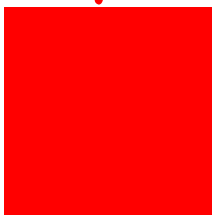
Simple GLSL Vertex Shader

```
void main() {
    gl_Position = gl_ProjectionMatrix
                * gl_ModelViewMatrix
                * gl_Vertex;
    gl_FrontColor = gl_Color;
    gl_BackColor = gl_Color;
}
```

Replace for per-vertex lighting



Input



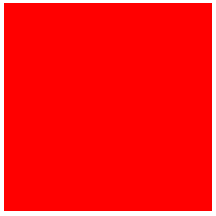
Output

http://graphics.stanford.edu/courses/cs148-10-summer/docs/08_rendering3.pdf

Simple GLSL Fragment Shader

```
void main() {
    gl_FragColor = gl_Color;
}
```

Input



Output

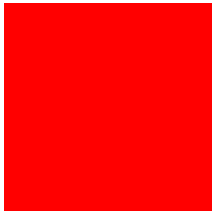
http://graphics.stanford.edu/courses/cs148-10-summer/docs/08_rendering3.pdf

Simple GLSL Fragment Shader

```
void main() {
    gl_FragColor = gl_Color;
}
```

Replace for per-fragment lighting

Input



Output

http://graphics.stanford.edu/courses/cs148-10-summer/docs/08_rendering3.pdf

GLSL Variable Types

- **`Attribute`**: Per-vertex property; input to vertex shader

  `gl_Color`, **`gl_Normal`**, **`gl_MultiTexCoord0`**

- **`Uniform`**: Constant during shader execution

  `gl_ModelViewMatrix`, **`gl_ProjectionMatrix`**

- **`Varying`**: Output by vertex shader and interpolated before running fragment shader

  `gl_FrontColor`, **`height`**

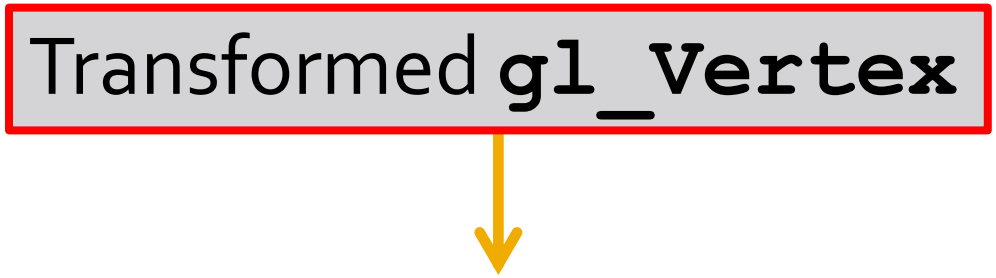
Texture Mapping

```
void main() {
    gl_Position = ftransform();
    gl_TexCoord[0] =
        gl_MultiTexCoord0;
}
```

http://graphics.stanford.edu/courses/cs148-10-summer/docs/08_rendering3.pdf

texture.vert

Texture Mapping




```
void main() {
    gl_Position = ftransform();
    gl_TexCoord[0] =
        gl_MultiTexCoord0;
}
```

