Introduction to Texture Mapping

CSE606: Computer Graphics
Jaya Sreevalsan Nair, IIIT Bangalore
March 19, 2025

Introduction to Textures

Material Properties and Shading

Assigning material properties, lighting effects and shading schemes can help produce different effects and improve realism

Default approach is to assign colors to vertices of the surface mesh (or uniformly to all vertices of the mesh), and use this to compute colors of interior points of the mesh polygons (during fragment processing)

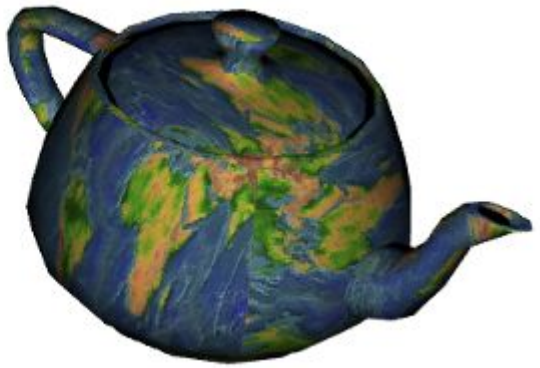
Edward Angel: Interactive Computer Graphics

More interesting effects!

How do we produce more interesting effects?

Need to compute fragment values that are not directly derivable from the properties of the vertices.

This must be done with low (additional) performance overhead.

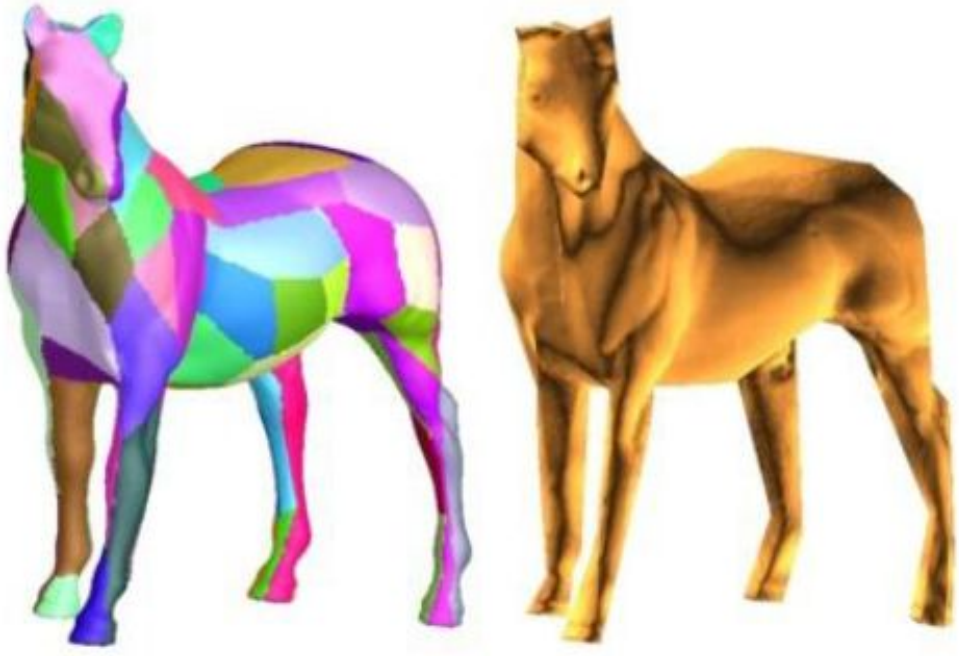


How are these images generated?

Image: Pixar

Edward Angel: Interactive Computer Graphics

Image: Hughes Hoppe



A sphere or an orange?

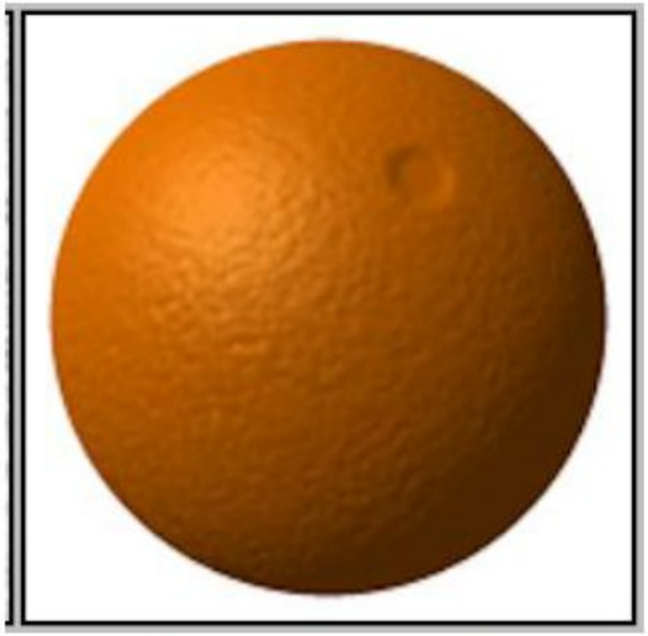
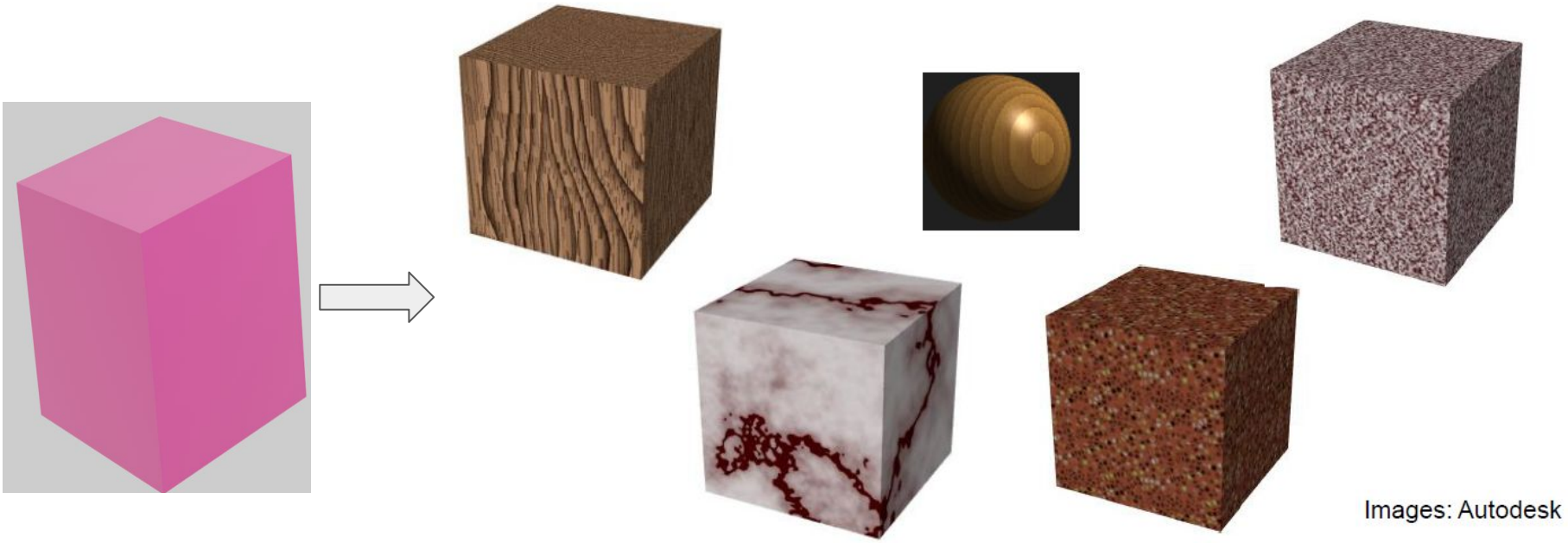


