Texture mapping

CS425: Computer Graphics I

Fabio Miranda

https://fmiranda.me

Overview

- Texture pipeline
- Texture mapping
- Texture filtering
- 1D, 2D, 3D textures
- Mipmapping
- Bump mapping
- Procedural textures

Textures

- Texturing is the process that applies an image, function, or other data source to a polygon.
- Why? Inexpensive way to add realism to a scene.
- Hardware support for 2D and 3D texturing.
- Simple example:
  - Represent a brick wall as a highly detailed geometry with appropriate colors.
  - Significant amount of resources: geometry, memory, rendering
  - Easier to apply an image of a brick wall to a simple polygon.

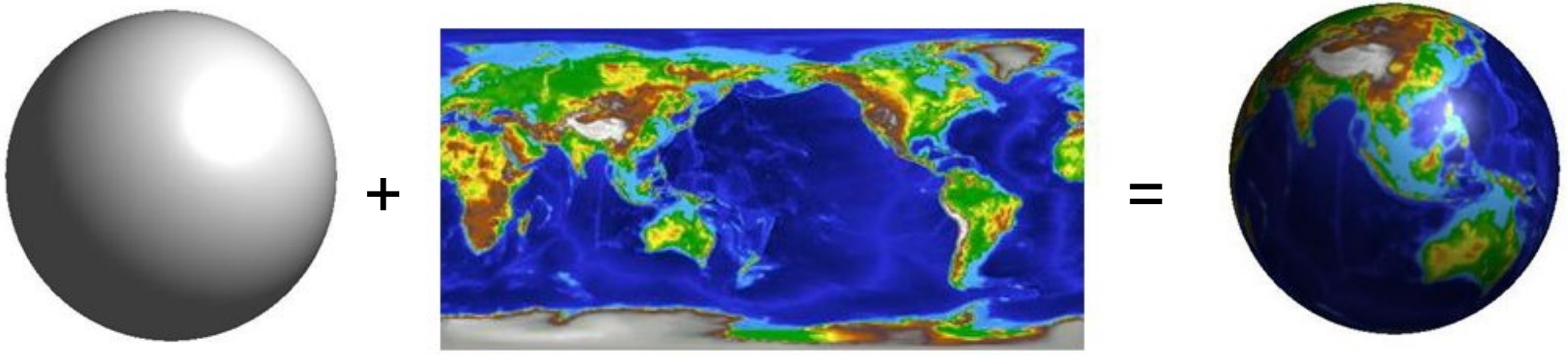
Textures

- Terminology:
  - Texture: an array of values
    - 1D, 2D, 3D.
    - Store color, alpha, depth, and even normal.
  - Texel: a single array element.
  - Texture mapping: process of mapping texture to geometry.
- Source:
  - Pixel maps: load from an image file.
  - Procedural textures: program generates texel values.