http://graphics.stanford.edu/courses/cs148-10-summer/docs/08_rendering3.pdf

texture.vert

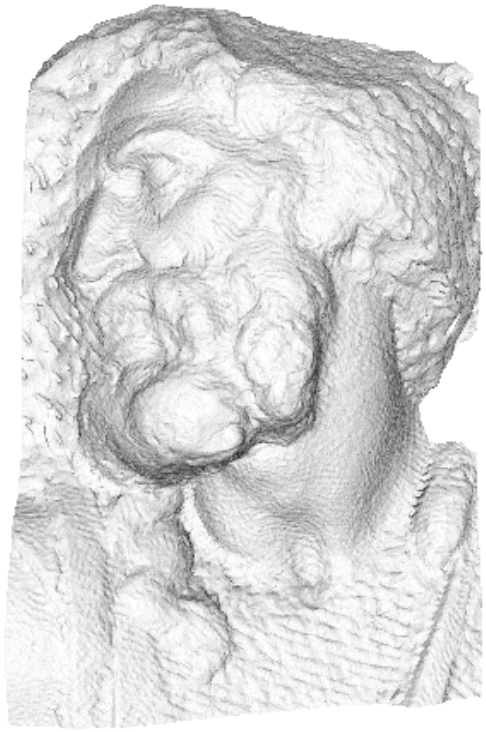
Texture Mapping

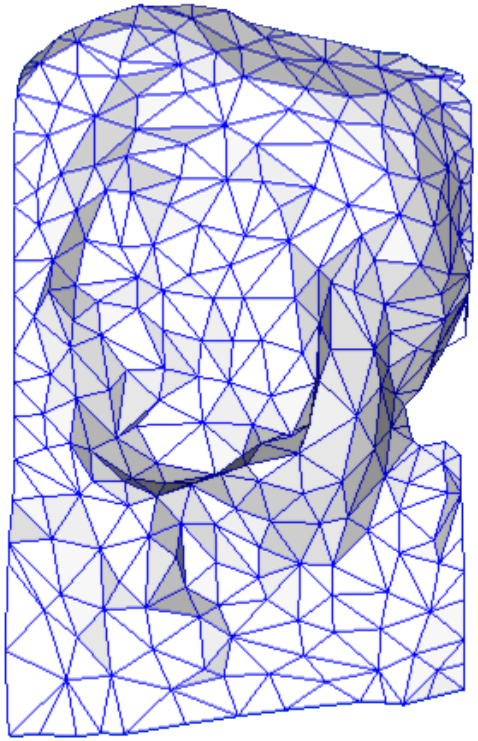
```
uniform sampler2D myTexture;

void main() {
    gl_FragColor =
        texture2D(myTexture,
         gl_TexCoord[0].xy);
}
```

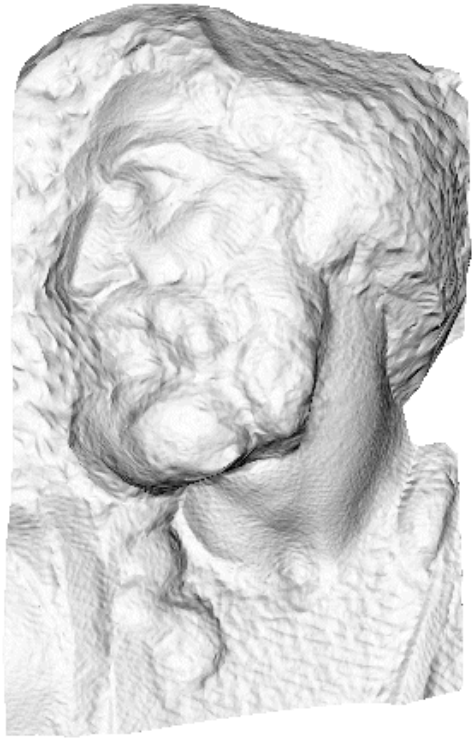
http://graphics.stanford.edu/courses/cs148-10-summer/docs/08_rendering3.pdf

texture.frag

Common Shader Tasks







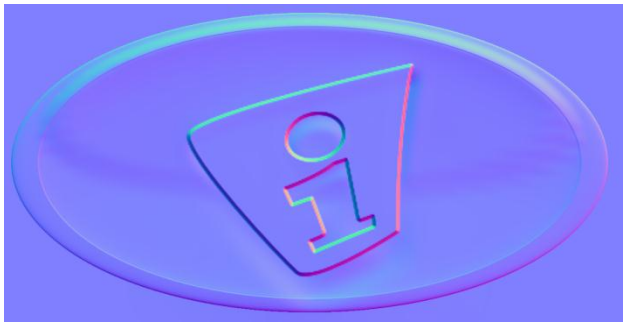
4M triangles

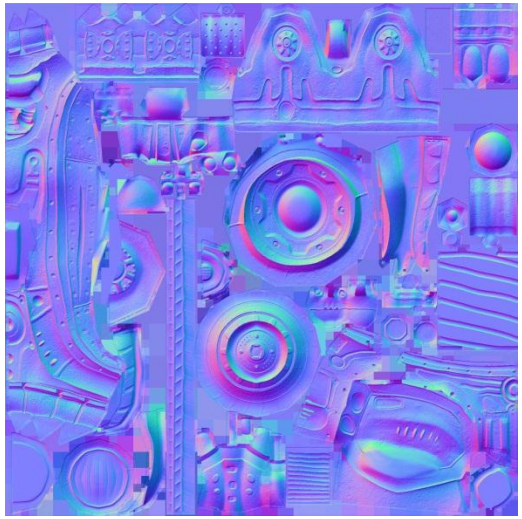
500 triangles

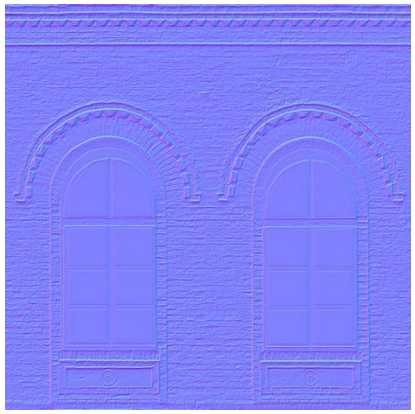
http://upload.wikimedia.org/wikipedia/commons/3/36/Normal_map_example.png