Image: Wikipedia Commons

“Natural” Appearance for Geometric Objects



Images: Autodesk

Shading Surfaces Using “Mappings”

- Typically, surfaces are modeled as geometric primitives or mesh surfaces.
  However, rendering such surfaces may not produce realistic images:
  - Rendering tessellated surfaces with material properties make the object very regular.
  - Rendering analytical surfaces will lack the fine detail of the actual surface.

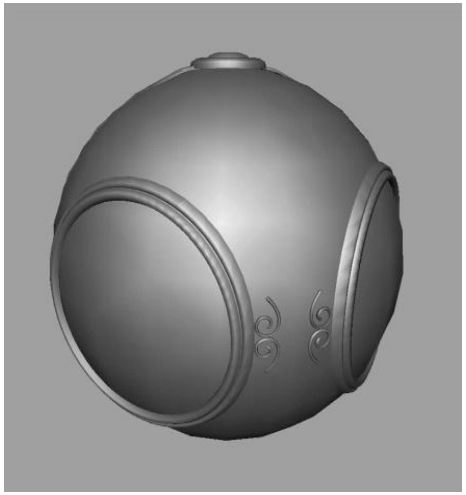
- Alternative: Build a simple model and add details to the rendering.
  - Modify shading algorithm based on a 2-dimensional map
  - Using such maps, modify colour, normals, material properties

Texture Mapping Techniques

- Texture Mapping
  - Uses images to fill inside of polygons
- Environment (reflection mapping)
  - Uses a picture of the environment for texture maps
  - Allows simulation of highly specular surfaces
- Bump mapping
  - Emulates altering normal vectors during the rendering process
- Procedural textures
  - compute texture patterns while rendering

