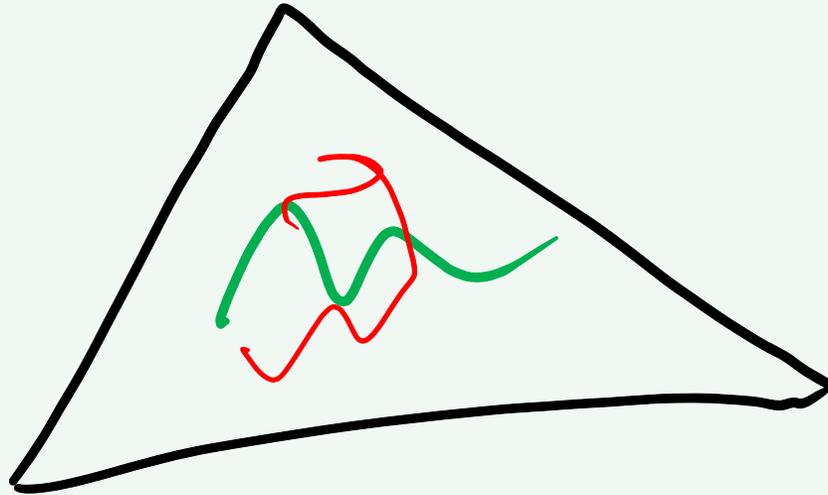


Lecture 18 Part B: Texture Basics

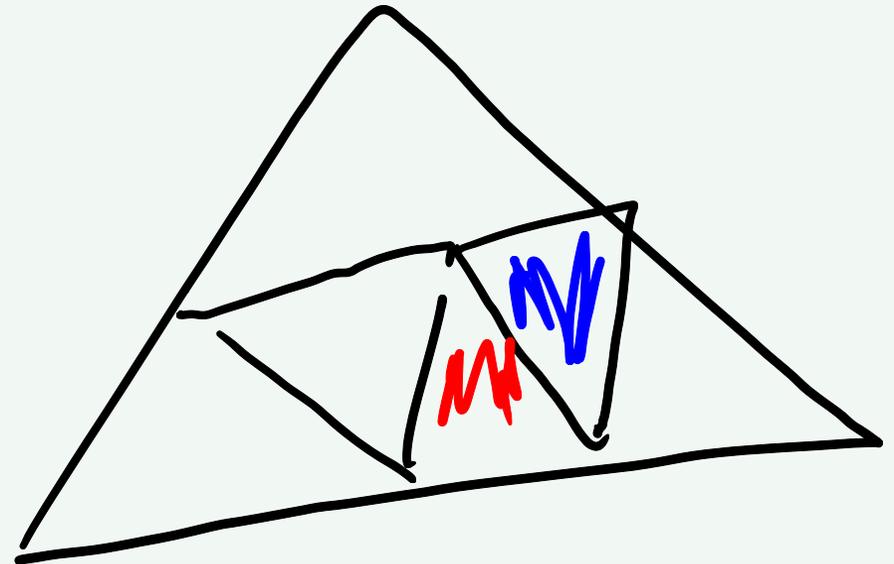
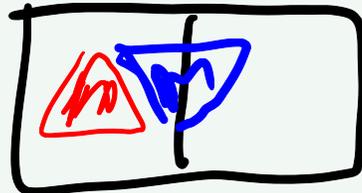
How to get more than 3 colors on a triangle?



Bad Idea #1

Break the triangle into smaller pieces?

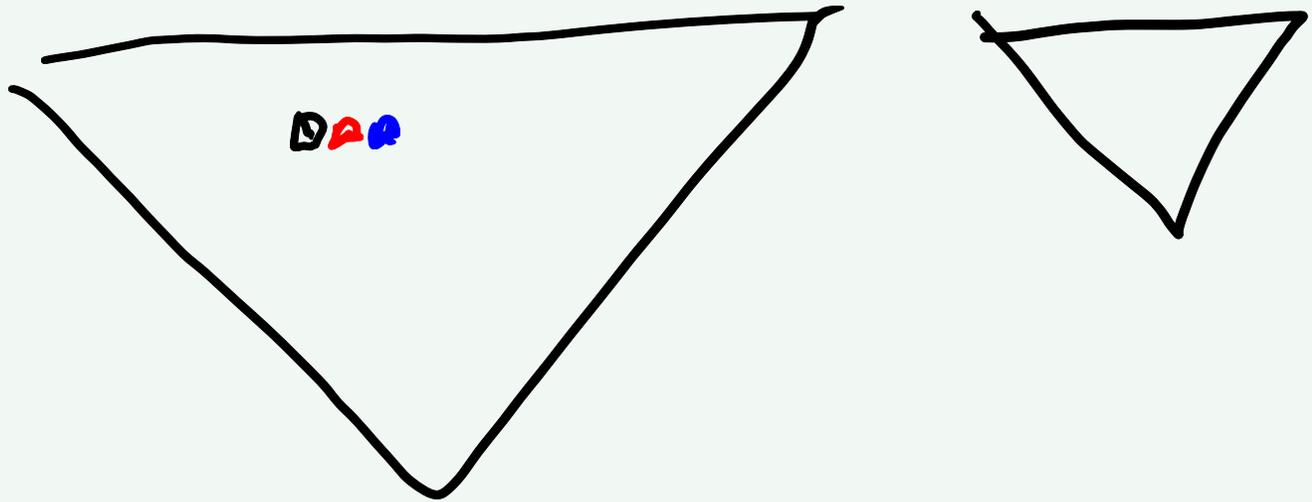
- need to keep track of little triangles
- need to transform/process little triangles
- a lot to draw
- sampling issues (triangles vs. pixels)