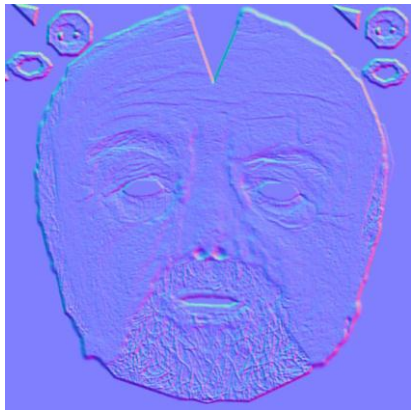
Bump/normal mapping

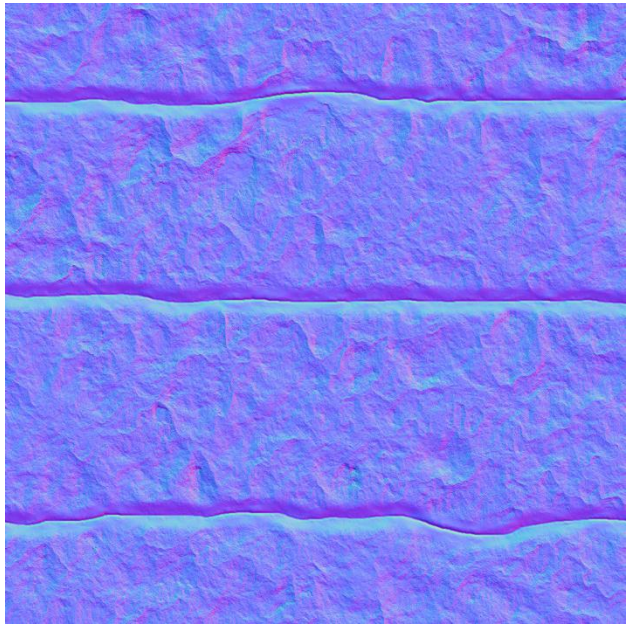
Common Shader Tasks







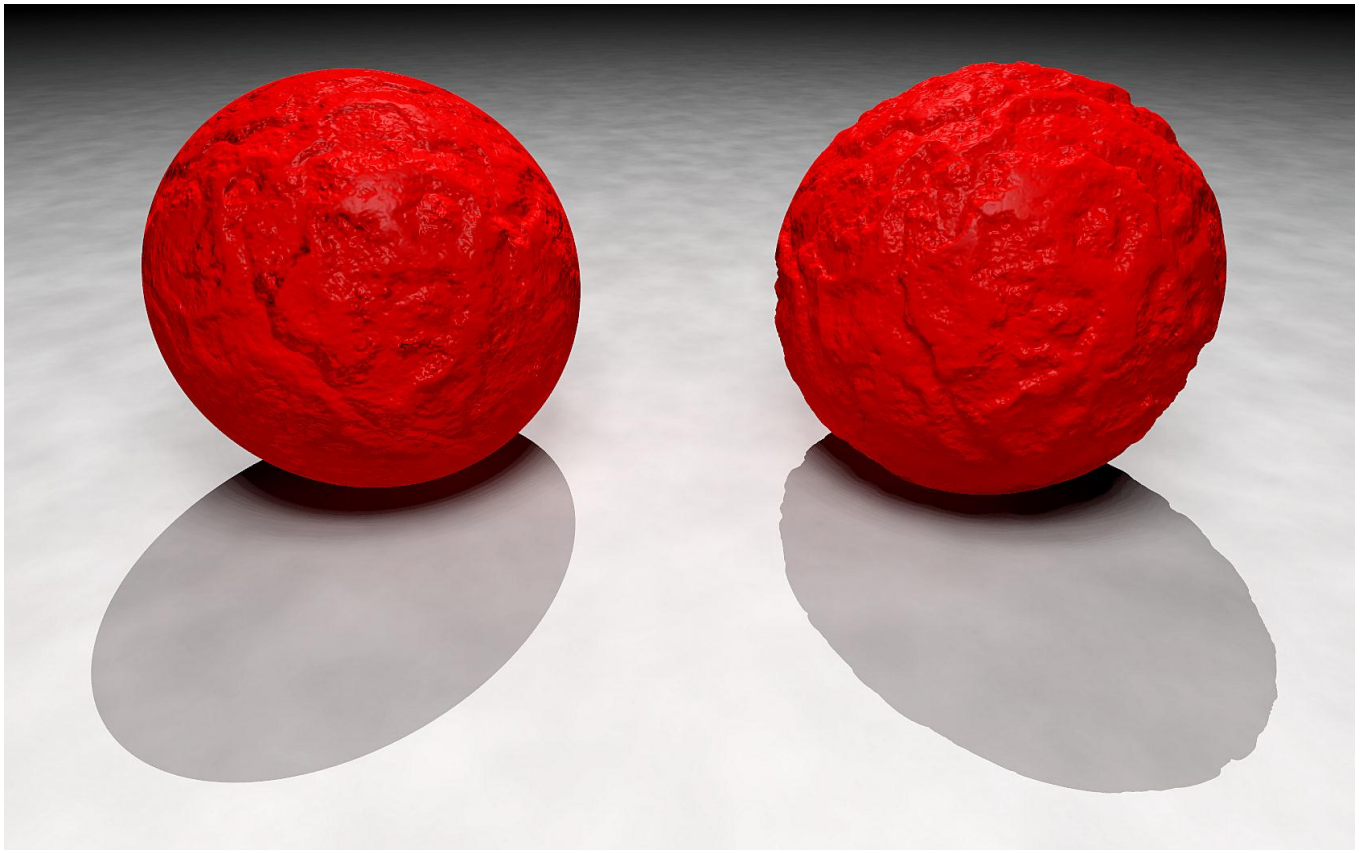




http://www.kxcad.net/lightwave/lightwave_3d_9/normalmap_normal.png
http://charhut.info/files/cs280/CliffNormal.png
http://imageshack.us/photo/my-images/412/gahumanefacenorce9.png/sr=1
http://users.tkk.fi/~mliukka/textures/Brickwall_windows_01_pom_ddn.jpg
http://www.ericspitler.com/images/2d/normalmap.jpg