Textures and shading

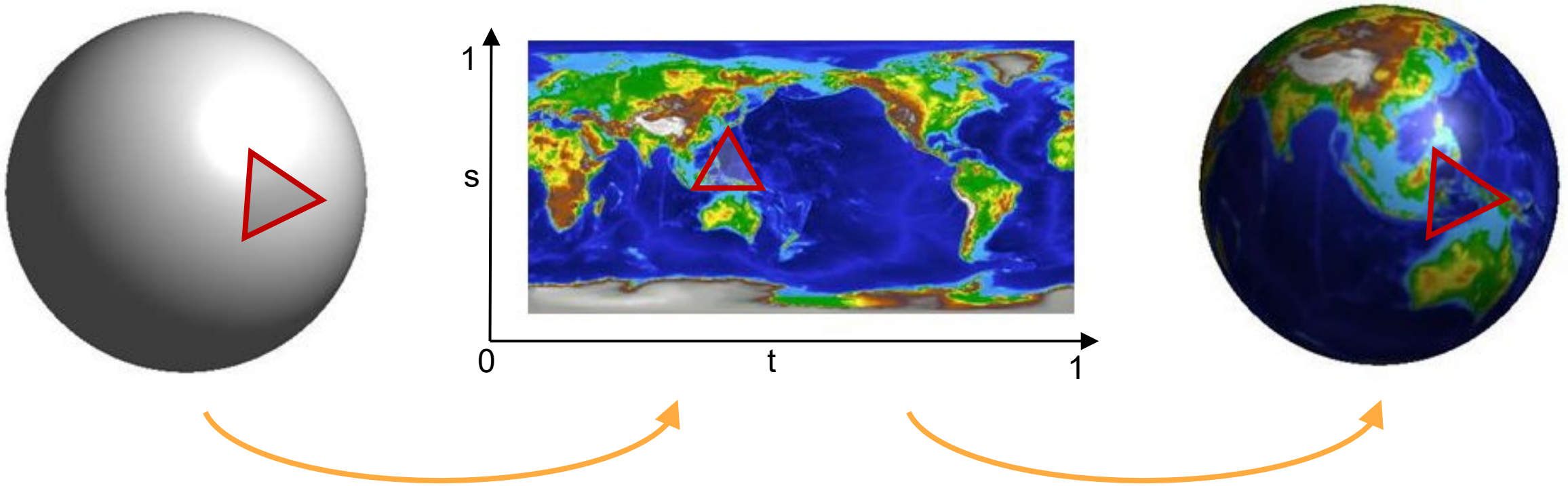
- Shading: takes into account lighting, material, transparency, position of the viewer.
- Texturing: modules the values used in the lighting equation.
  - Replace color by texture color.
  - Modify shininess with specular texture.
  - Modify surface normal with bump maps.

Texture mapping



Object Texture Texture mapped object

Texture mapping



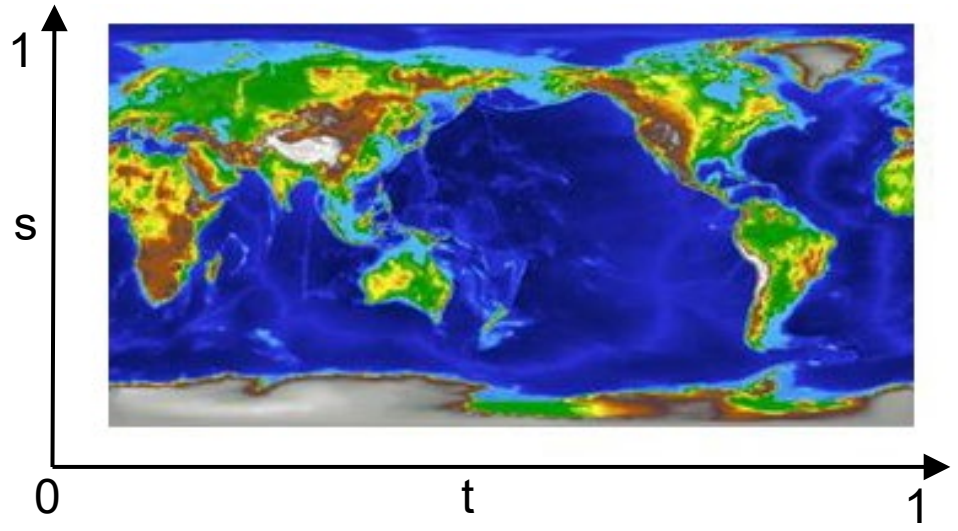
Build a mapping between the texture and the object

Texture mapping

- Problems:
  - Texture and objects are in two different spaces.
  - Where to specify this mapping?
    - Object space.
    - World space.
  - A point on the surface maps to a location between texels in the texture. How to handle that?

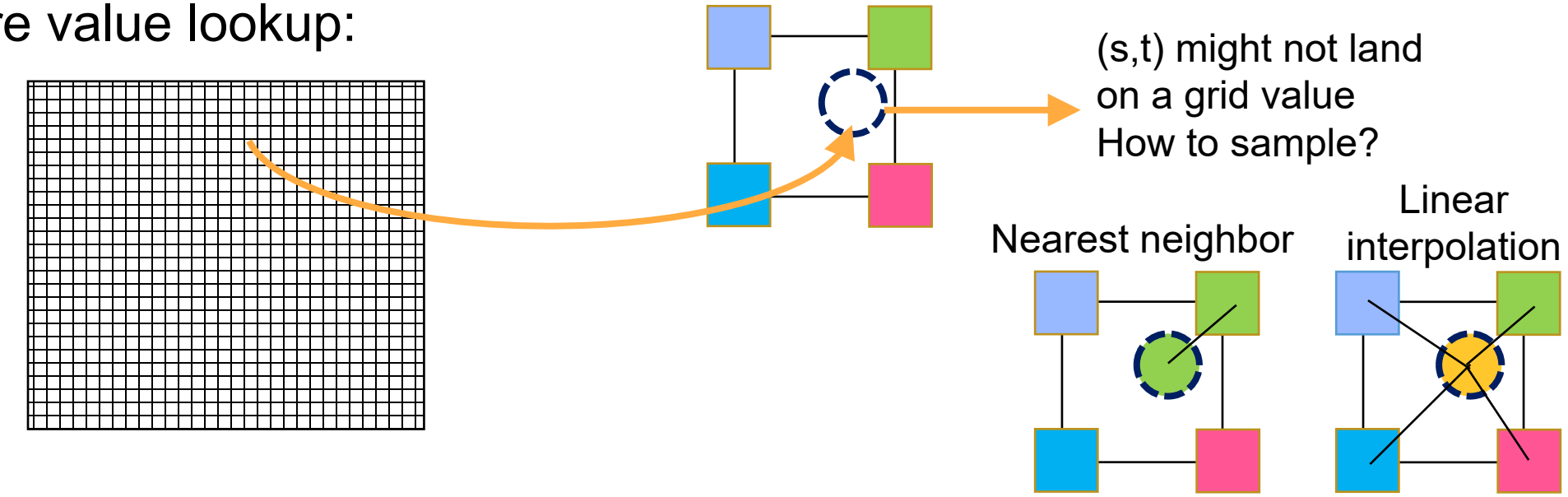
Texture space

- A texture is defined in a normalized space.

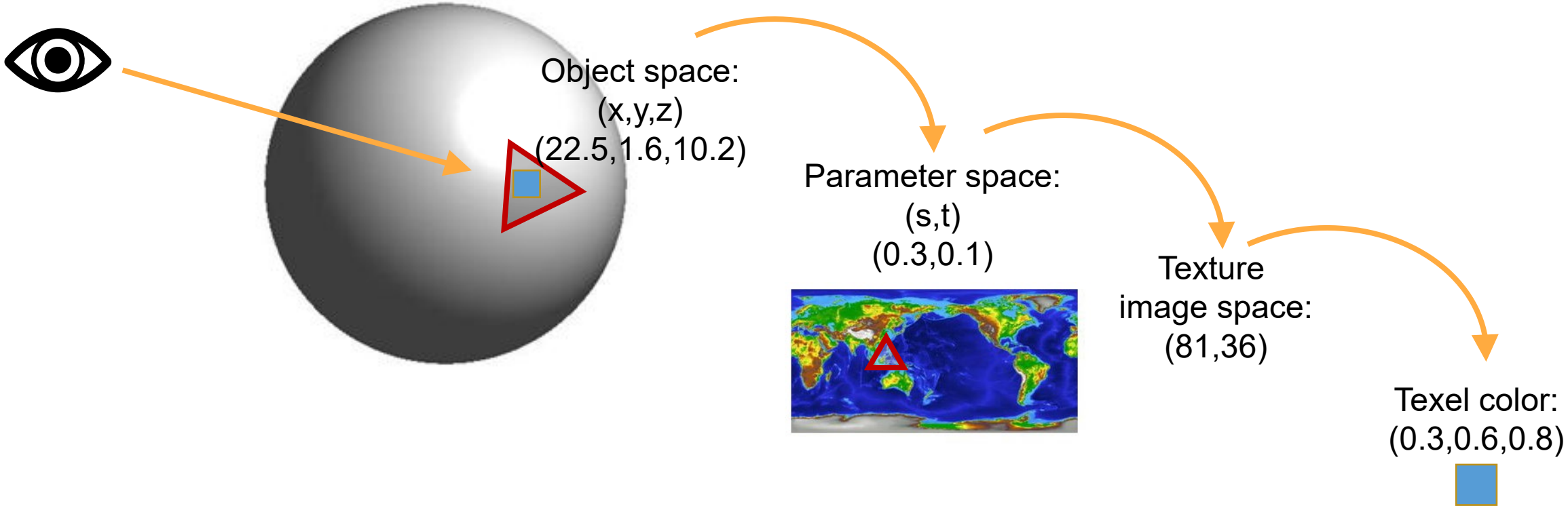


Texture space

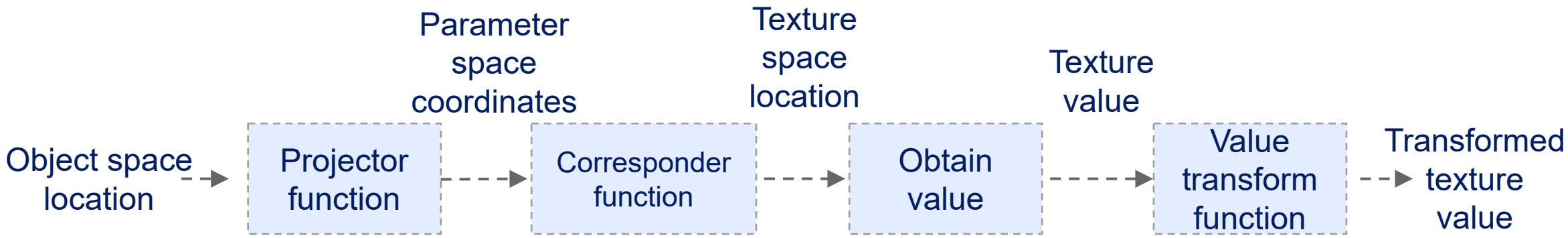
- Texture space is continuous, but textures are discrete arrays.
- Texel values are defined on a cartesian grid.
- Texture value lookup:



Texturing a single fragment



Texture pipeline



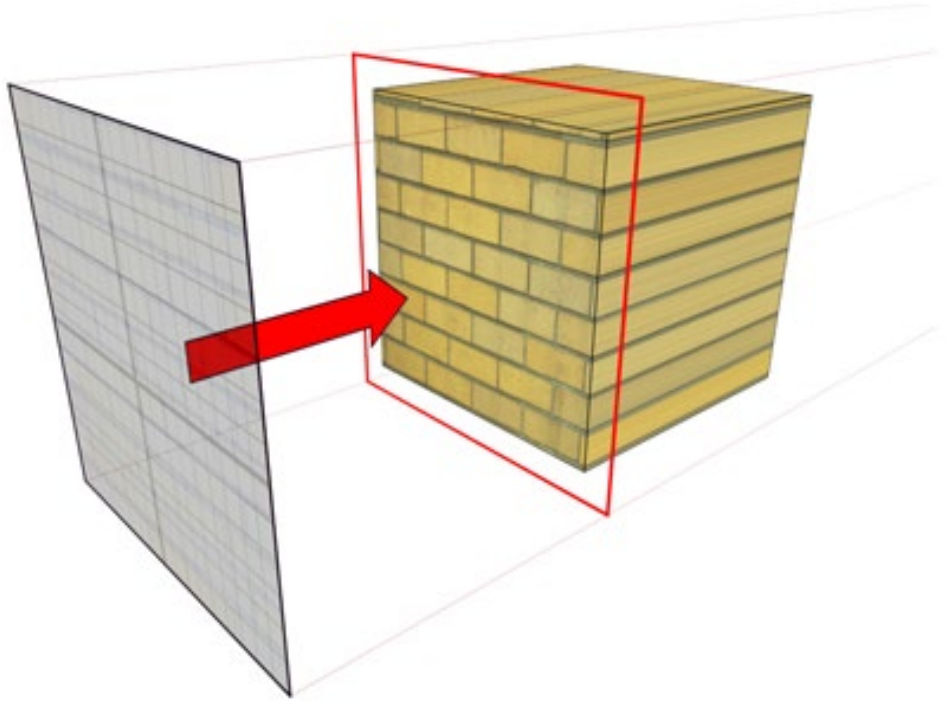
Projector function

- Defines mapping between object (x,y,z) and parameter space (s,t).
- Intermediate mapping:
  - Map the texture onto a simple intermediate surface.
  - Map the intermediate surface to the final object.
  - Intermediate objects:
    - Plane
    - Sphere
    - Cylinder

Planar mapping

- Convert (x,y,z) point to (x,y).
- Simply drop z coordinate:

$$s = x$$
$$t = y$$



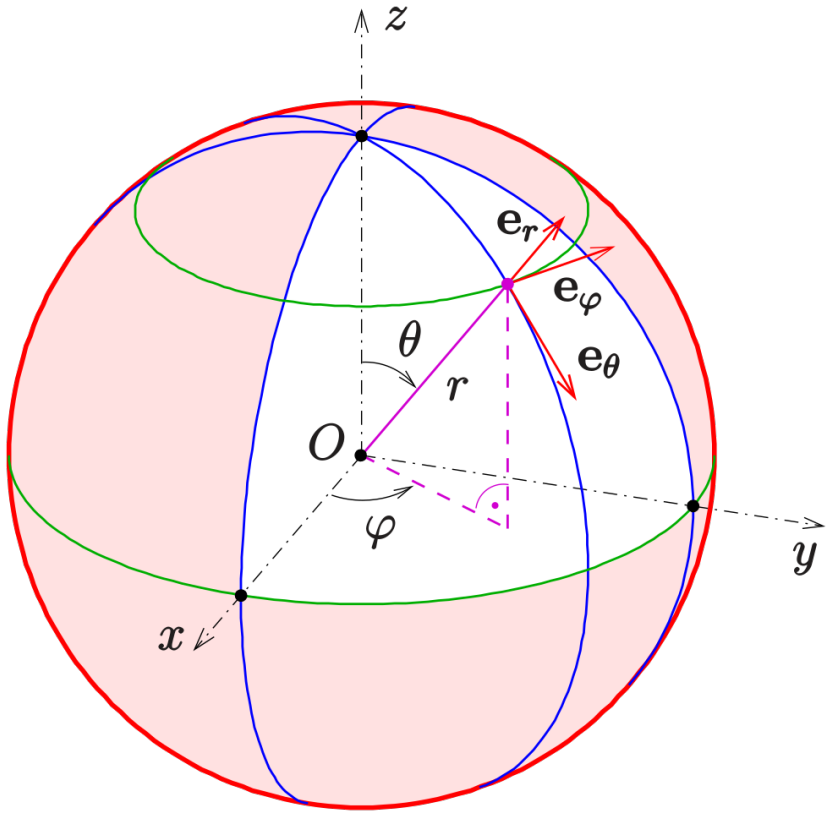
disguise.one

Spherical mapping

- Convert (x,y,z) point to spherical coordinate (θ, φ).

$$\theta = tan^{-1}\left(\frac{\sqrt{x^2 + y^2}}{z}\right)$$

$$\varphi = tan^{-1}\left(\frac{y}{x}\right)$$

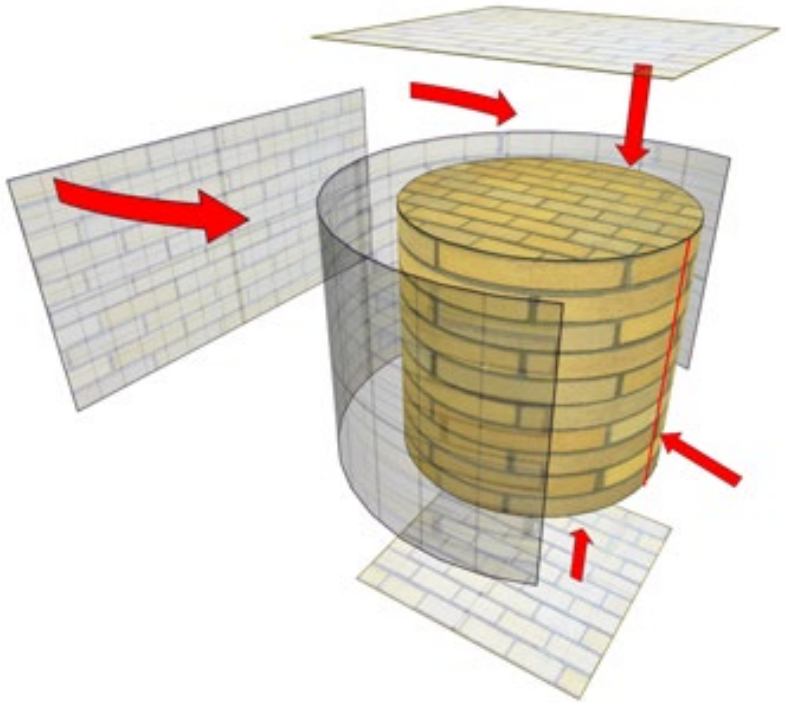


Cylindrical mapping

- Convert (x,y,z) point to cylindrical coordinates (θ, ρ).

$$\theta = tan^{-1}\left(\frac{y}{x}\right)$$

$$\rho = \sqrt{x^2 + y^2}$$

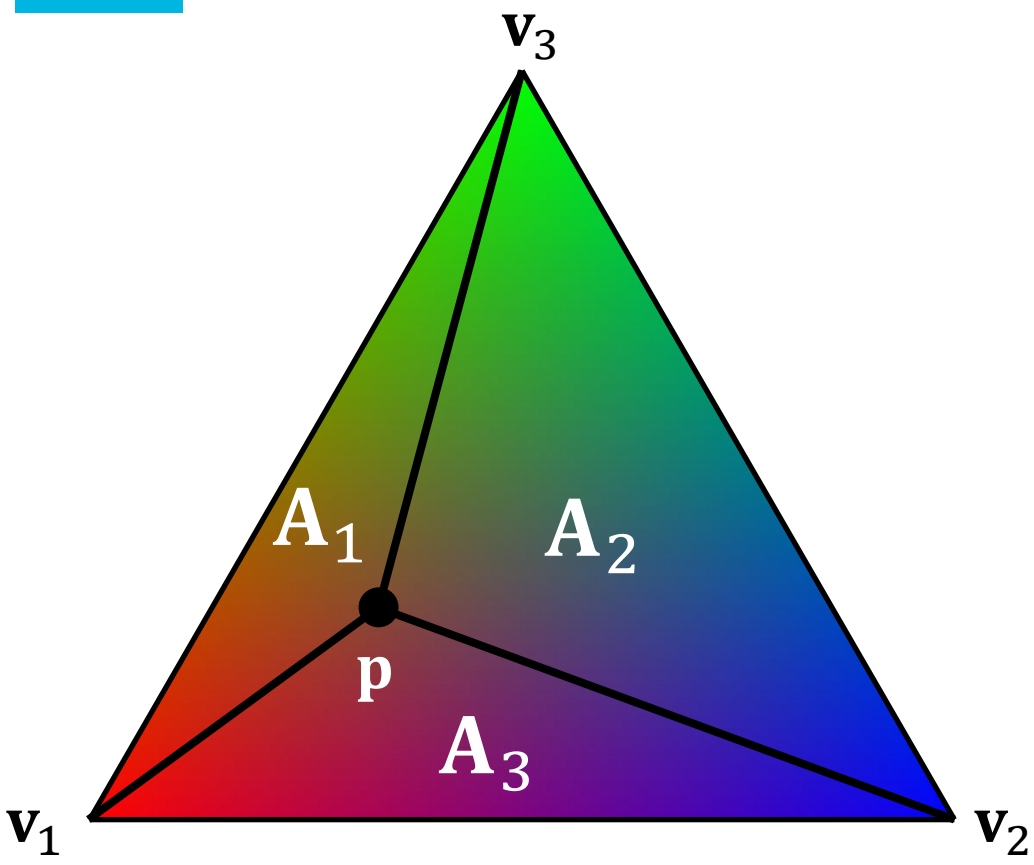


disguise.one

Texture mapping for triangles

- Assign texture coordinates to vertices.
- Interpolate within polygon.

Barycentric interpolation



$$f_p = \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3$$

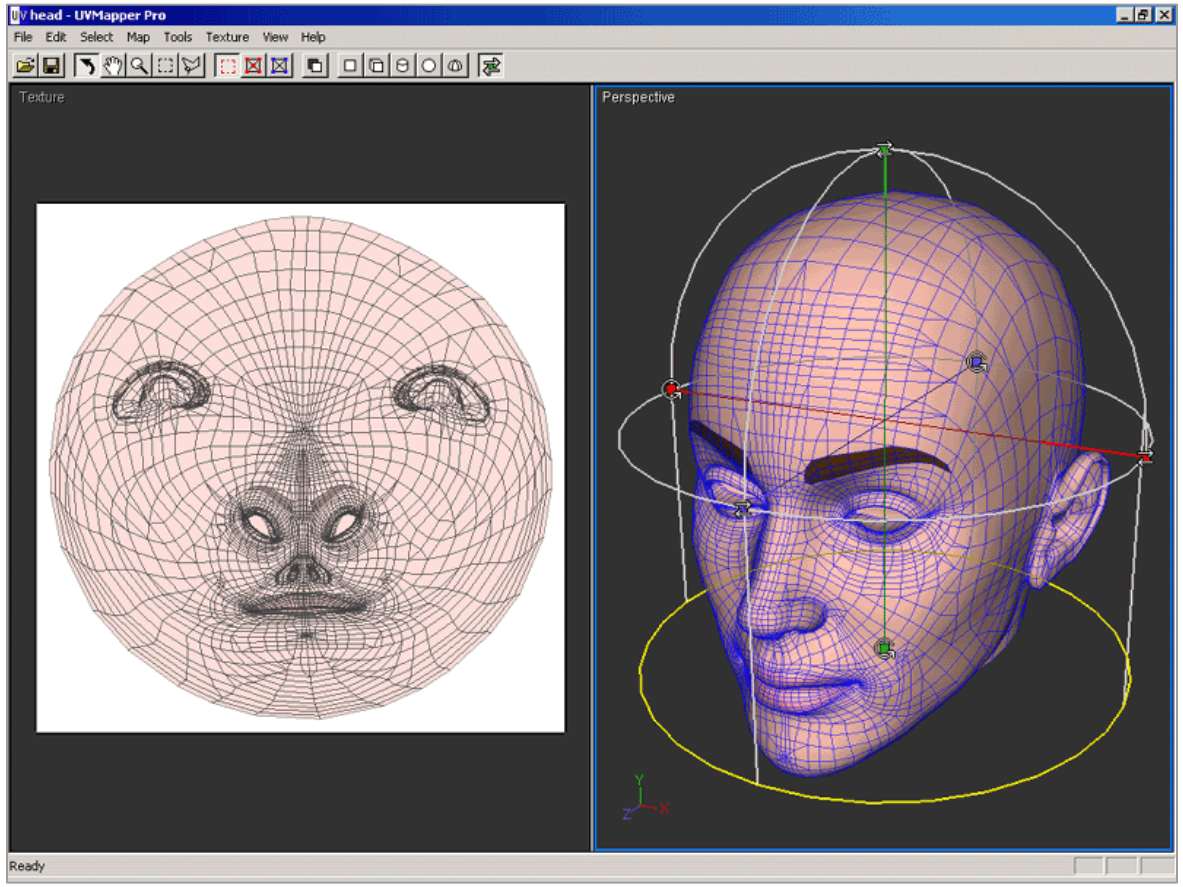
$$\alpha_1 = \frac{\mathrm{A}_1}{\mathrm{A}_{total}}$$

$$\alpha_2 = \frac{\mathrm{A}_2}{\mathrm{A}_{total}}$$

$$\alpha_3 = \frac{\mathrm{A}_3}{\mathrm{A}_{total}}$$

$$area(\mathbf{qrs}) = \frac{1}{2}\|\mathbf{qr} \times \mathbf{qs}\|$$

Texture mapping for meshes



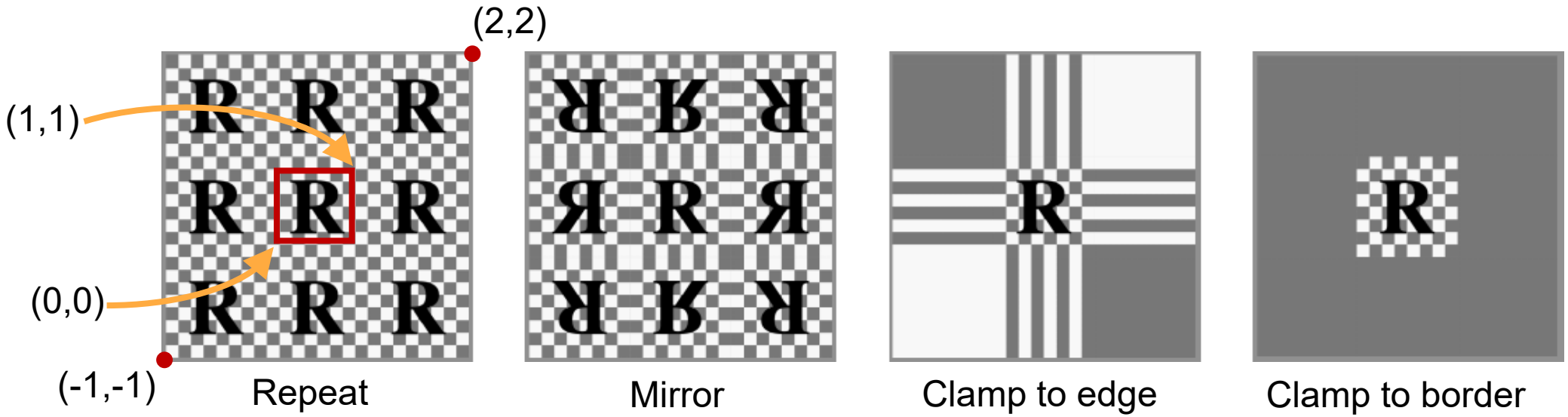
uvmapper.com

Corresponder functions

- Defines mapping between parameter-space values to texture-space locations.
- Flexibility in applying textures.
  - Boundary effects (wrapping modes).
  - Subset of an existing texture.

Wrapping modes

Wrapping modes:

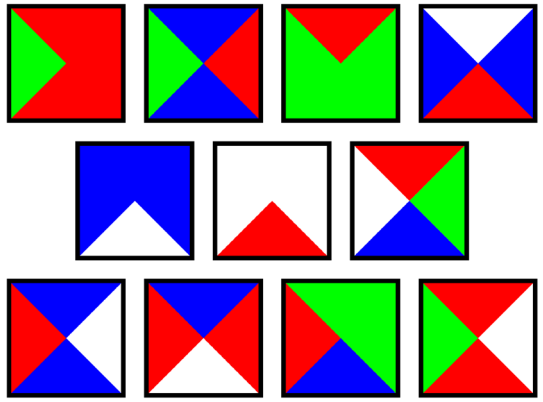


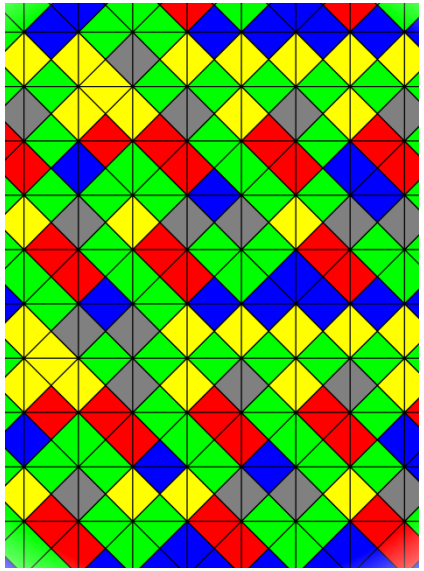
Repeated tiling

- Repeated tiling is an inexpensive way of adding more visual detail to a scene.
- If there are too many repetitions of the texture, it becomes unconvincing because eye picks out the pattern.
- Solutions:
  - Combine values with another, non-tiled, texture.
  - Aperiodic tiling: randomly combine texture patterns or tiles.

Aperiodic tiling: Wang tiles

- Small set of tiles with matching edges.
- Tiles are randomly selected.





"Procedural Wang Tile Algorithm for Stochastic Wall Patterns"