Bad Idea #2

Colors per Pixel?

- How do you know the pixels ahead of time?

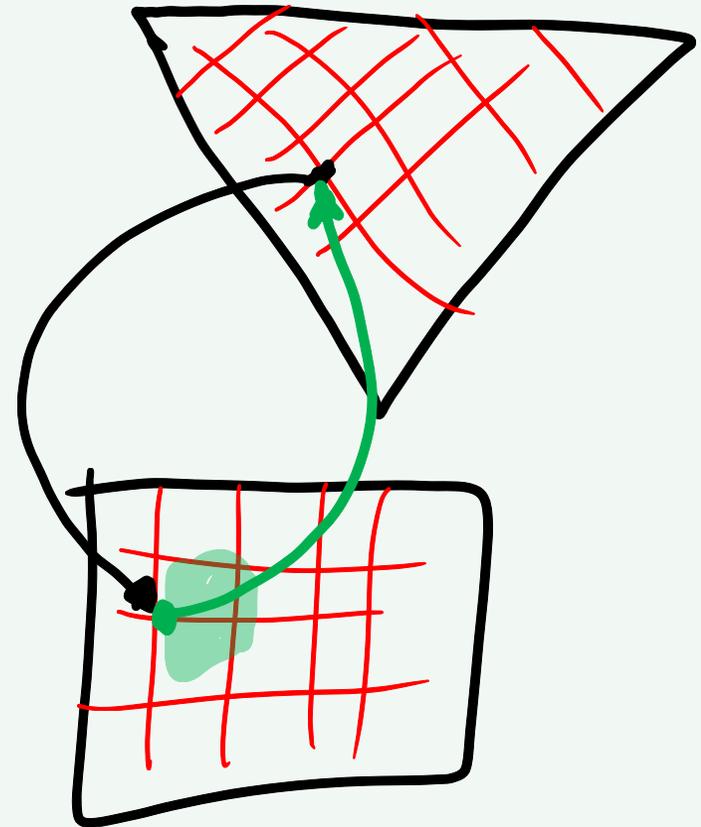


Texture Mapping: Basic Idea

1. Define a coordinate system on the triangle
2. Define a map from coordinate to color

When we draw, each pixel:

1. figures out its coordinate
2. looks up its color (*)



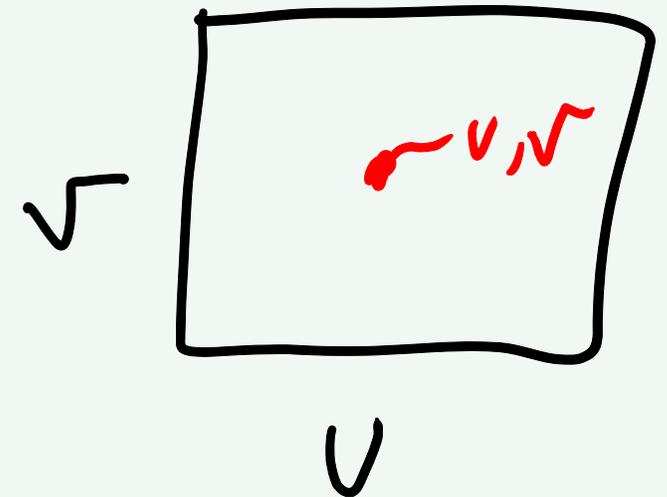
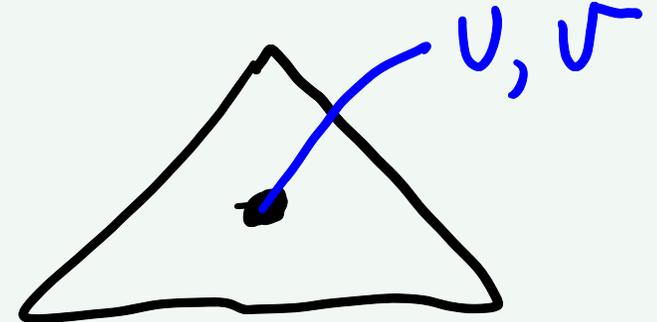
Texture Mapping