Bump/normal mapping

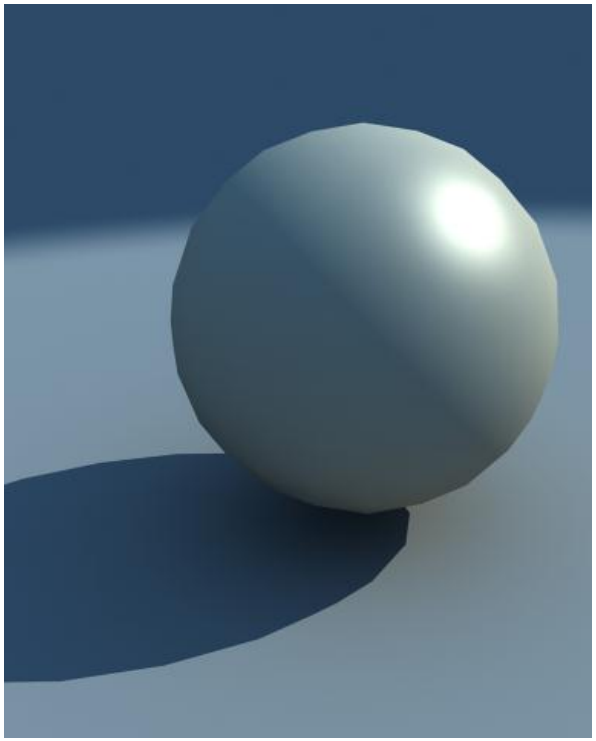
Common Shader Tasks



http://upload.wikimedia.org/wikipedia/commons/4/4e/Bump_map_vs_isosurface2.png

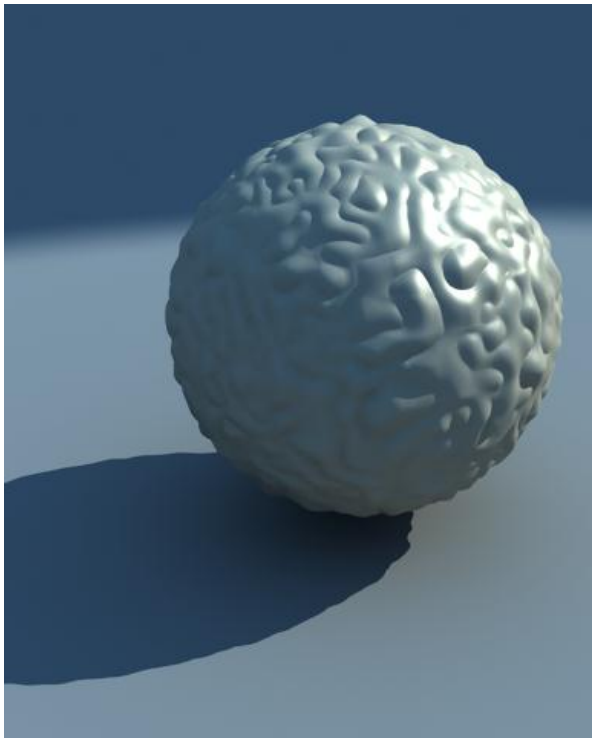
Bump/normal mapping

Common Shader Tasks



Original

Bump



Displacement

http://www.spot3d.com/vray/help/150SP1/tutorials_displacement.htm

Displacement mapping

Common Shader Tasks