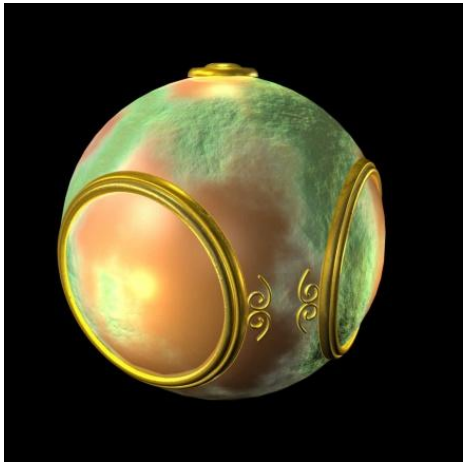
Texture Mapping Techniques

Geometric Model



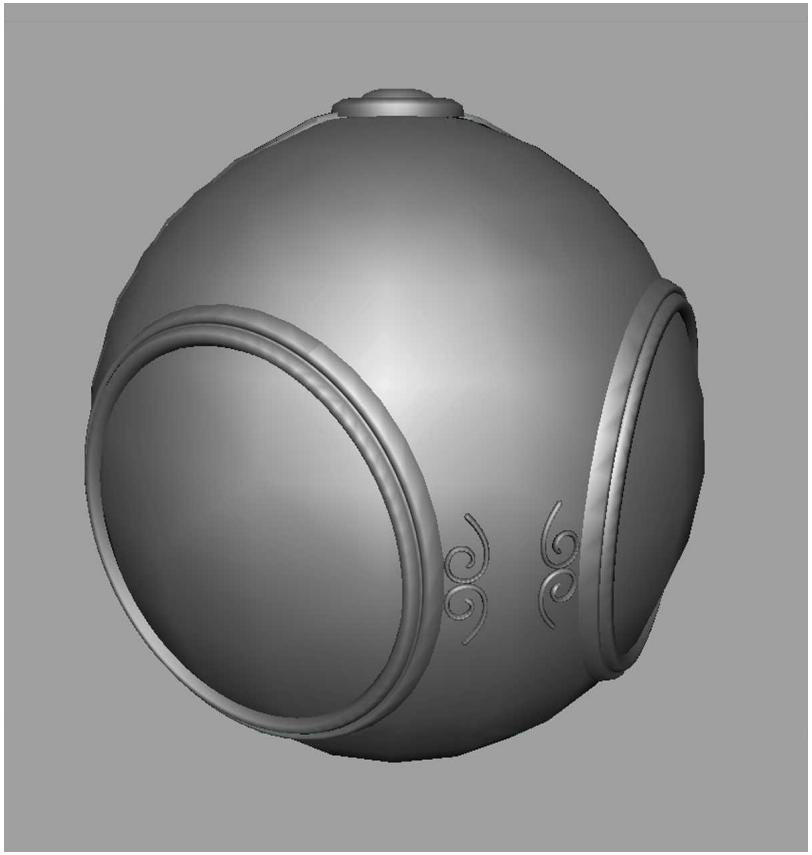
Texture Mapped

Environment Mapped



Bump Mapped

Texture Mapping

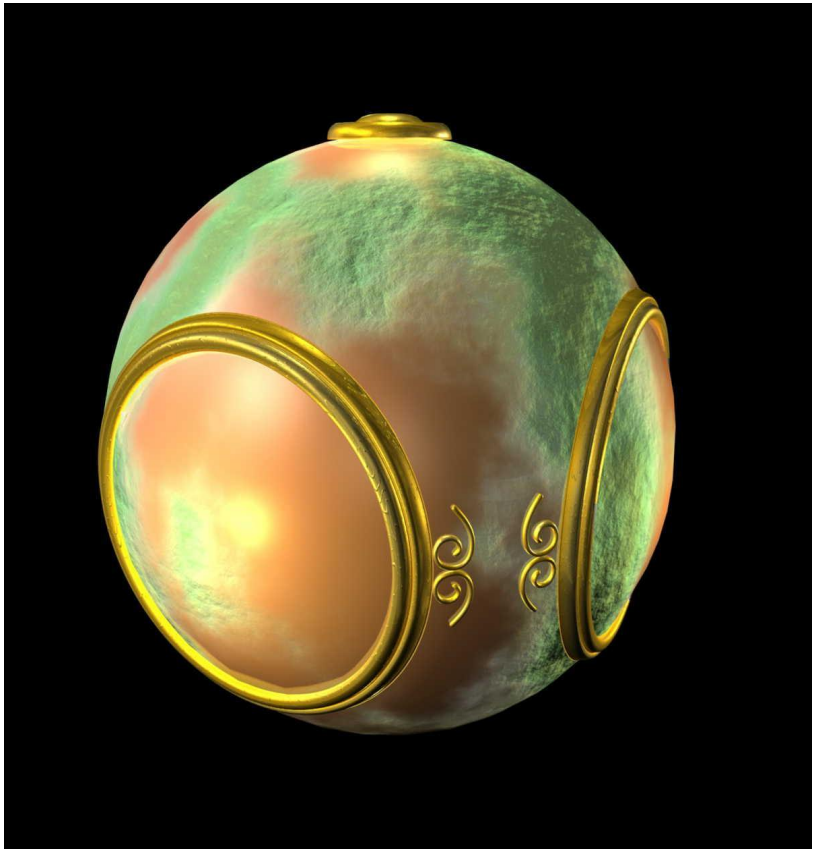


Geometric Model - default shading

Texture Mapped

Texture Mapping Techniques

Environment Mapped



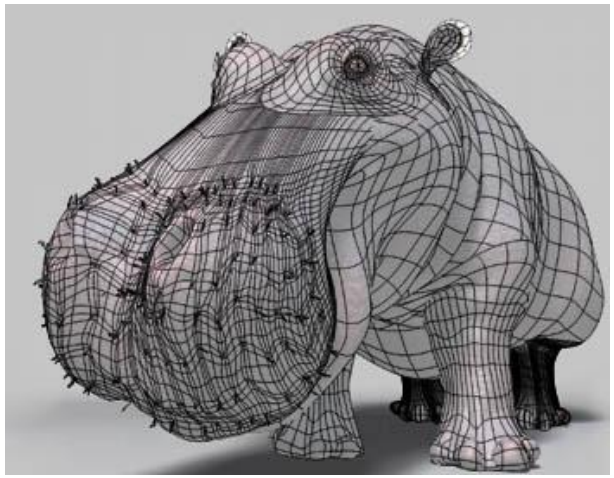
Bump Mapped

Texture Mapping Techniques - Examples

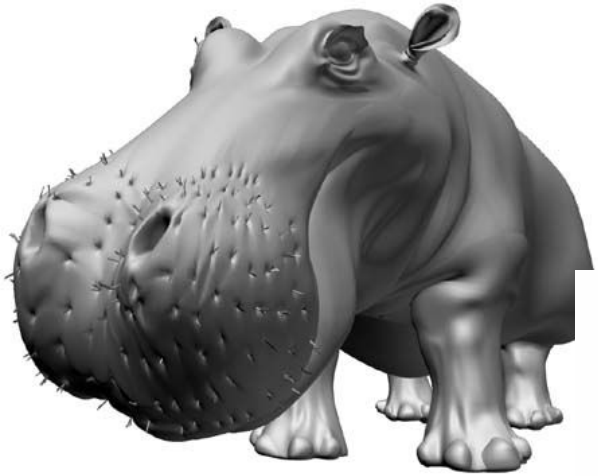
(L) Texture Mapping; (M) Environment Mapping; (R) Bump Mapping.

(Image Courtesy: (L) Mathworks; (M) David Henry; (R) Paul's Project.)

Adding Texture to a Model



Mesh Model



Shaded Model

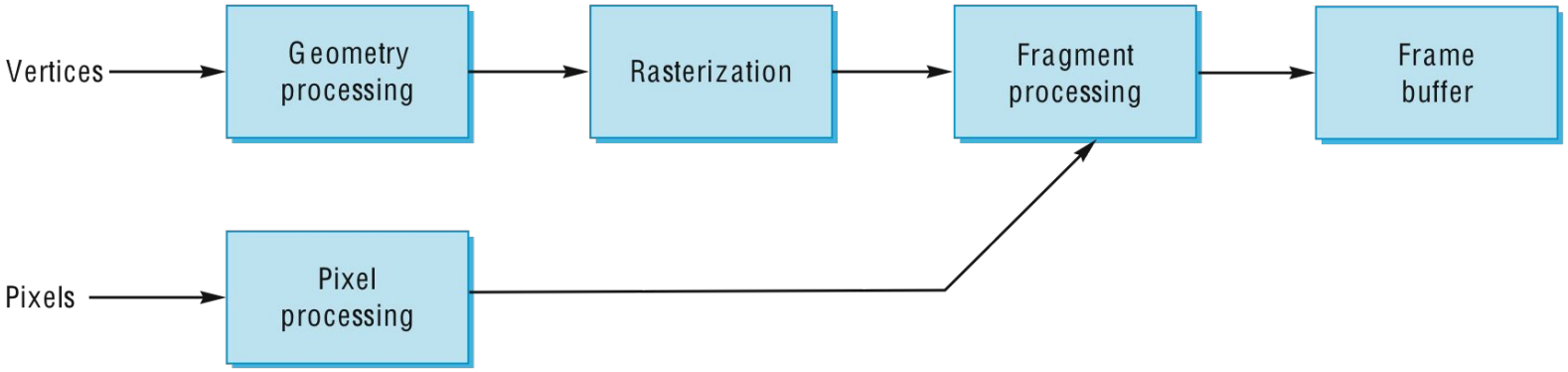
Textured Model

Jeremy Birn: 3drender.com/jbirn

2D Texture Mapping

Textures in the Rendering Pipeline

- Mapping techniques are implemented at the end of the rendering pipeline
  - Very efficient because few polygons make it past the clipper
  - Complexity of textures does not impact geometry processing
- Texture mapping done as part of fragment processing
- Z-buffering follows this



Texture Mapping: Coordinates

Parametric coordinates

- May be used to model curves and surfaces

Texture coordinates

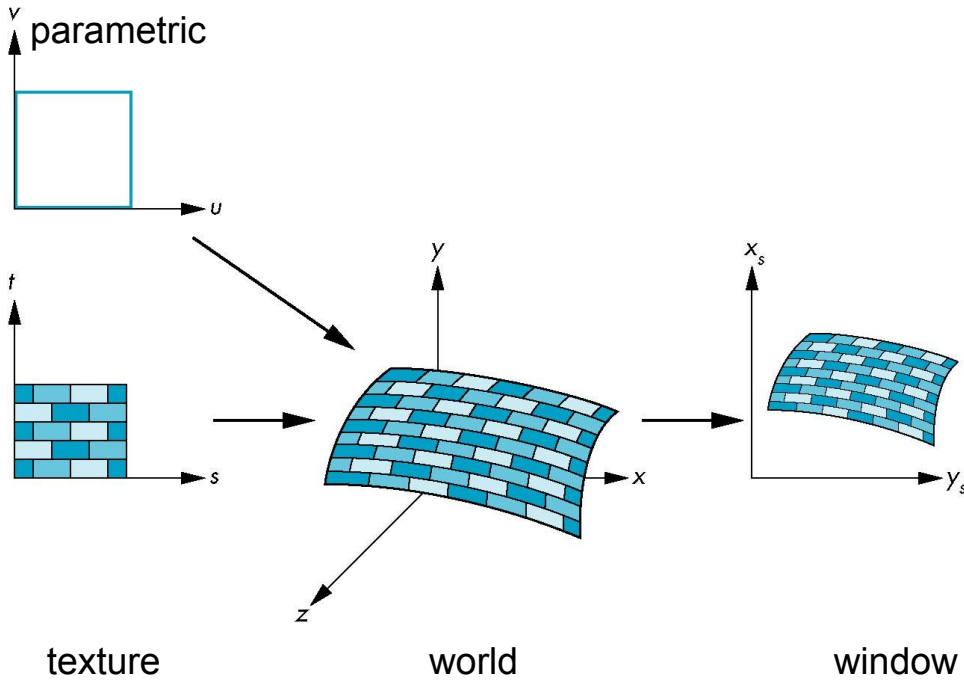
- Used to identify points in the image to be mapped

Object or World Coordinates

- Conceptually, where the mapping takes place

Window/Screen Coordinates

- Where the final image is really produced



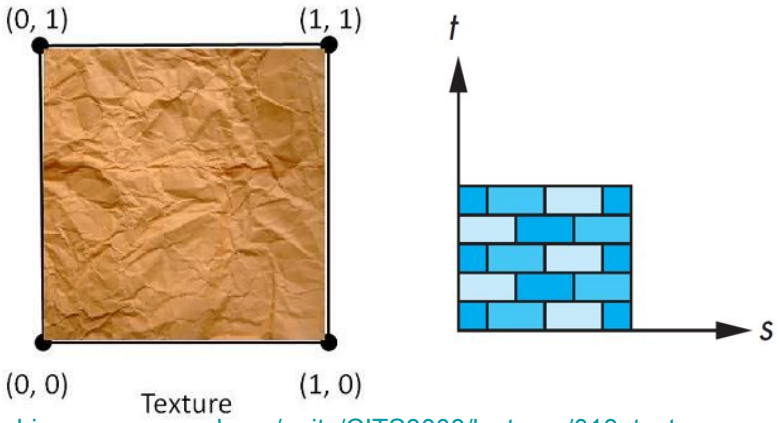
Edward Angel: Interactive Computer Graphics

Basics

Texture Images:

In most applications, textures start out as two-dimensional images

- One can scan texture images from the world (wood, skin, clouds) or paint oneself
- A 2D image - represented by 2D array texture[height][width]



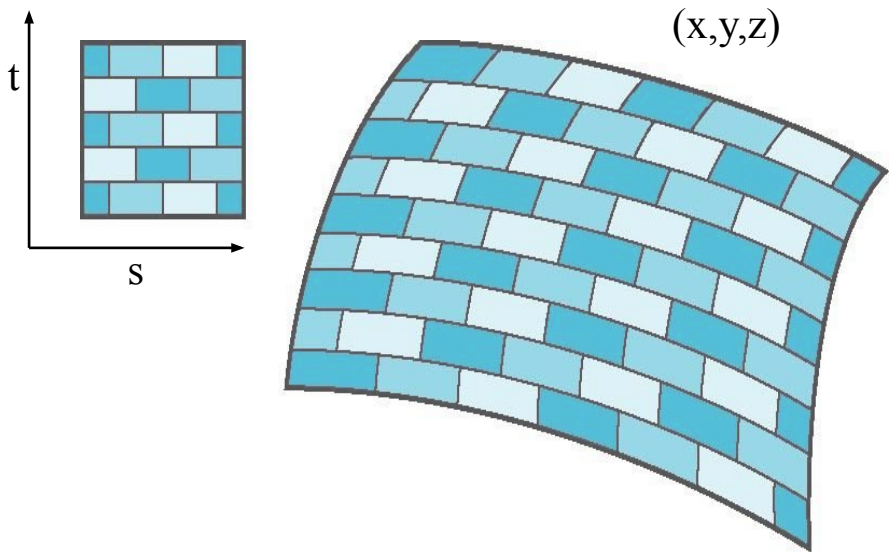
Texture Coordinates

- Used to identify points in the image to be mapped
- s and t normalized to [0,1] range
- s, t are used for the horizontal, vertical coordinates on the image, respectively
- Note that texture coordinates are not based on pixels.
- No matter what size the image is, values of s and t between 0 and 1 cover the entire image.

https://teaching.csse.uwa.edu.au/units/CITS3003/lectures/018_texture_mapping.pdf

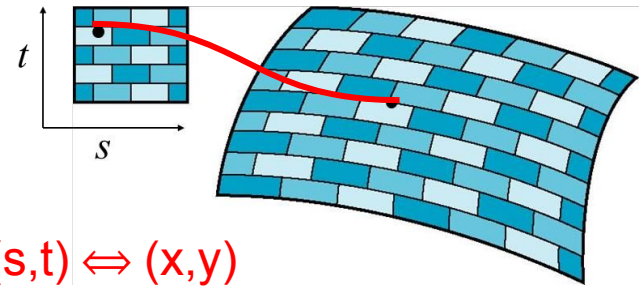
Forward Mapping

- Textures, irrespective of its origin, are stored in processor memory as arrays, whose elements are called **texels**.
  - The array can also be represented as a continuous pattern *T(s, t)*, where *s* and *t* are **texture coordinates.**
- A **texture map** associates each point on a texel to a corresponding point on the geometric object.
- Thus, object coordinates *(x, y, z)* can be represented as (*x(s, t), y(s, t), z(s, t)*)



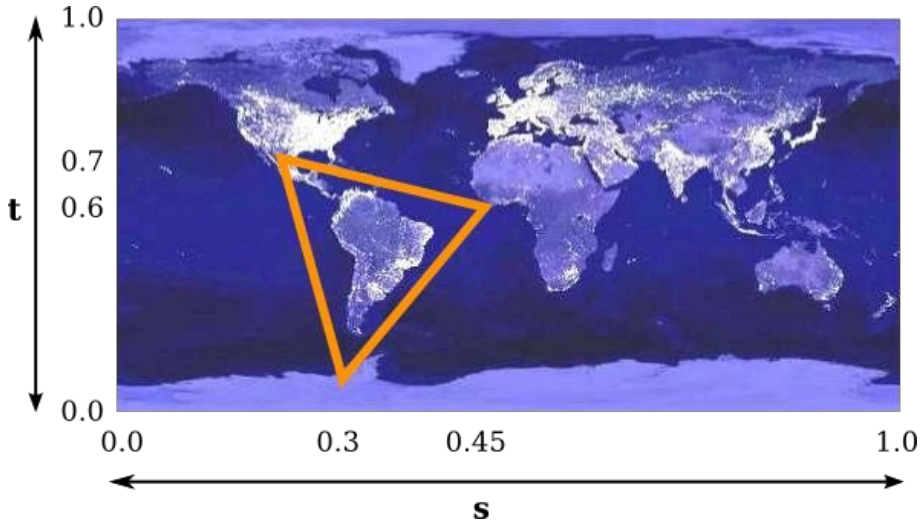
Inverse Mapping

- **Forward Mapping**: Given texture coordinates, we can compute the world coordinates and then the window coordinates for each point on an object



(s,t) ⇔ (x,y)

area in the image that is to be mapped onto the primitive is the triangle (outlined in thick orange)



https://teaching.csse.uwa.edu.au/units/CITS3003/lectures/018_texture_mapping.pdf

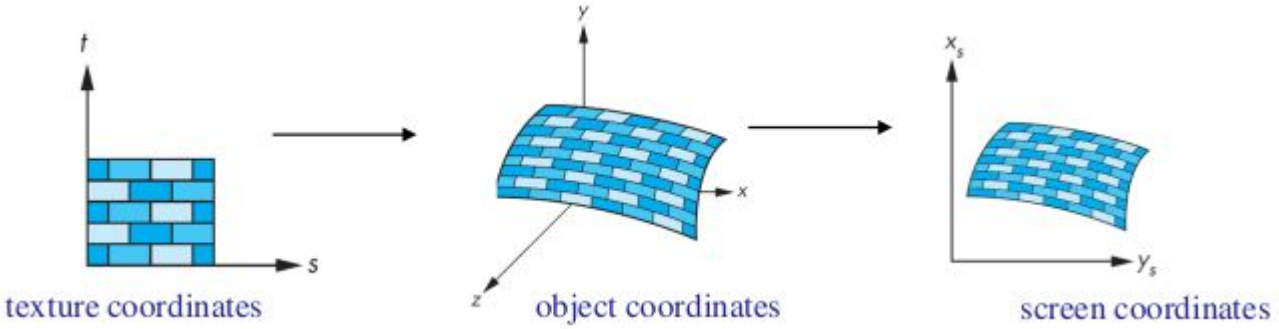
Inverse Mapping

- **Forward Mapping**: Given texture coordinates, we can compute the world coordinates and then the window coordinates for each point on an object
- What we want is the reverse - **Inverse Mapping**:
  - Given a pixel, compute the point on an object it corresponds to (window to world coordinates) - this is forward mapping.
  - Given a point on an object, compute the point on the texture it corresponds to (world to texture coordinates) - this is inverse mapping.
- Need a map of the form:

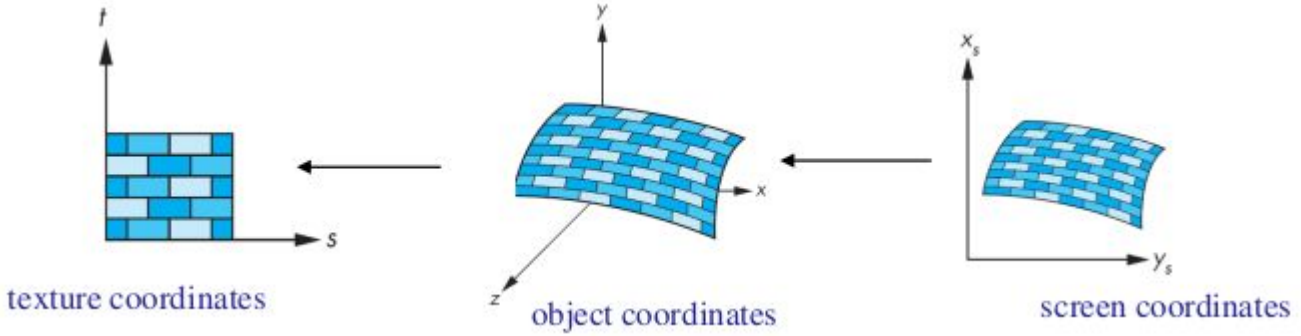
  s = s(x,y,z); t = t(x,y,z)

Forward vs Inverse Mapping

Forward Mapping
(Difficult to render)



Inverse Mapping
(Needed for rendering)



https://teaching.csse.uwa.edu.au/units/CITS3003/lectures/018_texture_mapping.pdf

Types of Textures

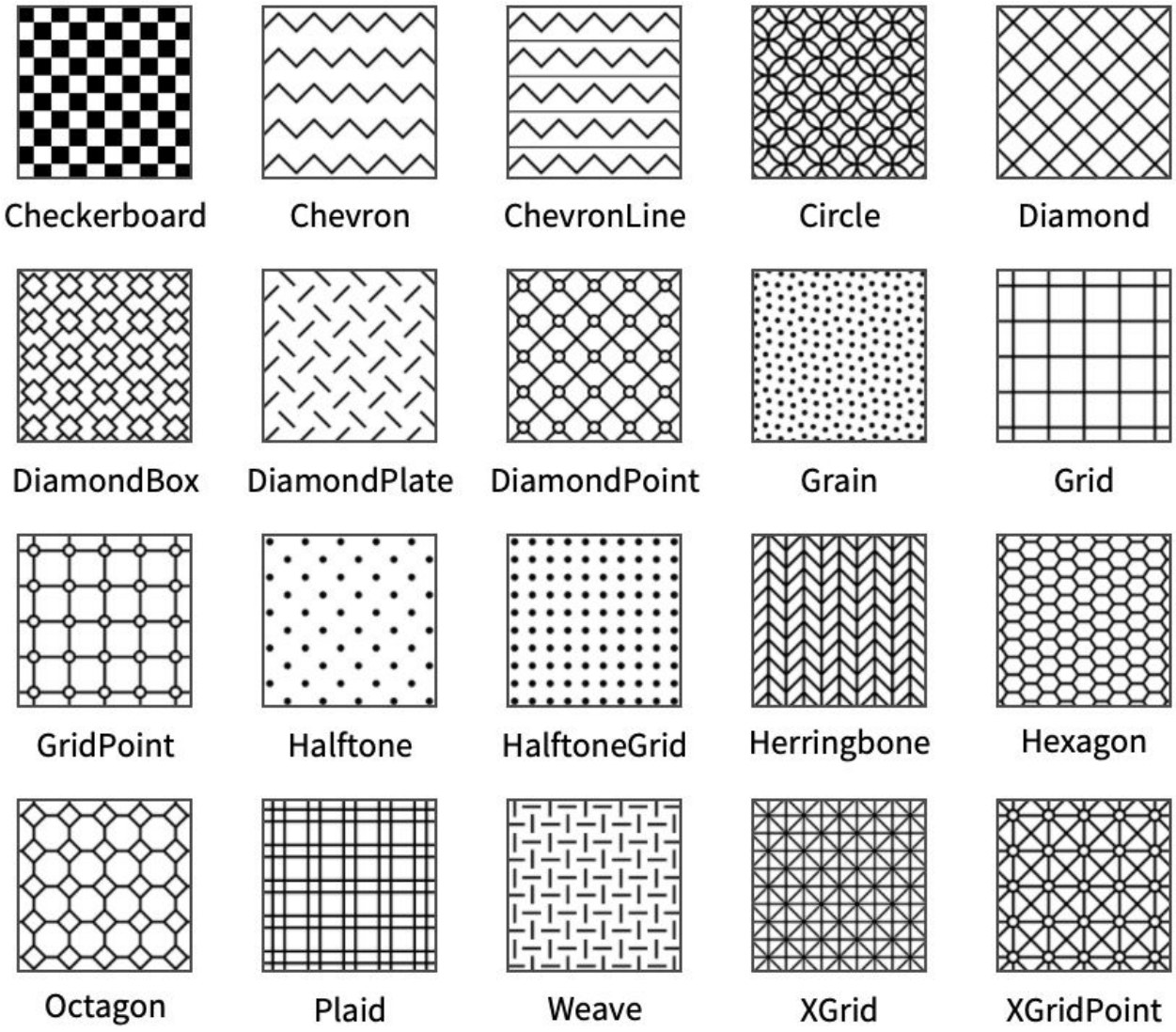
In texture mapping, the texture image, i.e., patterns used for determining color of objects, can be:

A. Fixed pattern, such as the one used for polygon fills,

B. Digitized images,

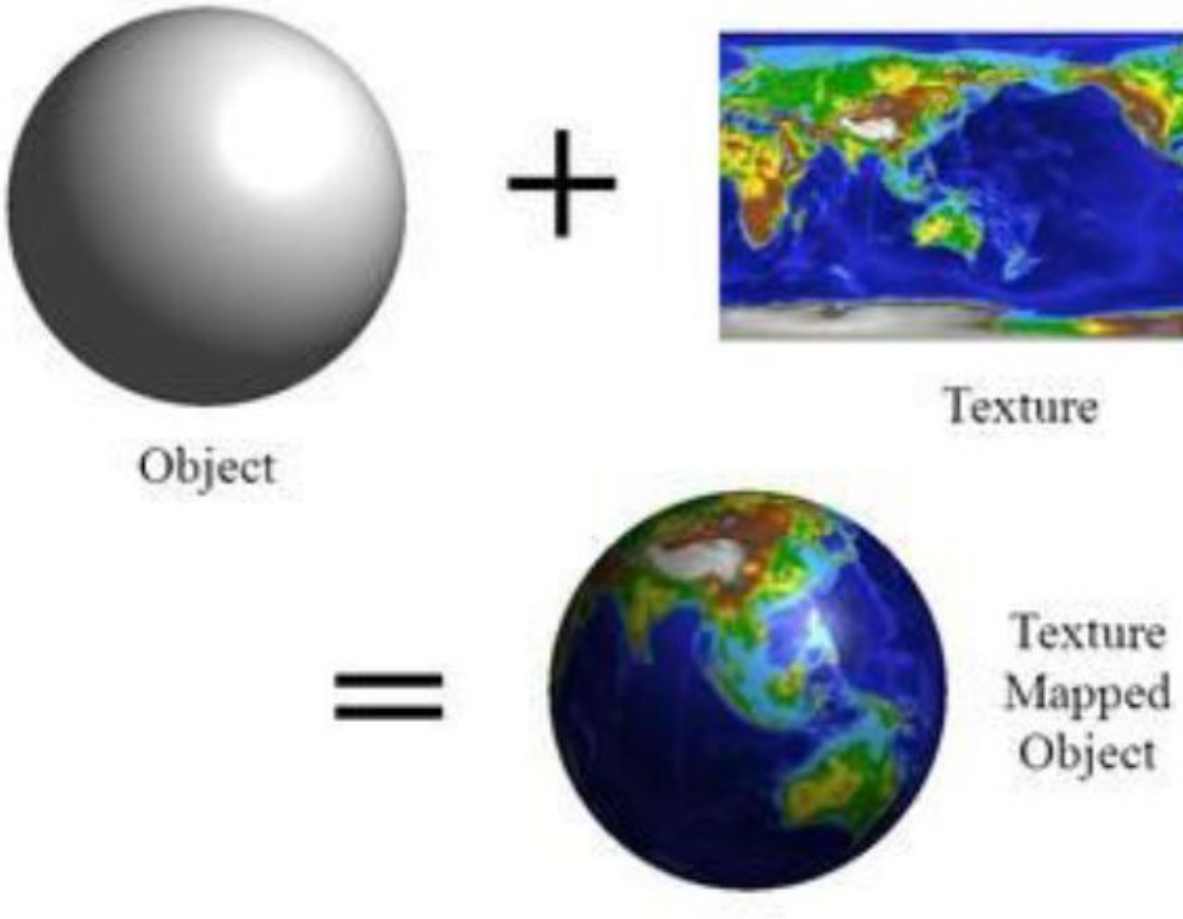
C. Procedural texture-generation method.

A. Fixed Patterns



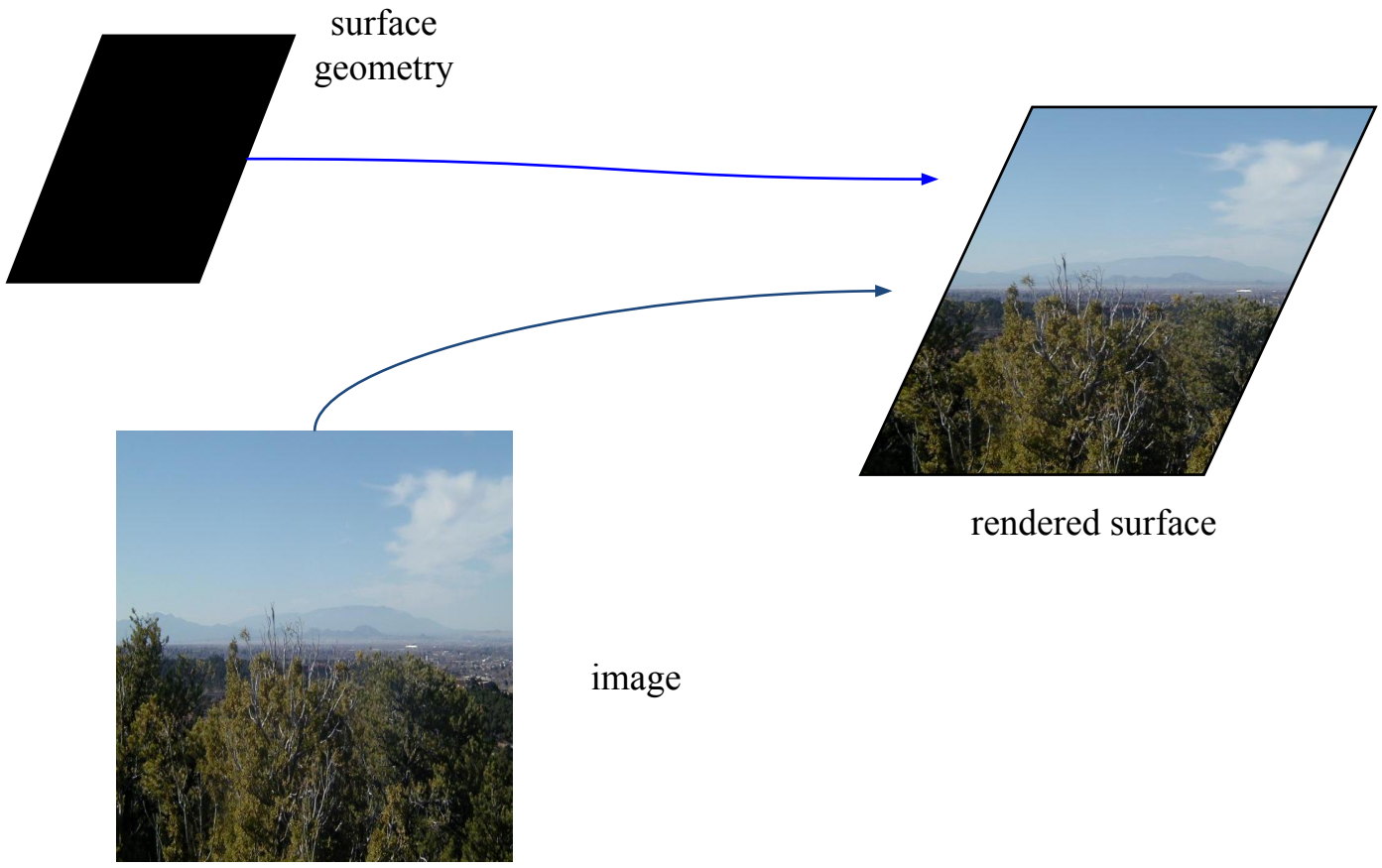
https://reference.wolfram.com/language/ref/PatternFilling.html

B. Digitized Images



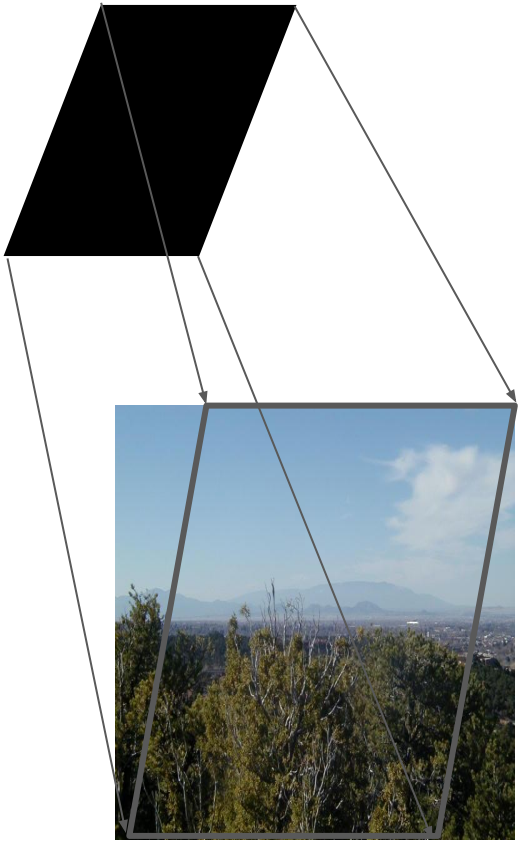
Source: Map a texture onto a model

Texture Mapping Using Digitized Images



Edward Angel: Interactive Computer Graphics

Texture Mapping Using Digitized Images



Surface geometry

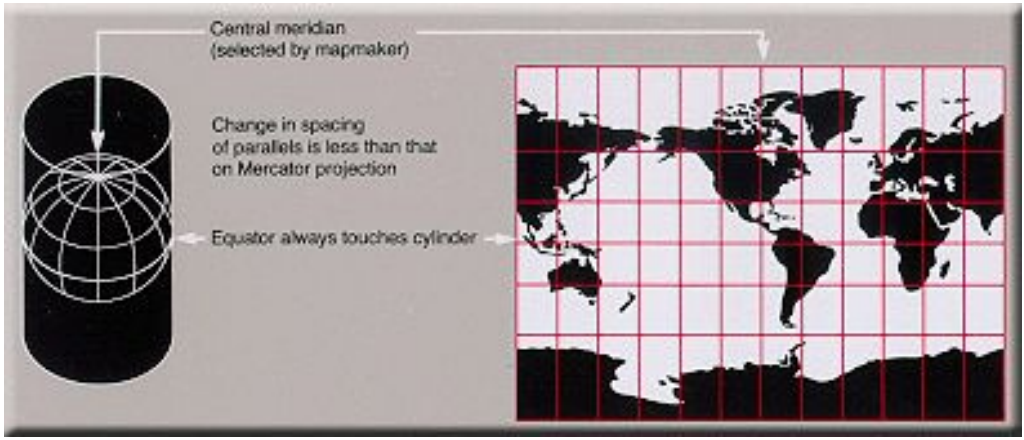
Digitized Image

Two of many possible mappings of texture parameters

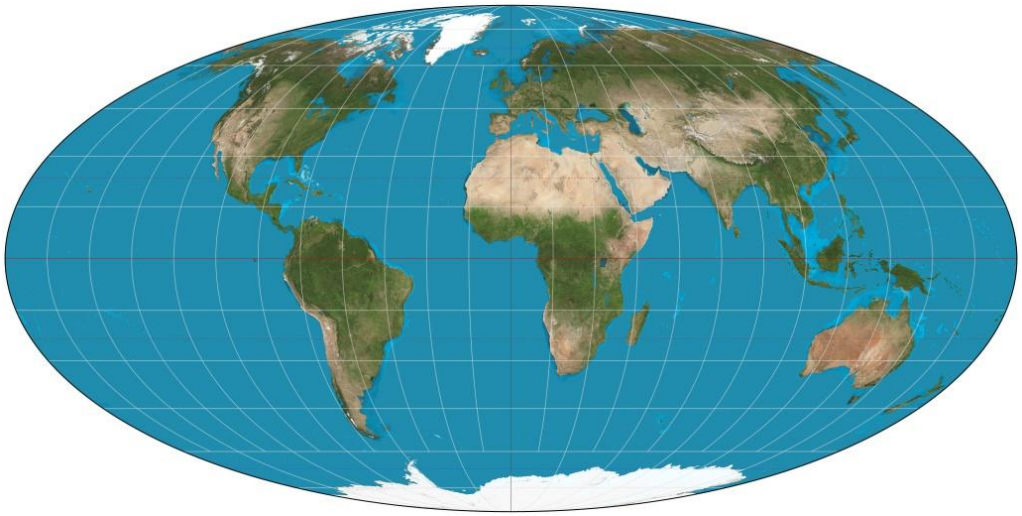
Map of the Earth

What does a 2D map of the world look like?

Mercator projection



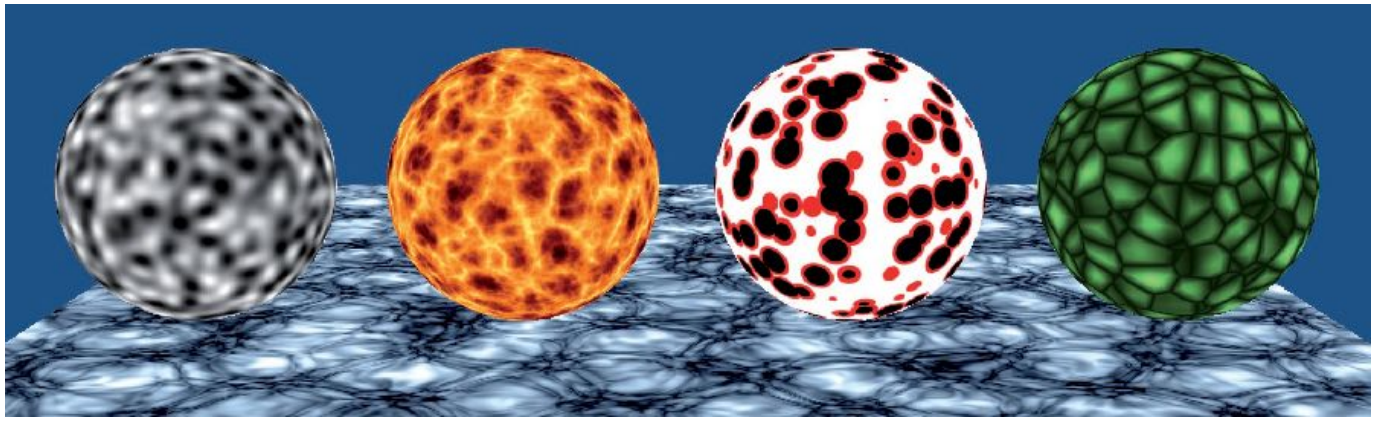
Miller cylindrical projection



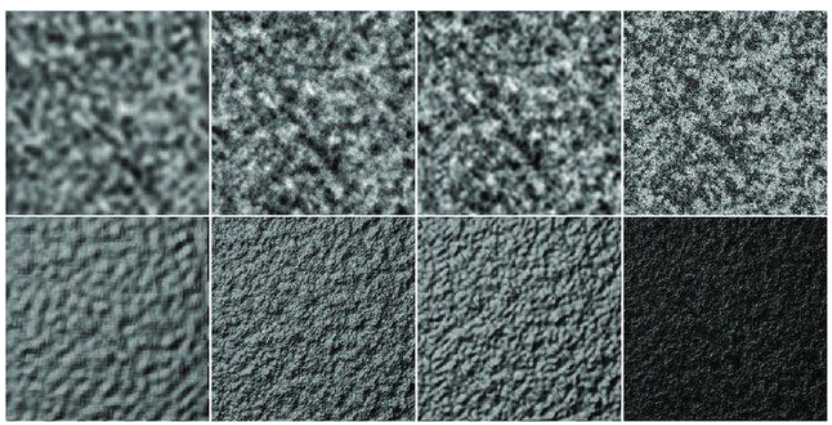
Molleweide projection: equal area

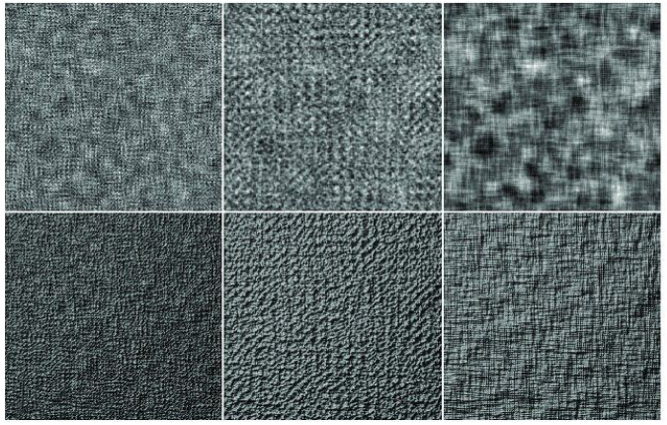
Images and descriptions from Wikipedia Map_projection

C. Procedural Textures



Perlin Noise



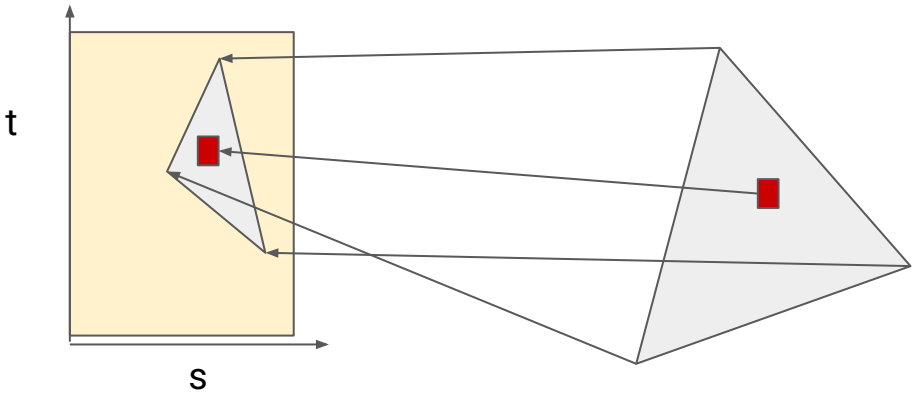


Wavelet Noise

Gustavson, S., 2012. Procedural textures in GLSL. *OpenGL Insights*, pp.105-120.
Dong, J., Liu, J., Yao, K., Chantler, M., Qi, L., Yu, H. and Jian, M., 2020. Survey of procedural methods for two-dimensional texture generation. *Sensors*, *20*(4), p.1135.

Using Textures in WebGL Shaders

1. Specify an image as the texture for the surface being rendered
2. Compute/assign texture coordinates for each vertex of the mesh
   a. Define a 2d vector attribute, similar to the vertex coordinates and normals
3. In the *vertex shader*, pass this through to the output (varying) attribute
4. This is made available to the *fragment shader* as the linear interpolation of the texture coordinates of corner vertices of the triangle being rendered
5. Use this (*s,t*) value to lookup the color in the texture image array



Texture image and coordinates

Summary

- Introduction to Textures
  - Texture Mapping Methods
- 2D Texture Mapping
  - Texture Mapping in WebGL