Defined by three parts:

1. Some way to assign coordinates
2. Some way to look up the value for that coordinate
3. Something to do with the value when we look it up

Most common/basic form:

1. UV coordinates on triangles (2D coord per point)
2. Images that define colors (lookup color in 2D)
3. Use looked up color as the color for lighting

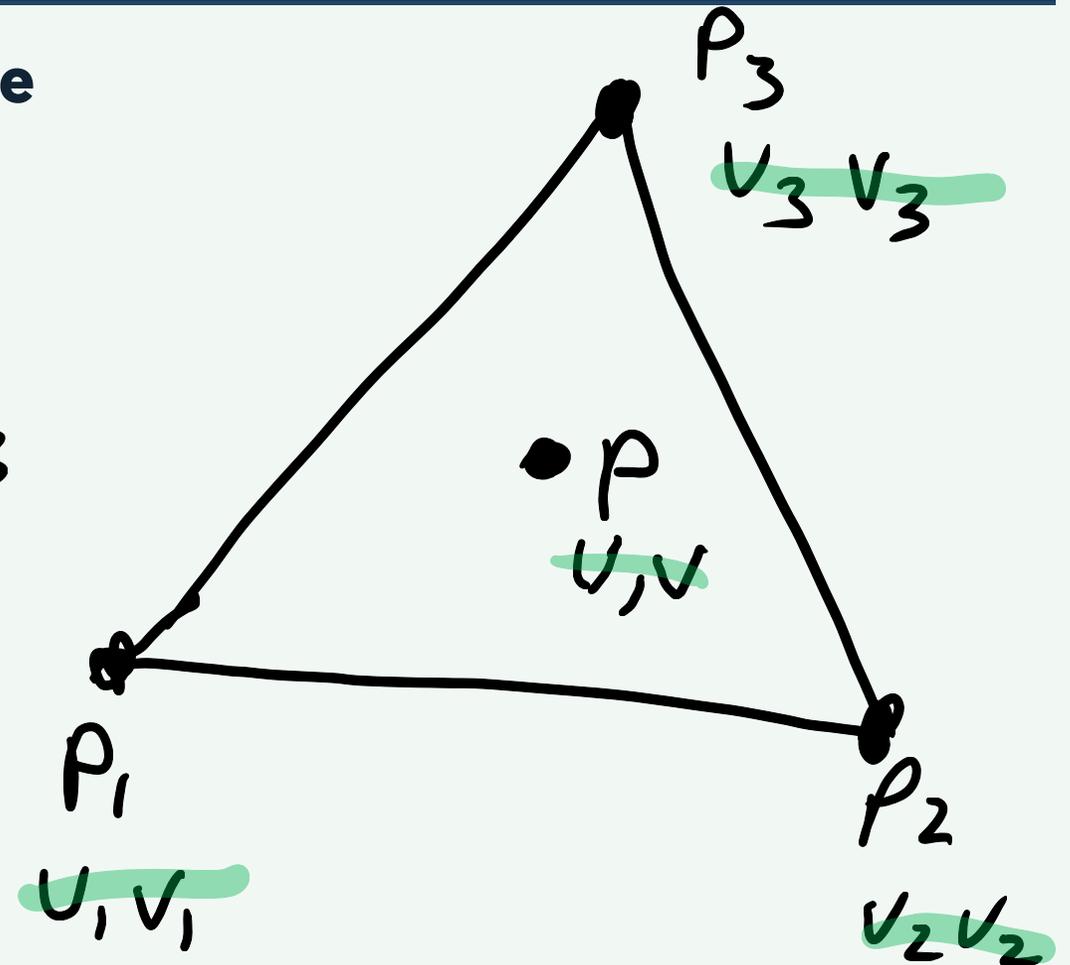


Basic Texture Mapping

1. Define UV coordinates for the triangle
2. Use an image to map UV to color
3. Use the color as the surface color