Use a smaller set of colors

http://www.cse.unr.edu/~mahsman/courses/cs791a/full.png

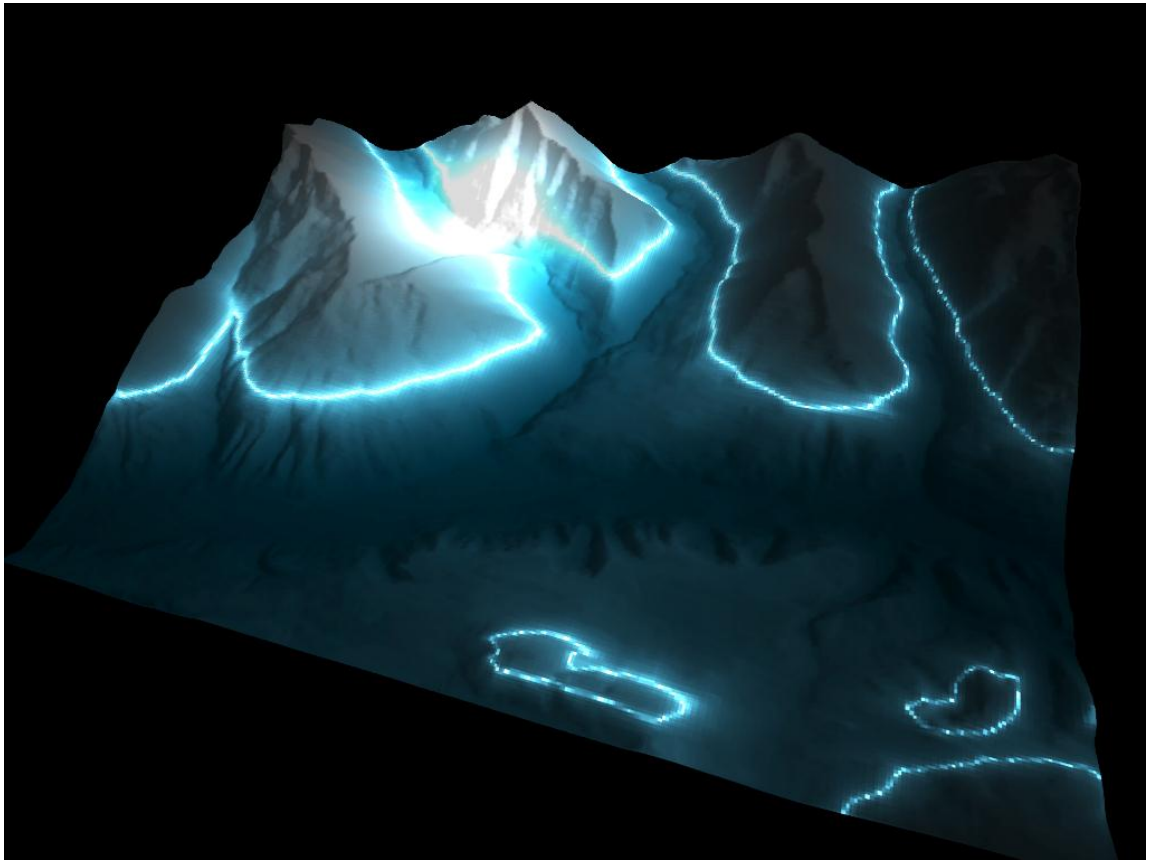
Toon shading

Common Shader Tasks

http://images.digitalmedianet.com/2006/Week_28/youtm5jb/story/10c.jpg

Complex materials/lighting

Common Shader Tasks



http://thomasdiewald.com/blog/wp-content/uploads/2011/08/diewald_projection_shaderized.jpg

Specialized computations

Common Shader Tasks

http://groups.csail.mit.edu/graphics/bilagrid/bilagrid_web.pdf

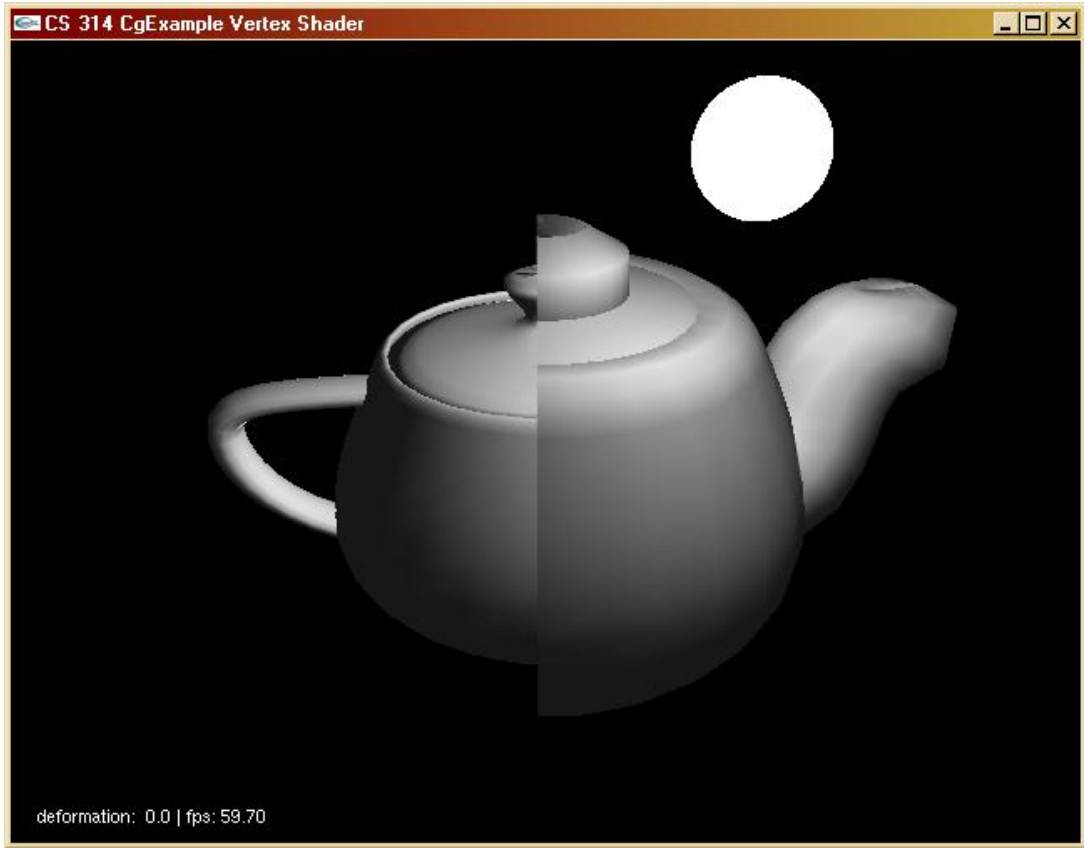
Image processing

Common Shader Tasks

http://www.geeks3d.com/20080812/nvidia-physx-powerpack-download/
http://pcper.com/images/reviews/245/graw_physx_3.jpg