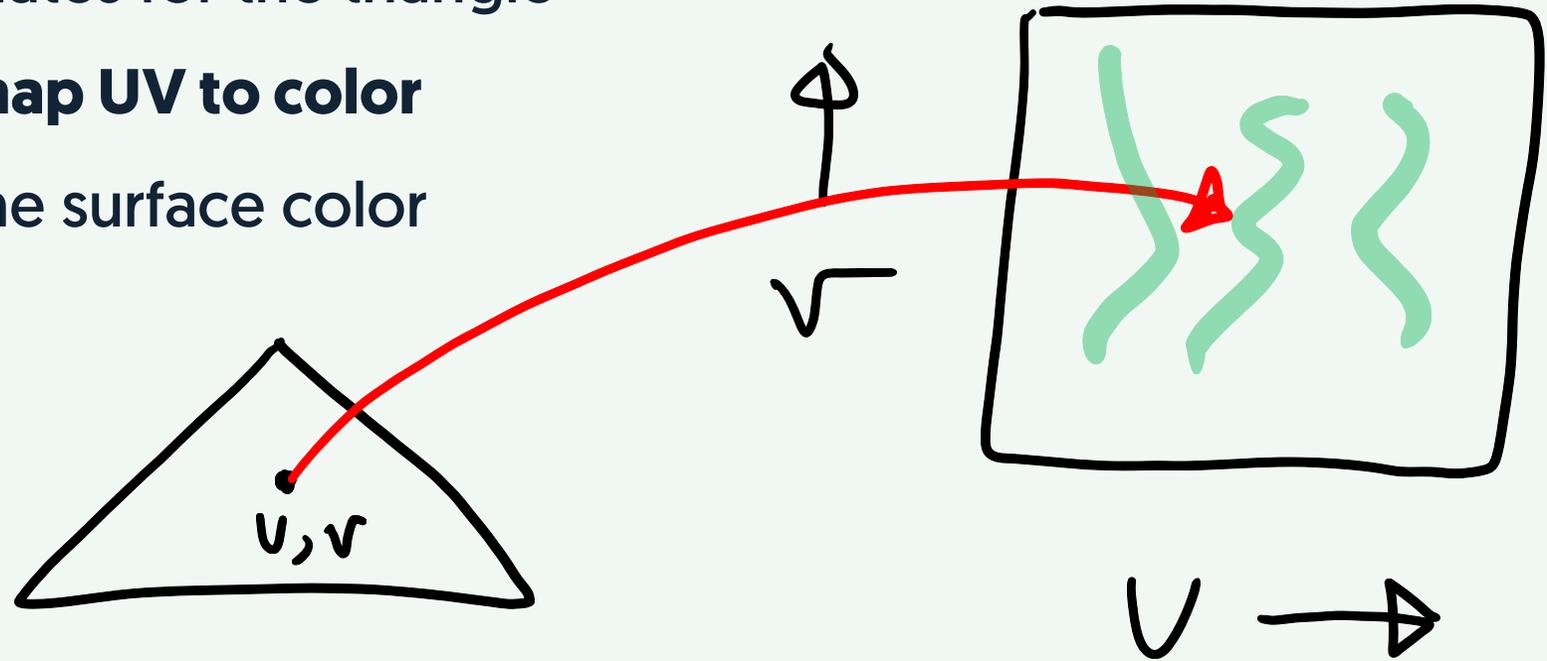
$$P = \alpha P_1 + \beta P_2 + \gamma P_3$$

$$uv = \alpha u + \beta v + \gamma v$$



Basic Texture Mapping

1. Define UV coordinates for the triangle
2. **Use an image to map UV to color**
3. Use the color as the surface color



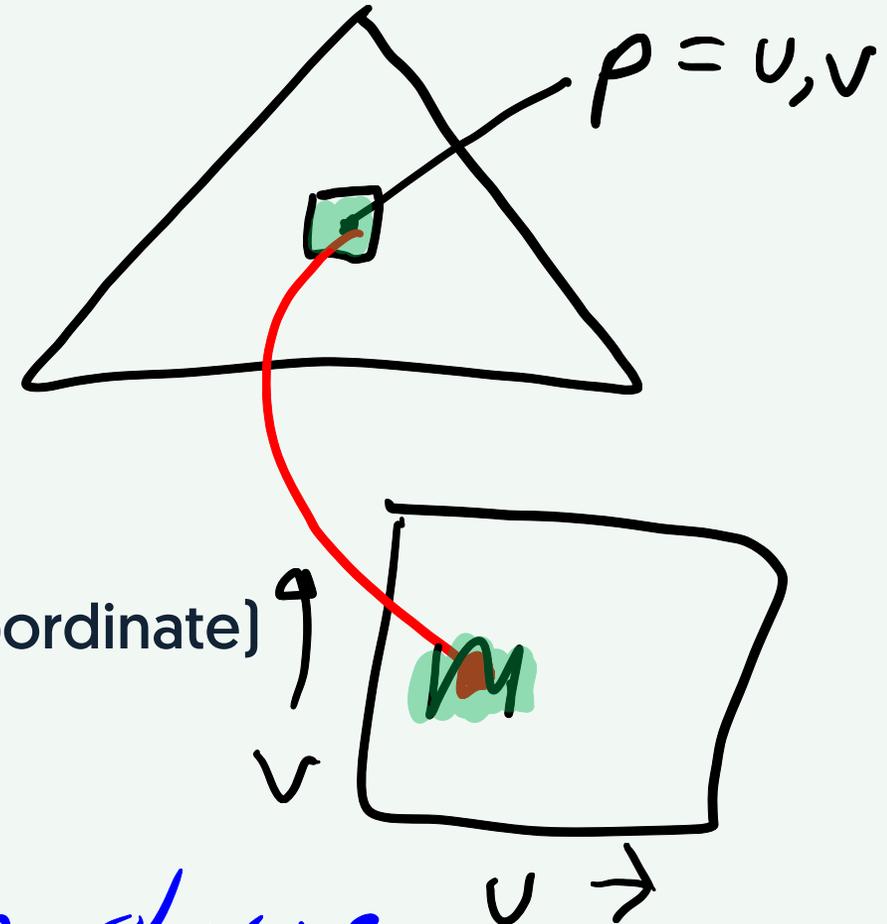
Basic Texture Mapping

1. Define UV coordinates for the triangle
2. Use an image to map UV to color
3. **Use the color as the surface color**

For each pixel inside triangle:

1. get the Barycentric coordinate
2. use this to interpolate the UV values (get UV coordinate)
3. use the UV coordinate to look up the color
4. use the color as the surface color

inside graphics hardware

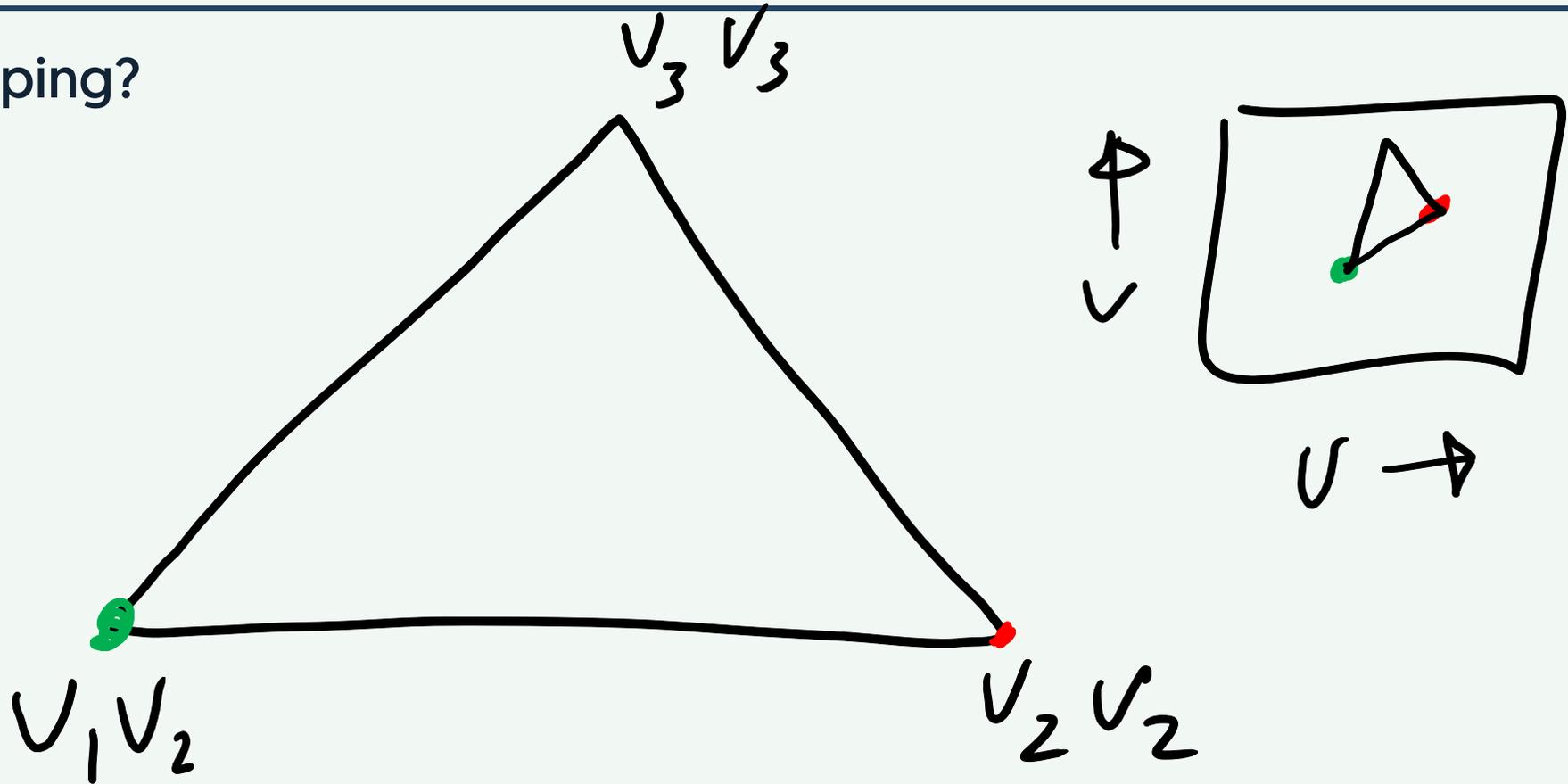


Demo

(available online)

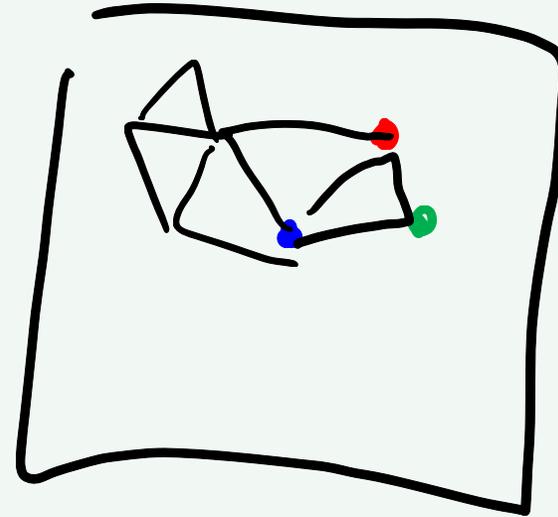
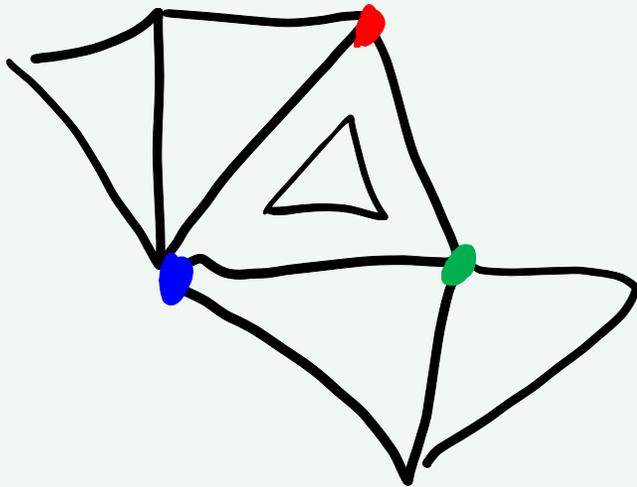
UV Mapping

Or is it ST Mapping?



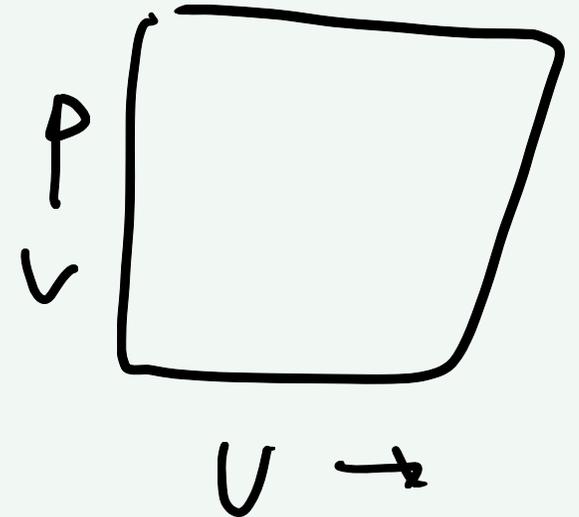
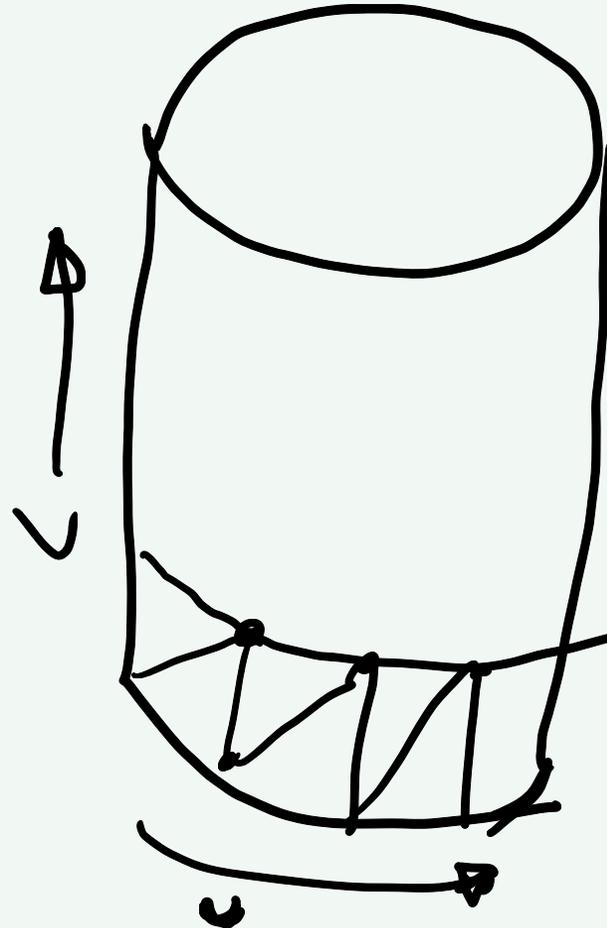
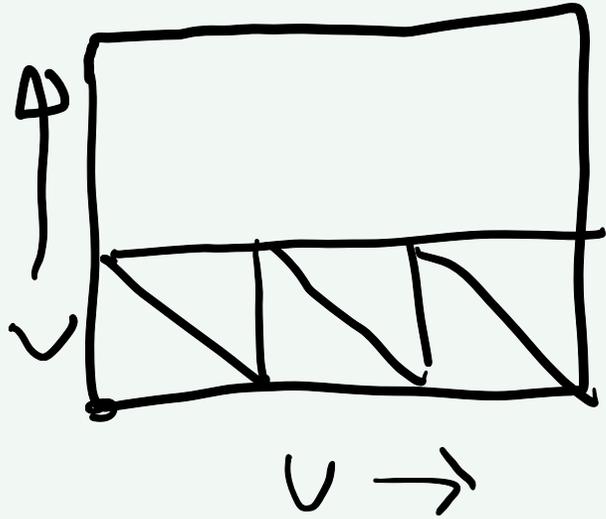
UV mapping and multiple triangles

Pack the triangles into the UV plane



UV mapping for standard objects

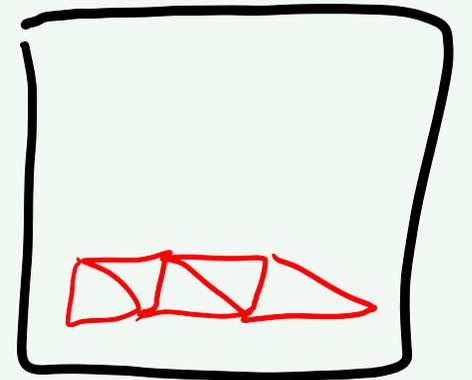
Many objects have a natural parameterization



Where do UVs come from?

We specify the UV values!

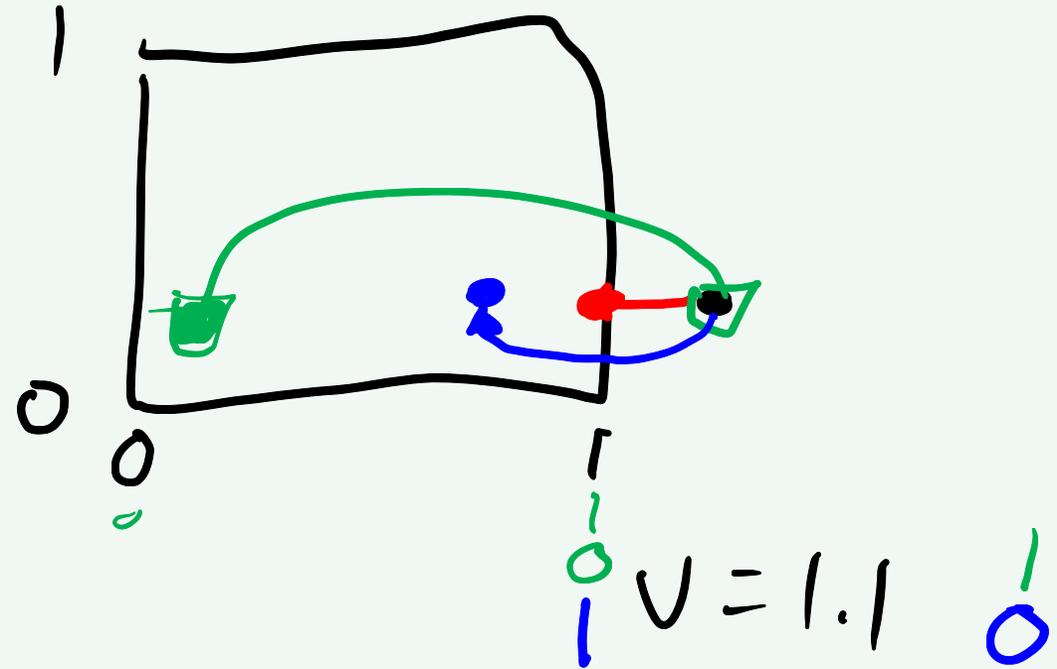
- If you have the image to start with
 - pick UV so the triangle gets the right piece
- If you don't have the image first
 - pick UV so each triangle is "nice" in UV space



Texture Wrapping

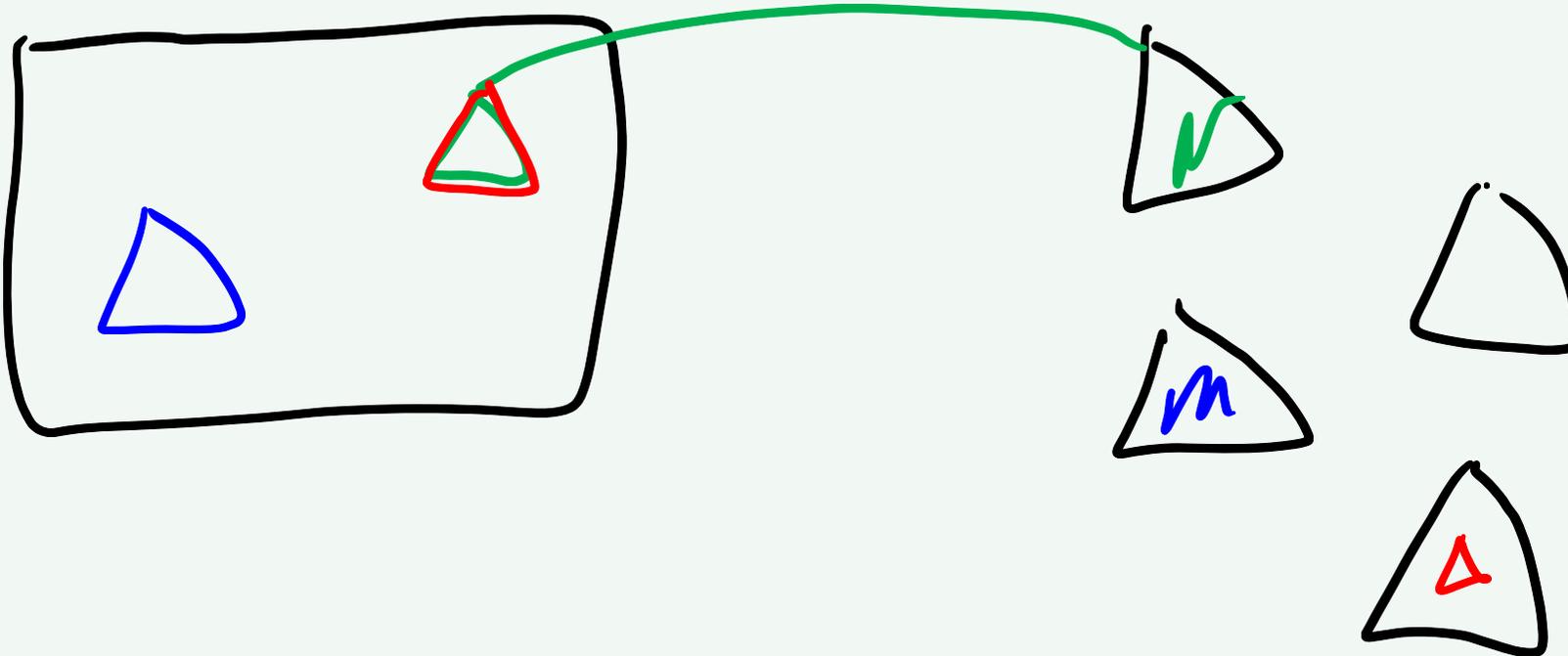
What if U or V outside of $[0, 1]$?

- clamp wrapping
- repeat wrapping
- mirror repeat wrapping



Texture use and Re-Use

Loading, processing and storing textures is resource intensive

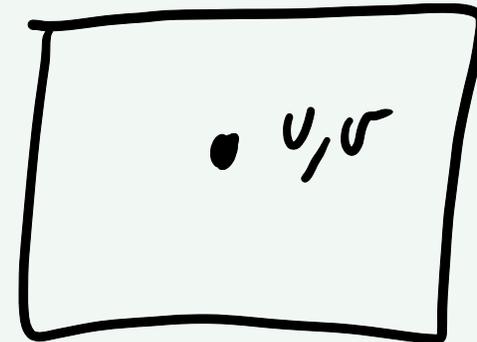
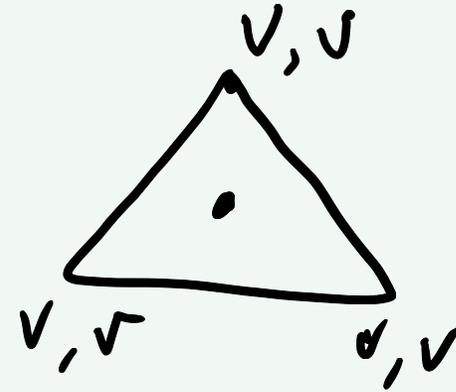


Steps for using Texture Mapping

1. Specify UVs for Triangles
2. Specify an Image to be used for lookup
3. Specify how to use the image in the material

and (a hidden first step)

4. Specify all the different parameters
lookup type, wrapping mode, etc.



Texture Basics

1. Specify UVs per vertex
2. Look up colors in an image
3. Use looked up value in an image

Texture (Beyond) Basics

1. Specify UVs per vertex
compute the UVs dynamically
2. Look up colors in an image
use something other than simple lookup for map
3. Use looked up value as color
use the values for something other than color

just wait, this we'll see examples of all of these