Simulation

Common Shader Tasks



http://web.media.mit.edu/~gordonw/OpenGL/images/teapot_vertexShader.jpg

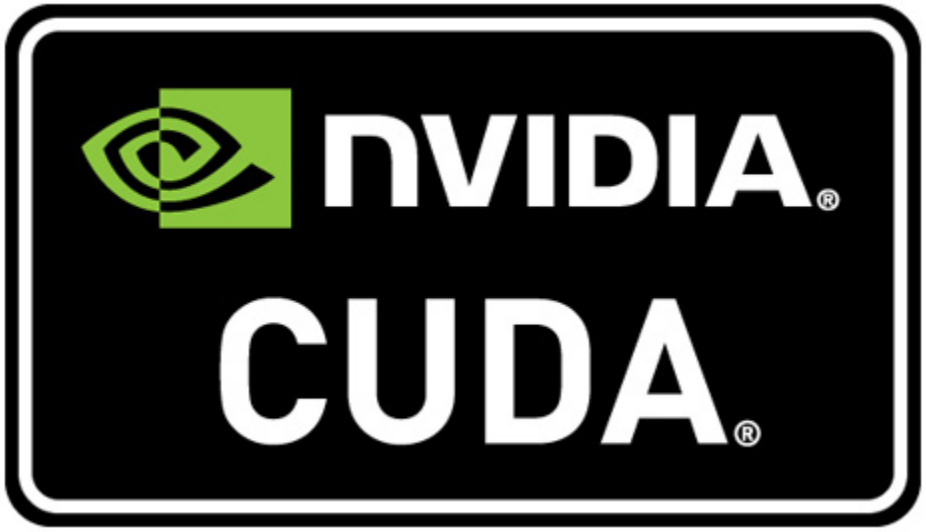
Geometry processing

Common Shader Tasks

http://developer.amd.com/gpu_images/ManagingLargeCrowds.png

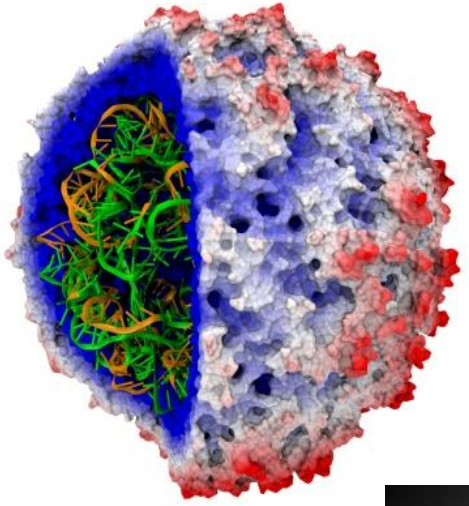
Crowds

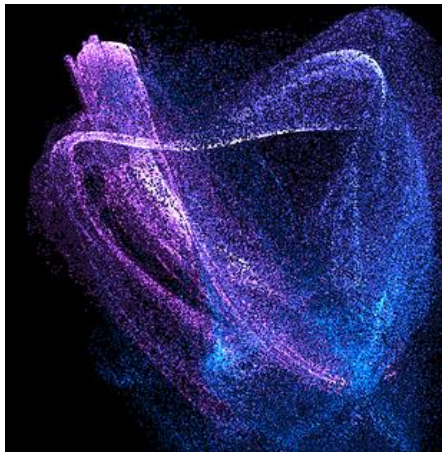
GPGPU

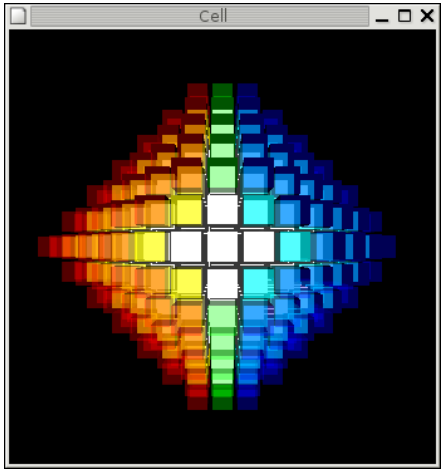


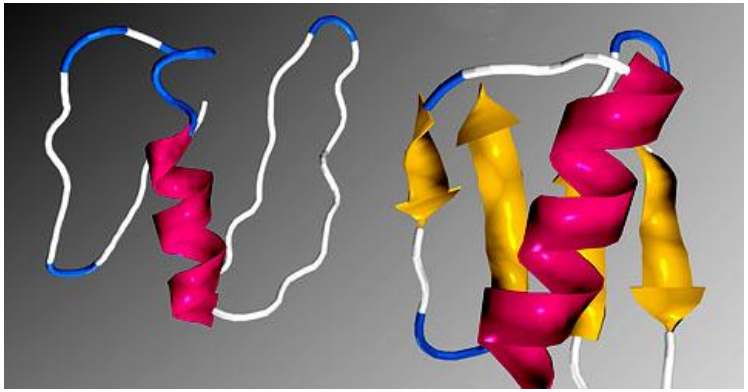
"General purpose" GPU

GPGPU









http://www.techonlineindia.com/article/11-12-08/Hot_technologies_for_2012.aspx
http://eric_rollins.home.mindspring.com/gpgpu/cell3_1.png
http://estoniadonates.files.wordpress.com/2009/02/head.jpg

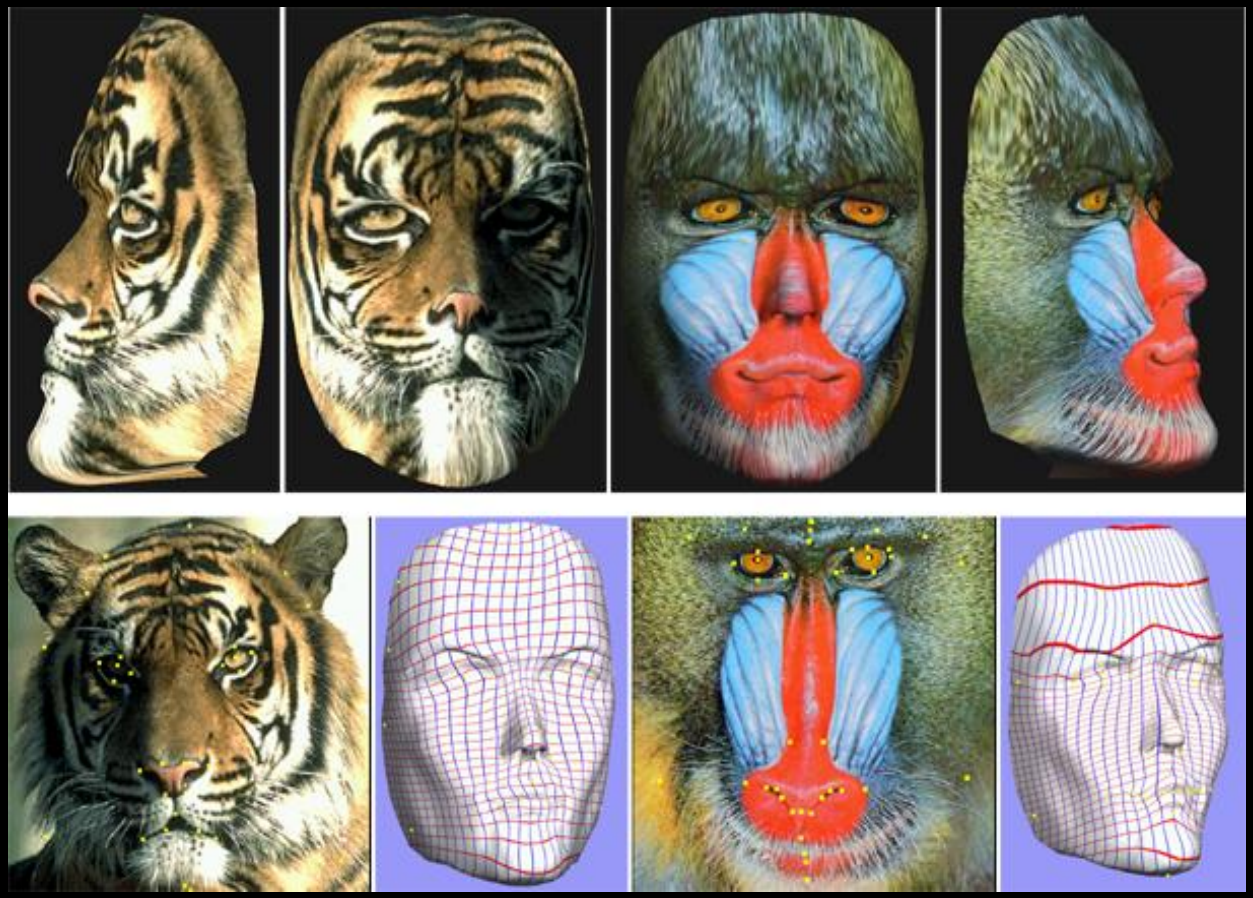
"General purpose" GPU

GPGPU

http://www.digitaltrends.com/wp-content/uploads/2010/10/nvidia-supercomputer-china-tianhe-1a.jpg

"General purpose" GPU

Levy: Constraint Texture Mapping, SIGGRAPH, 2001



Textures and Shaders

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon