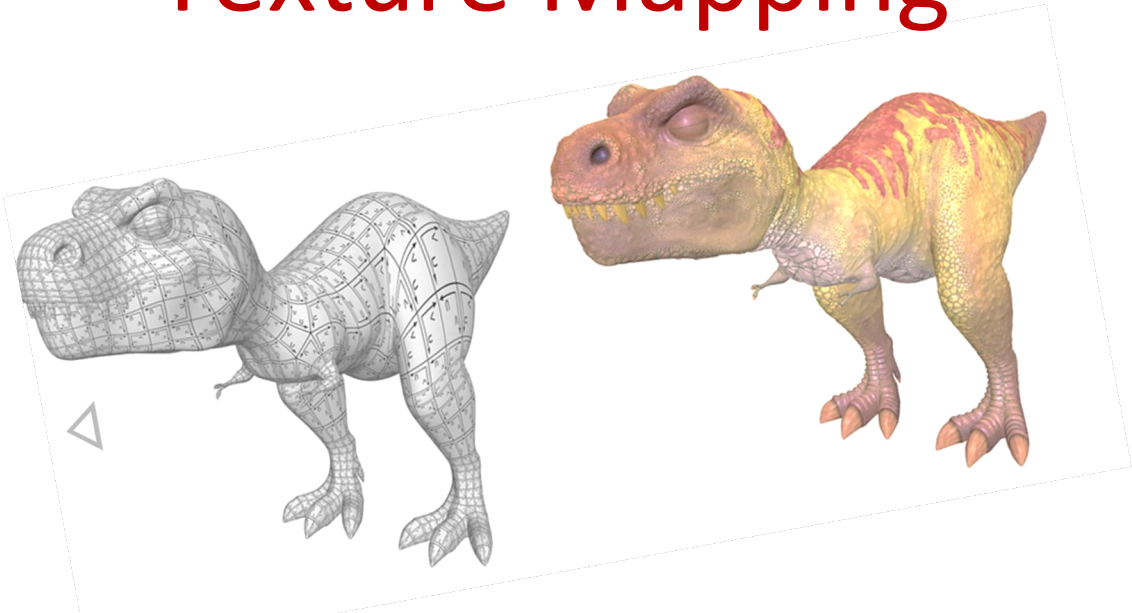
FACULTY OF ENGINEERING AND ARCHITECTURE

E016712: Computer Graphics

Texture Mapping



Lecturers: Aleksandra Pizurica and Danilo Babin

GHENT UNIVERSITY

Motivation

Although graphics cards can render millions of polygons per second, that number is insufficient for many phenomena

Overview

- Mapping methods
  - Texture mapping
  - Environment mapping
  - Bump mapping
- Basic strategies
  - Forward vs backward mapping
  - Point sampling vs area averaging

The material based on: E. Angel and D. Shreiner: Interactive Computer Graphics 6E © Addison-Wesley 2012

Example: modelling an orange

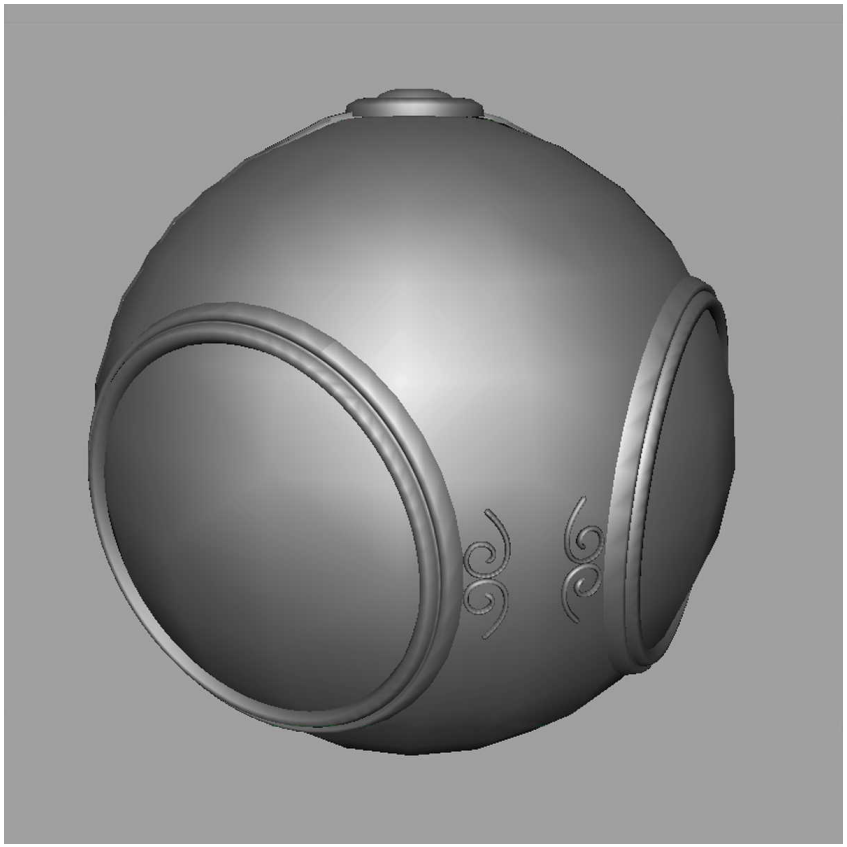
- Start with an orange-colored sphere
  - Too simple
- Replace sphere with a more complex shape
  - Does not capture surface characteristics (small dimples)
  - Takes too many polygons to model all the dimples
- Take a digital picture of a real orange, and "paste" it onto a simple geometric model
  - This process is known as texture mapping
- Still might not be sufficient because resulting surface will be smooth
  - Need to change local shape
  - Bump mapping

Three Types of Mapping Methods

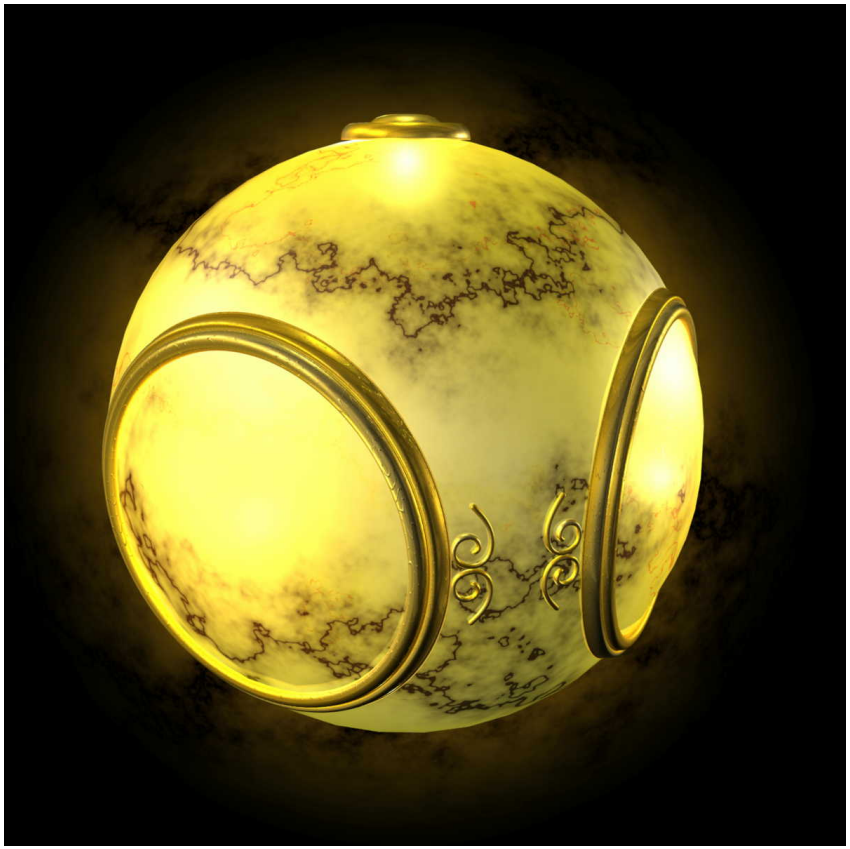
- Texture Mapping
  - Uses images to fill inside of polygons
- Environment (reflection mapping)
  - Uses a picture of the environment for texture maps
  - Allows simulation of highly specular surfaces
- Bump mapping
  - Emulates altering normal vectors during the rendering process

Texture Mapping

geometric model



texture mapped

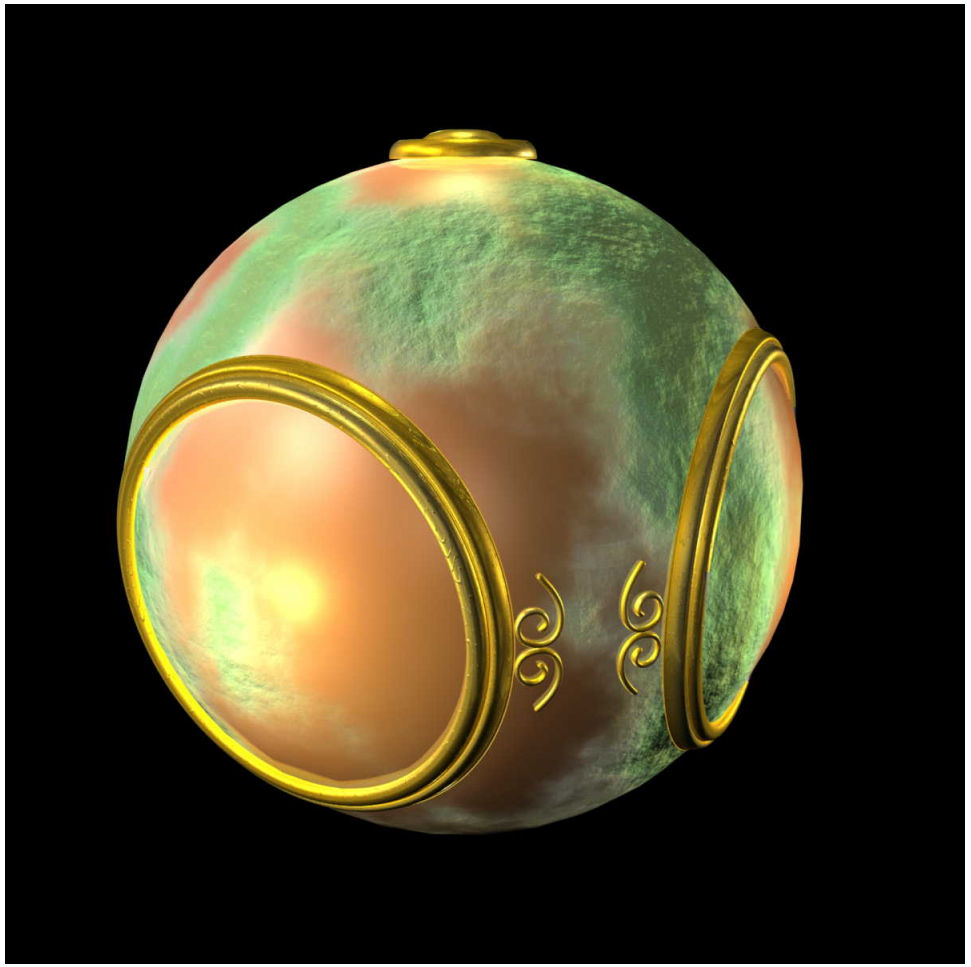


E. Angel and D. Shreiner: Interactive Computer Graphics 6E © Addison-Wesley 2012

Environment Mapping

E. Angel and D. Shreiner: Interactive Computer Graphics 6E © Addison-Wesley 2012

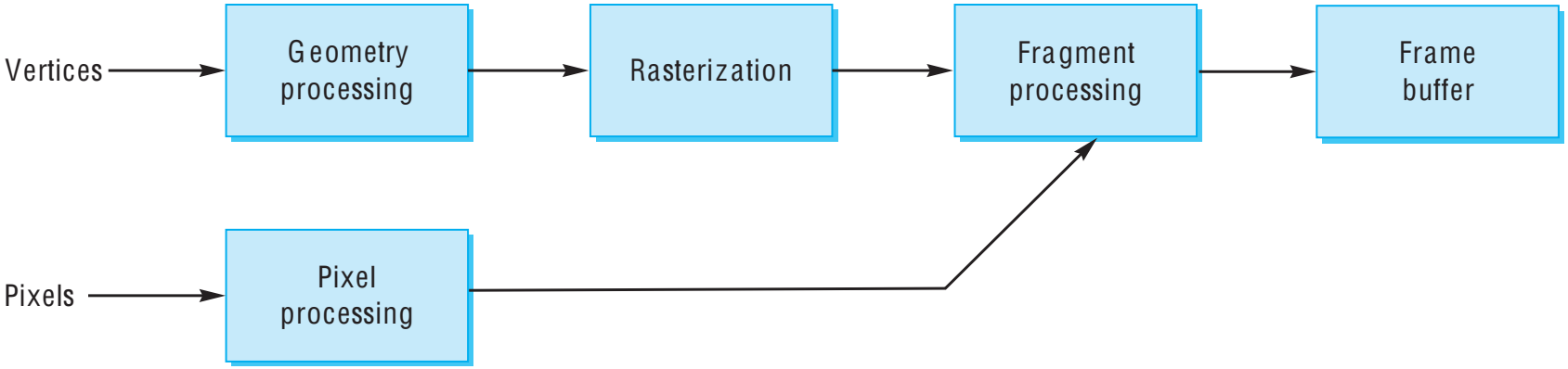
Bump Mapping



E. Angel and D. Shreiner: Interactive Computer Graphics 6E © Addison-Wesley 2012

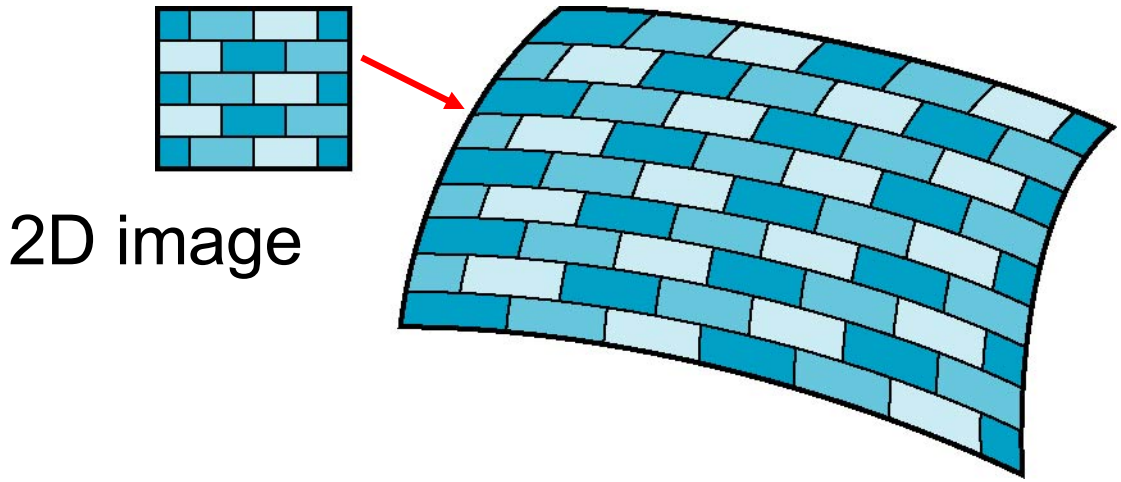
Where does mapping take place?

- Mapping techniques are implemented at the end of the rendering pipeline
  - Very efficient because few polygons make it past the clipper



Is it simple?

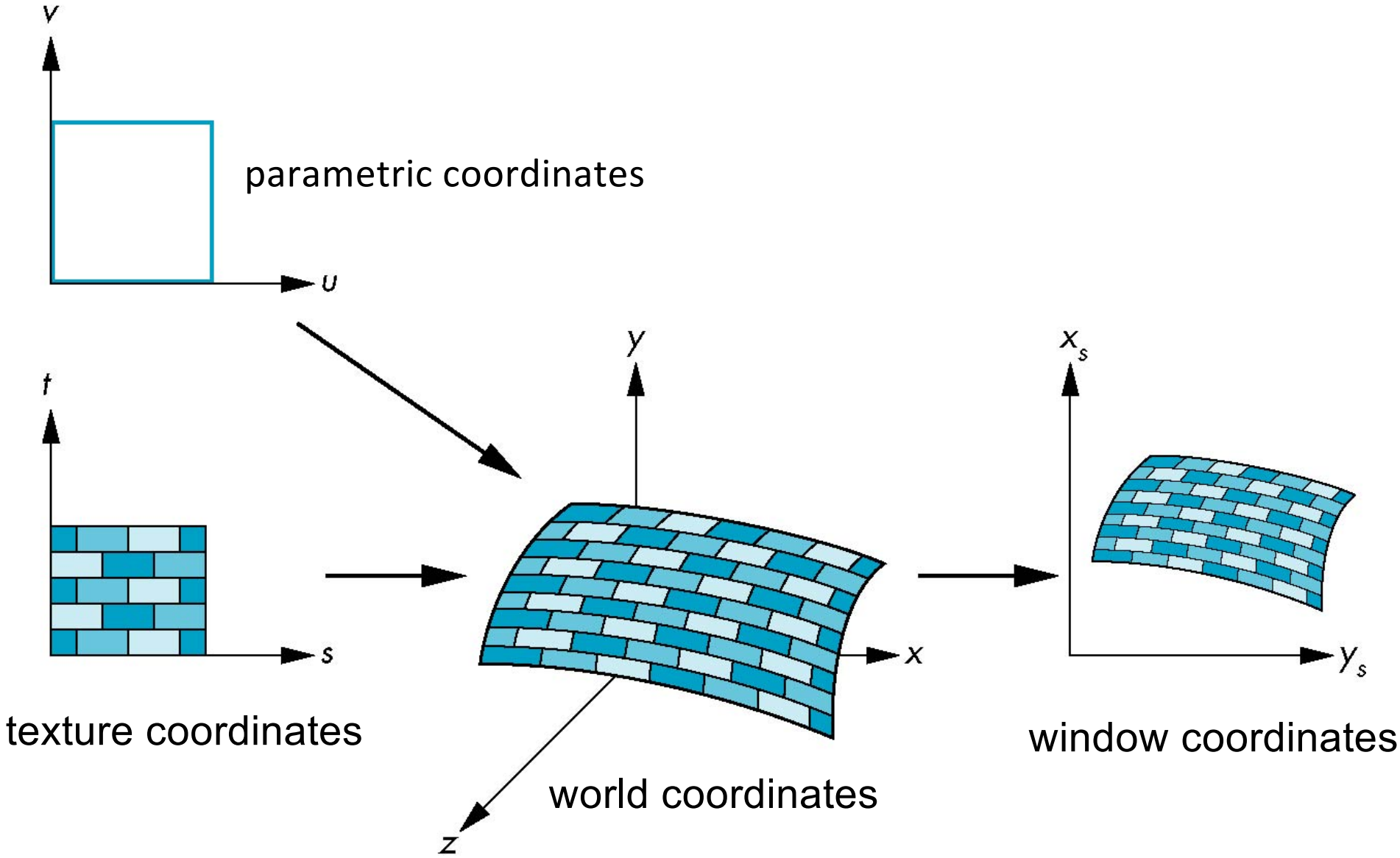
- Although the idea is simple − map an image to a surface − there are 3 or 4 coordinate systems involved



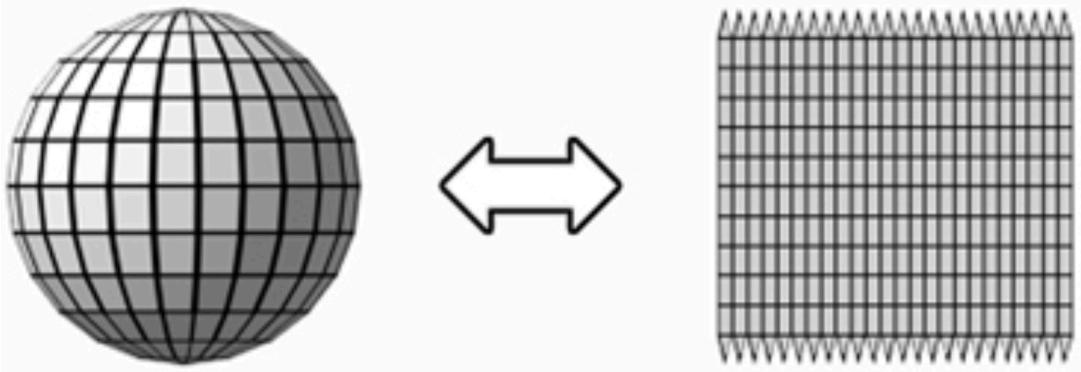
Coordinate Systems

- 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 Coordinates
  - Where the final image is really produced

Texture Mapping



Texture Mapping

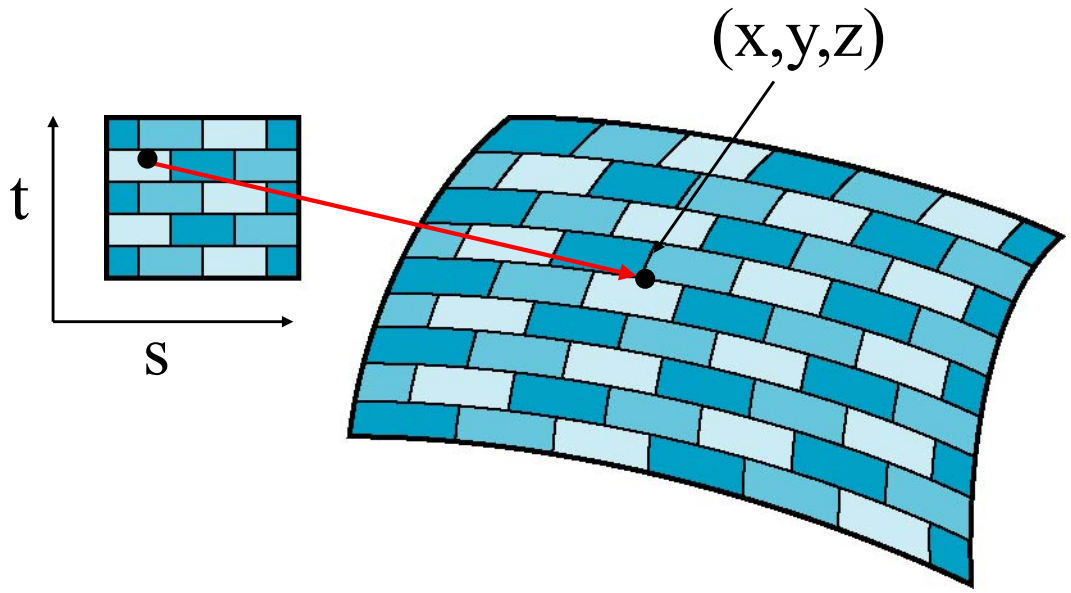


world coordinates parametric coordinates texture coordinates

Mapping Functions

- Basic problem is how to find the maps
- Consider mapping from texture coordinates to a point on a surface

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



- But we really want to go the other way!

Backward Mapping

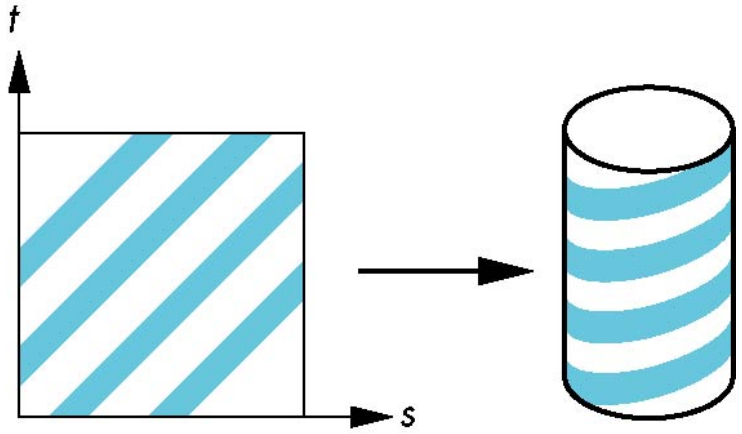
- We really want to go backwards
  - Given a pixel, we want to know which point on an object it corresponds to.
  - Given a point on an object, we want to know which point in the texture it corresponds to.
- Need a map of the form

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

- Such functions are difficult to find in general

Two-part mapping

- One solution to the mapping problem is to first map the texture to a simple intermediate surface
- Example: map to cylinder



Cylindrical Mapping

Suppose that texture coordinates (s,t) vary over the unit square $0 < s < 1$ and $0 < t < 1$, and that we use the surface of a cylinder with radius r and height h as our intermediate object.

The parametric equations:

$$x = r \cos(2\pi u)$$
$$y = r \sin(2\pi u)$$
$$z = v/h$$

map the rectangle in (u, v) space to cylinder of radius r and height h in world coordinates. Hence, we obtain the mapping from texture space with

$$s = u$$
$$t = v$$

Spherical Map

We can use a parametric sphere

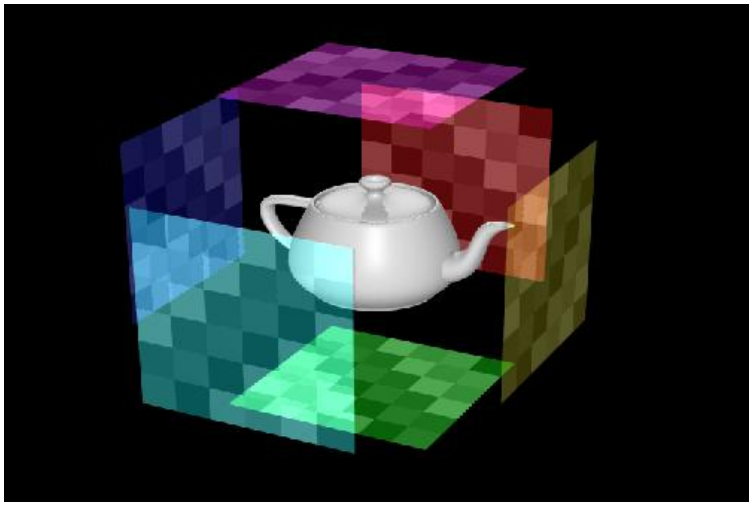
$$x = r \cos 2\pi u$$
$$y = r \sin 2\pi u \cos 2\pi v$$
$$z = r \sin 2\pi u \sin 2\pi v$$

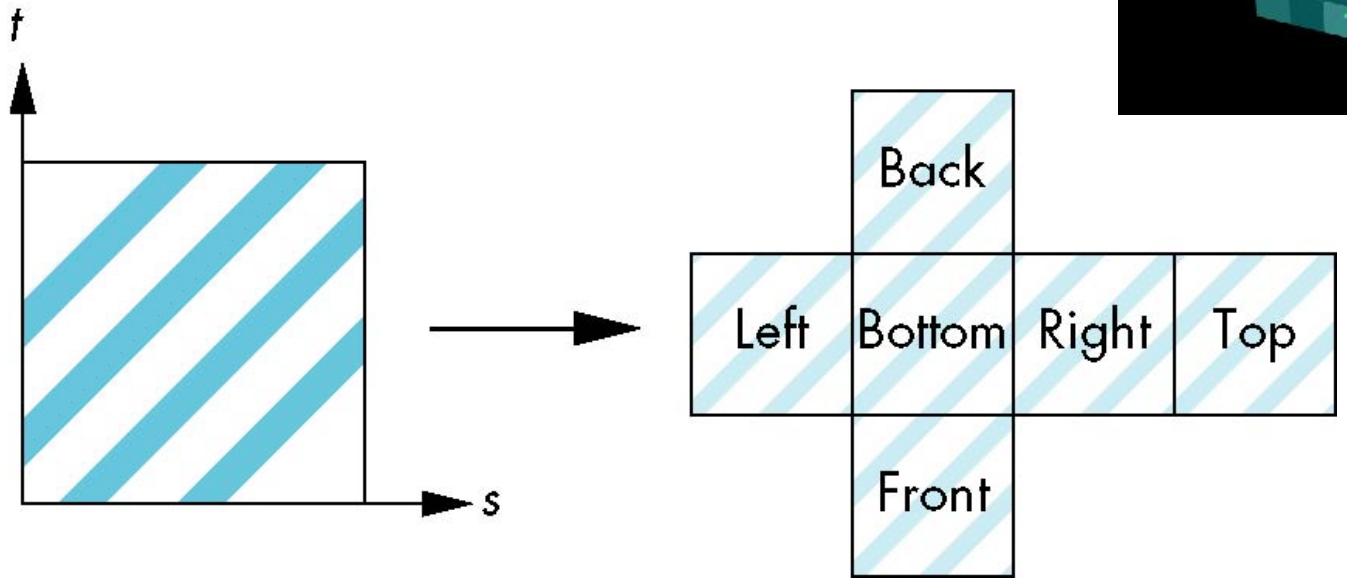
in a similar manner to the cylinder but have to decide where to put the distortion

Spheres are used in environmental maps

Box Mapping

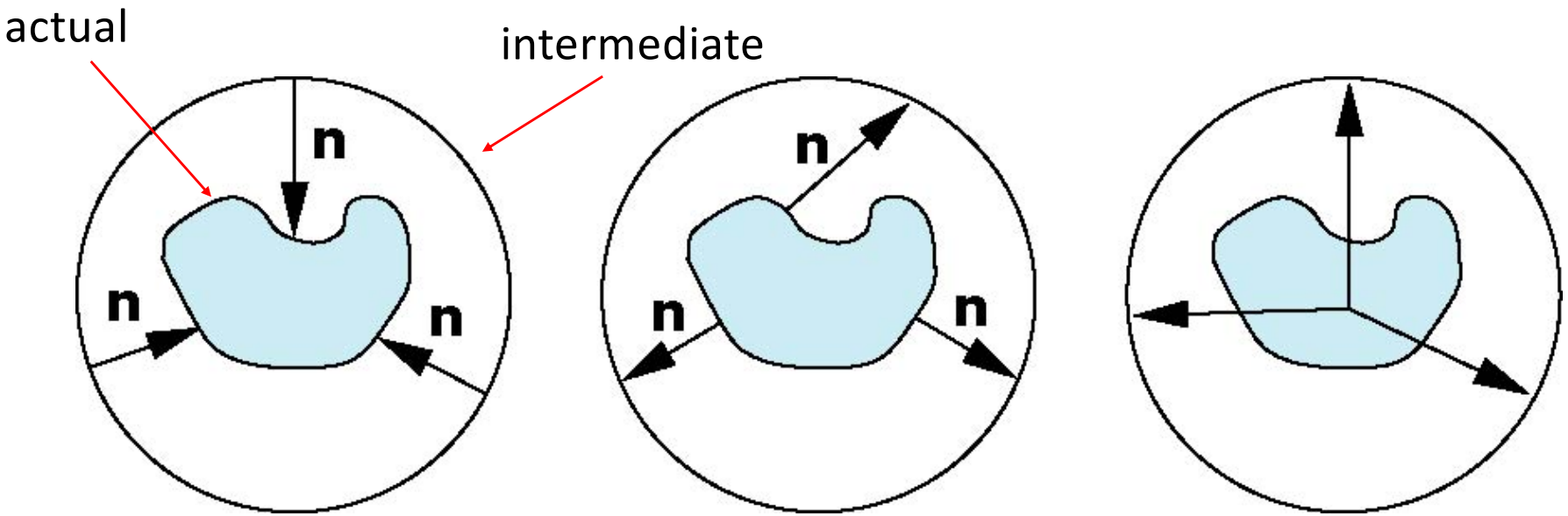
- Easy to use with simple orthographic projection
- Also used in environment maps





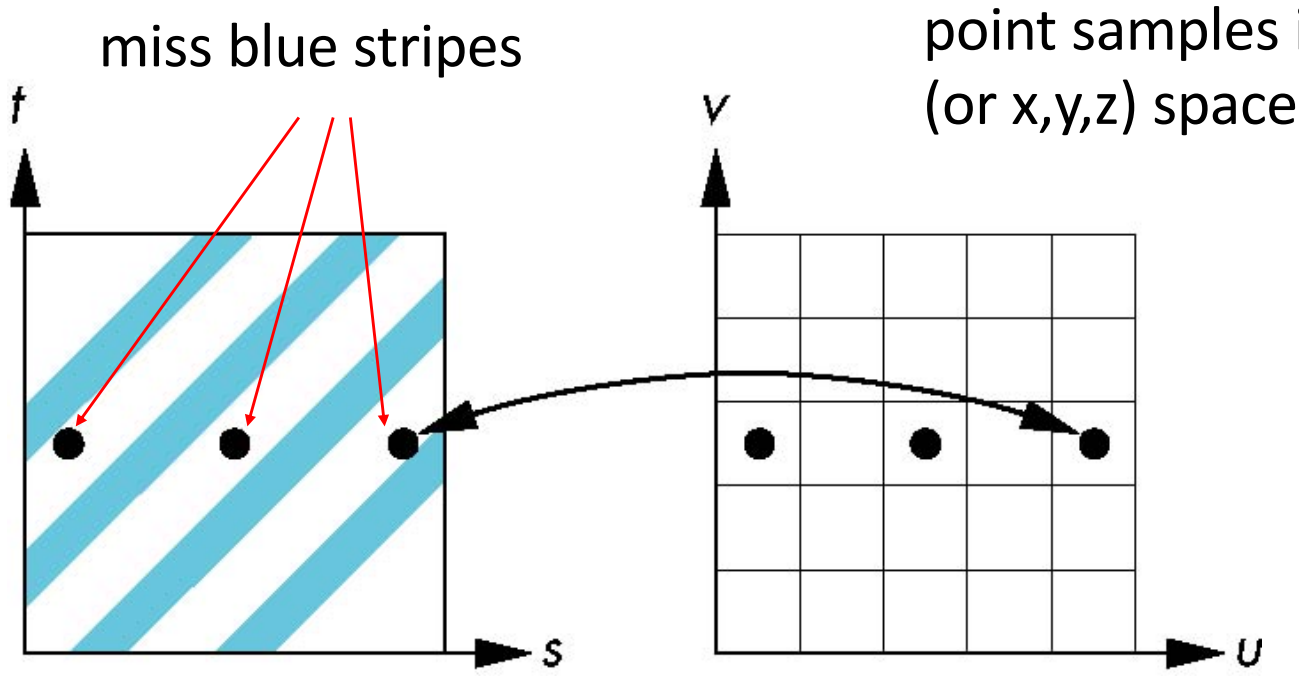
Second Mapping

- Map from intermediate object to actual object
  - Normals from intermediate to actual
  - Normals from actual to intermediate
  - Vectors from center of intermediate



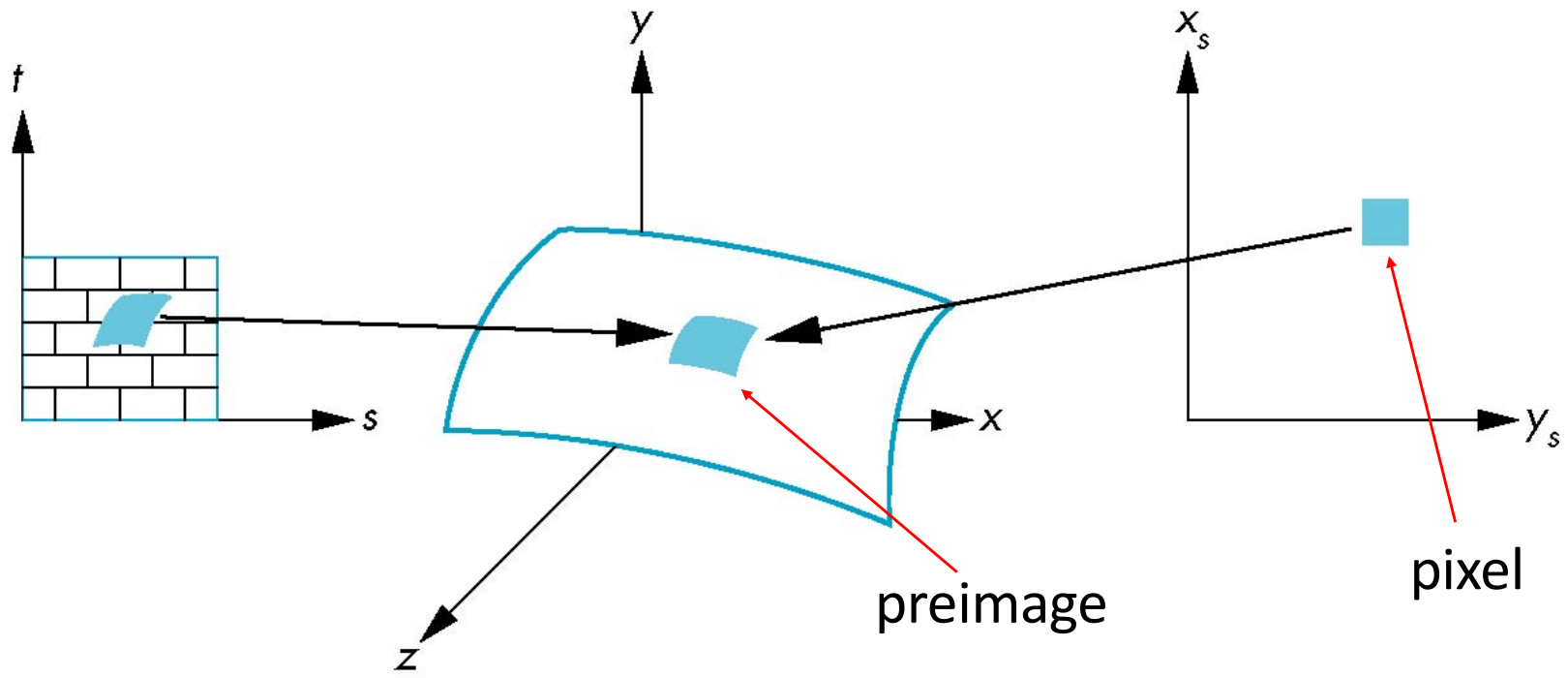
Aliasing

- Point sampling of the texture can lead to aliasing errors



Area Averaging

A better but slower option is to use *area averaging*



Note that *preimage* of pixel is curved

Texture Mapping in Practice: Basic Strategy

Three steps

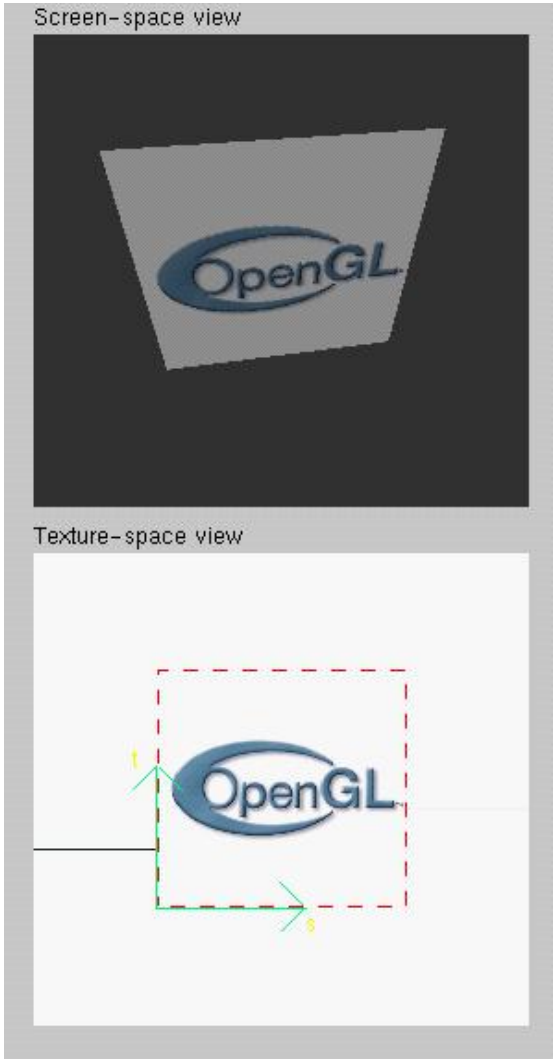
1. **specify the texture**
   - read or generate image
   - assign to texture
   - enable texturing

2. **assign texture coordinates to vertices**
   - Proper mapping function is left to application

3. **specify texture parameters**
   - wrapping, filtering

Texture Mapping Example

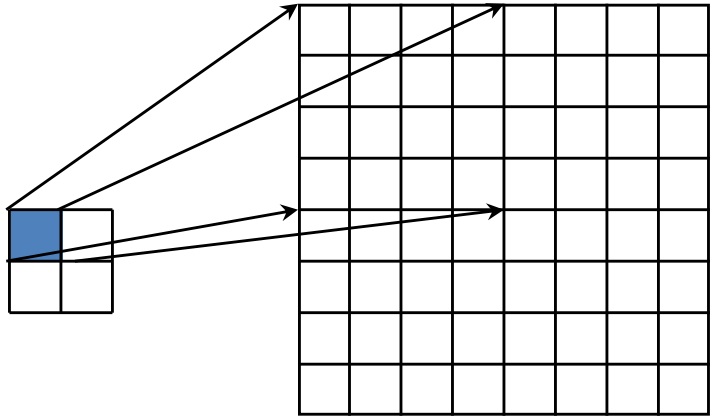
The texture (below) is a 256 x 256 image that has been mapped to a rectangular polygon which is viewed in perspective



Magnification and Minification

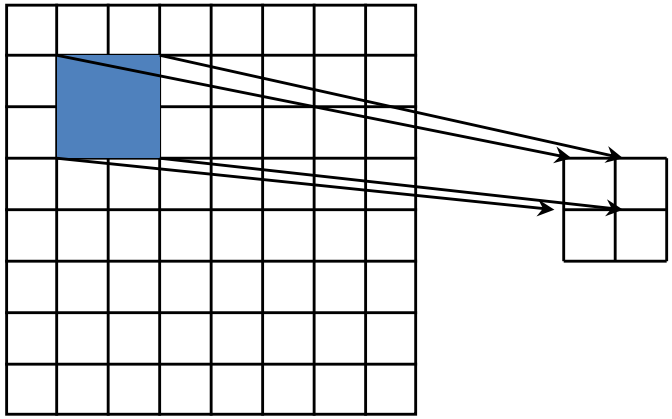
More than one texel can cover a pixel (*minification*) or more than one pixel can cover a texel (*magnification*)

Can use point sampling (nearest texel) or linear filtering (2 x 2 filter) to obtain texture values



Texture Polygon

Magnification



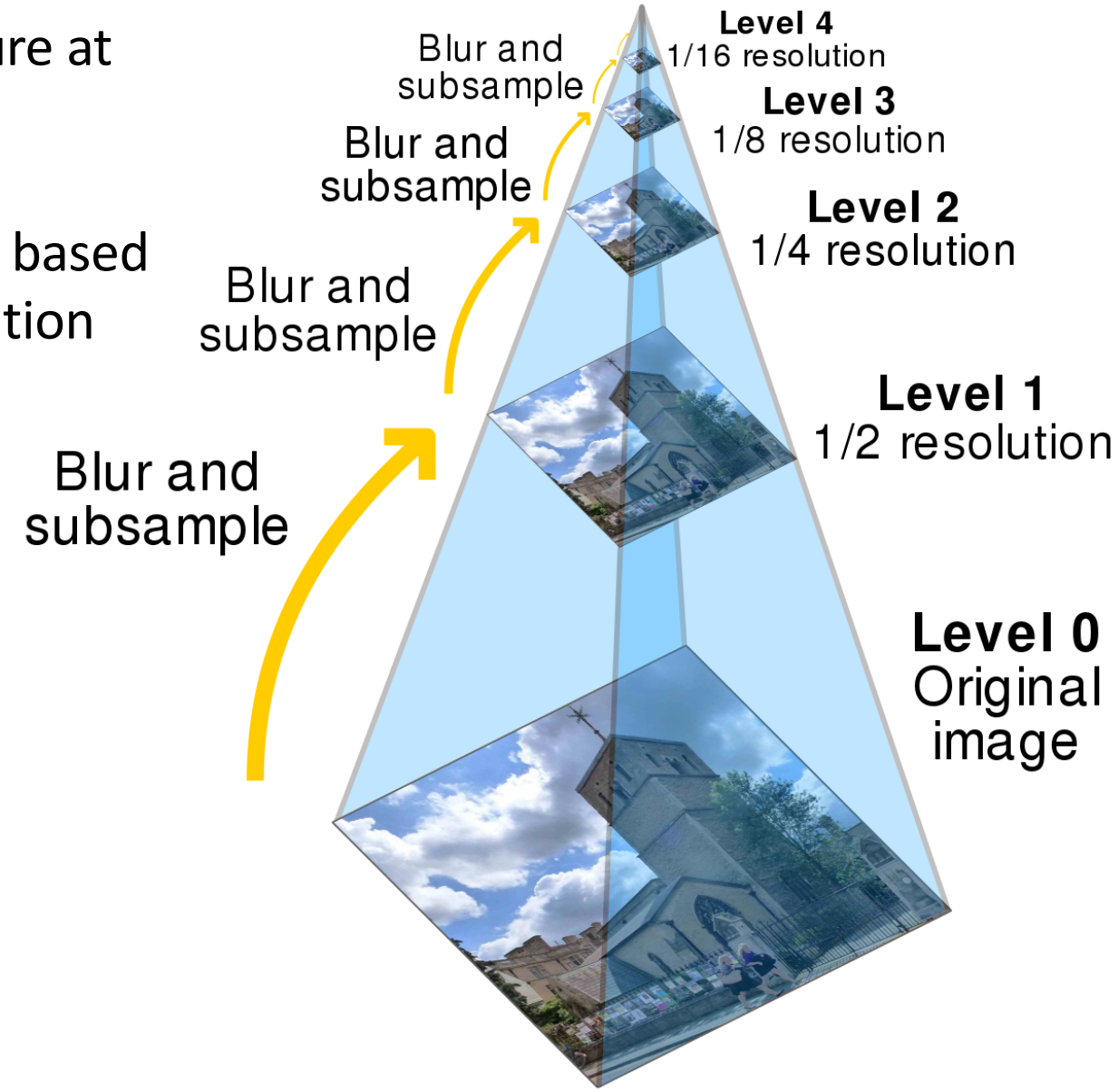
Texture Polygon

Minification

Magnification and Minification: Mipmaps

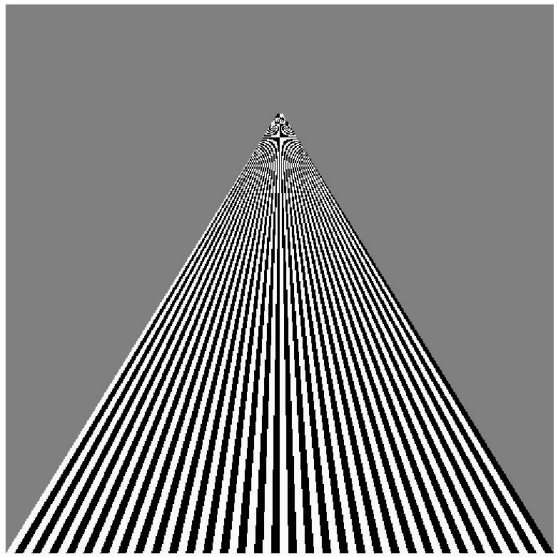
Produce images of a texture at different scales .

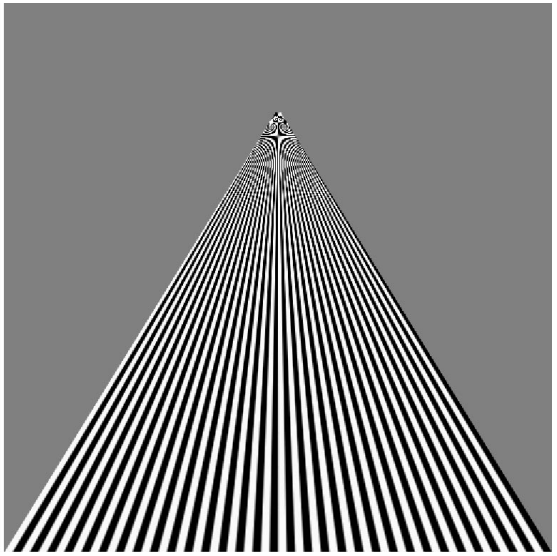
Different images are used based on the required magnification (or minification) level.



Example

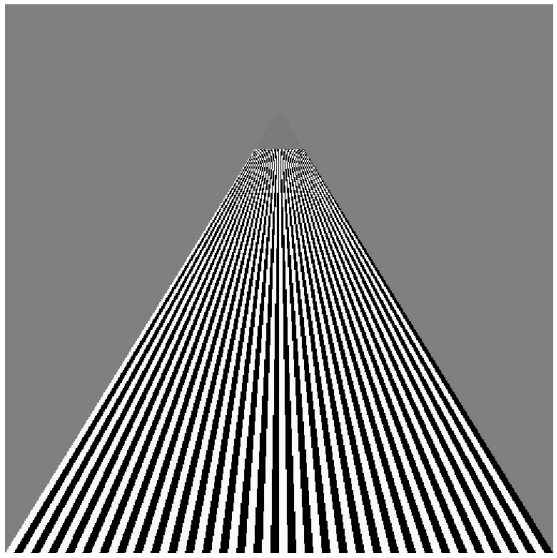
point sampling

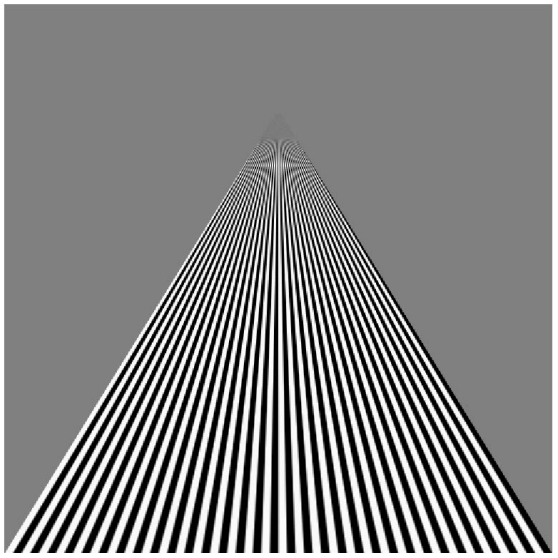




linear filtering

mipmapped point sampling

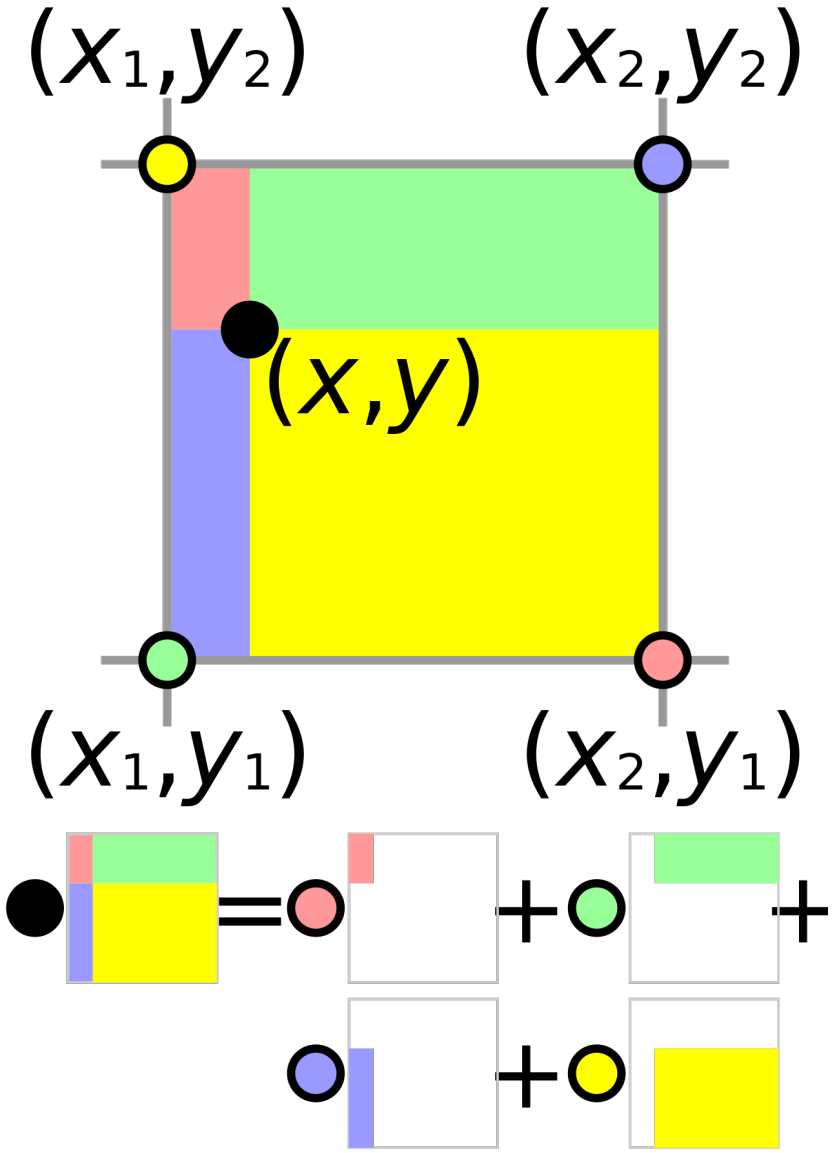




mipmapped linear filtering

Mipmaps: Bilinear filtering

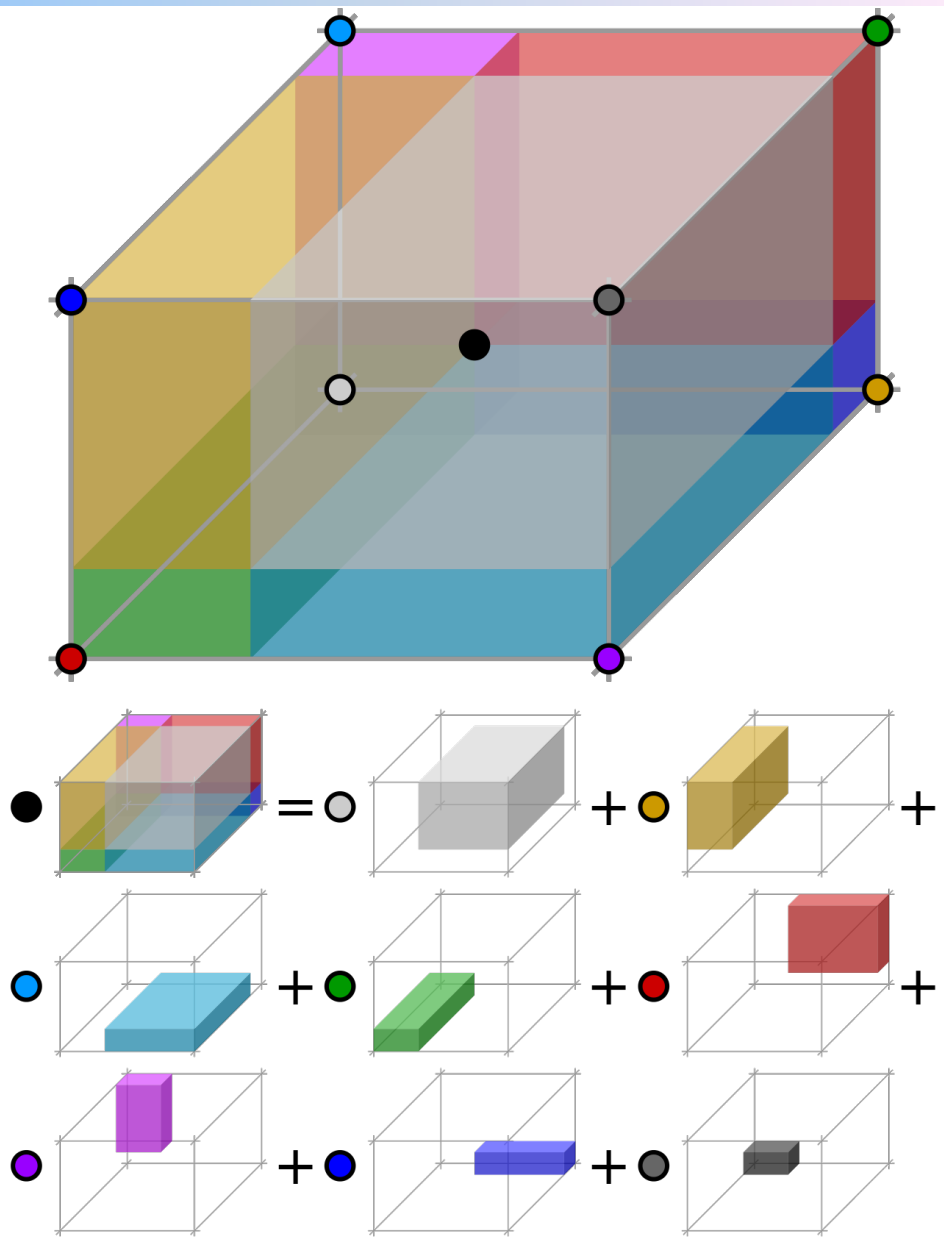
value_at_black_spot * total_area = sum (value_at _colored_spot * diagonal_area).



Source:
https://en.wikipedia.org/wiki/Bilinear_interpolation

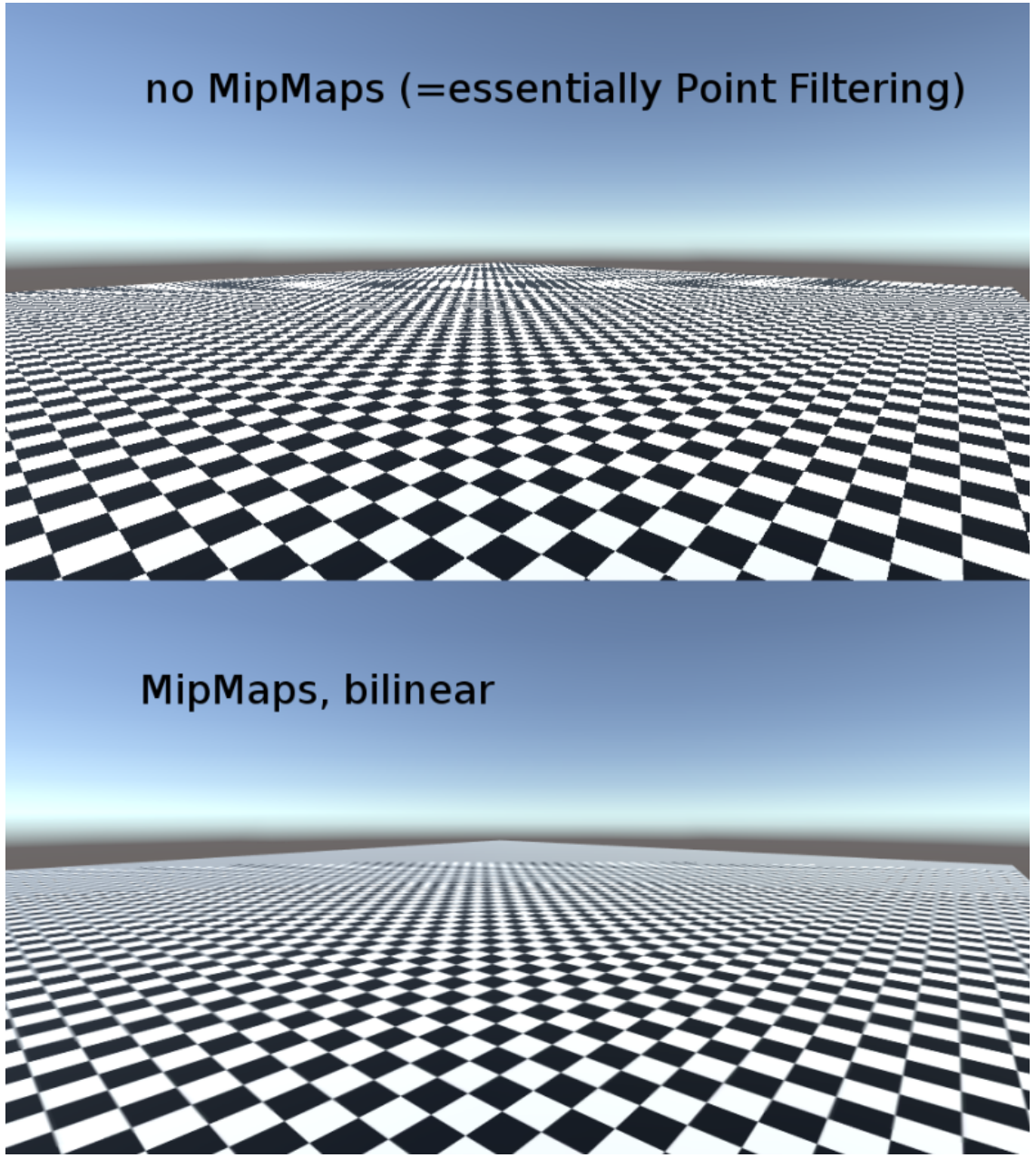
Mipmaps: Trilinear filtering

value_at_black_spot * total_volume = sum (value_at _colored_spot * diagonal_volume).



Source: https://en.wikipedia.org/wiki/Trilinear_interpolation

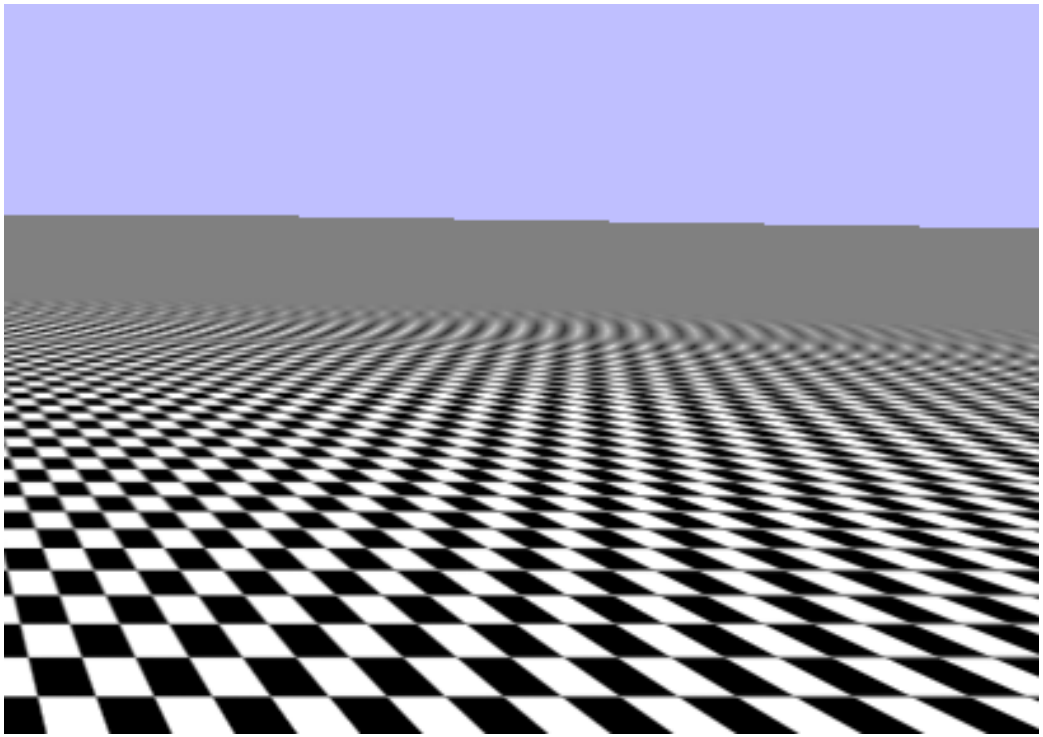
Example

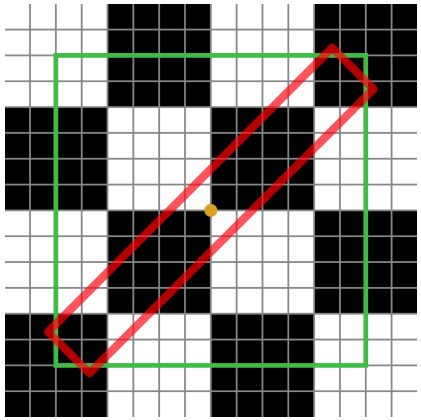


Source:
https://answers.unity.com/questions/310352/texture-mipmap-distance.html

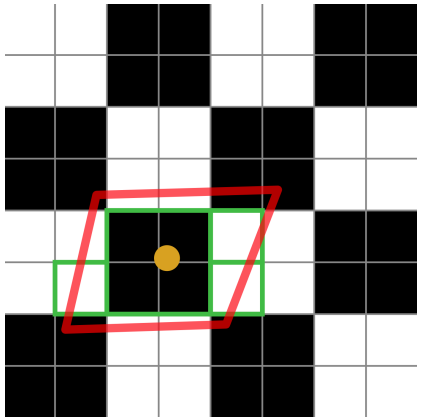
Mipmaps: Anisotropic filtering

Problem: mipmaps are filtered in both directions equally, results in blurred textures (for distant patterns).
Solution: sample using an anisotropic filter box.





Isotropic (regular)



Anisotropic

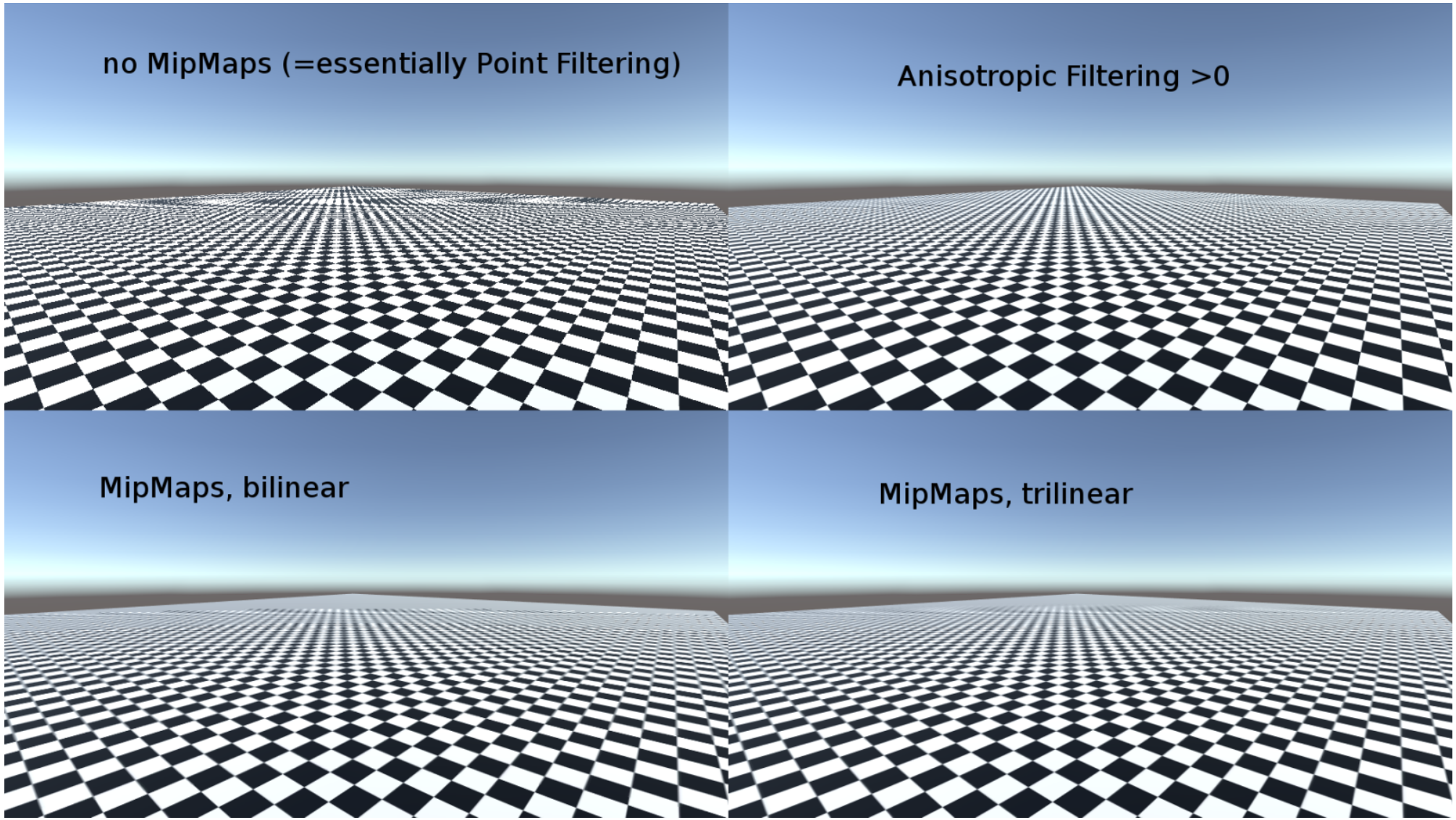
Sources:
https://en.wikipedia.org/wiki/Anisotropic_filtering
https://paroj.github.io/gltut/Texturing/Tut15%20Anisotropy.html

Anisotropic Mipmaps

In addition to isotropic sampling (e.g. 256x256, 128x128, etc.), textures are sampled anisotropically (e.g. 256x128, 32x128, etc.)

Source: https://en.wikipedia.org/wiki/Anisotropic_filtering

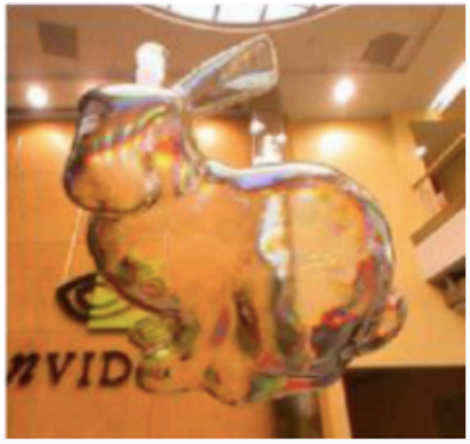
Example



Source: https://answers.unity.com/questions/310352/texture-mipmap-distance.html

Environment mapping: Background

- Many objects are glossy or transparent
- Glossy objects reflect the external world
- The world is refracted through transparent objects
- Important to make the scene appear realistic



Courtesy of Taku Komura www.inf.ed.ac.uk/teaching/courses/cg

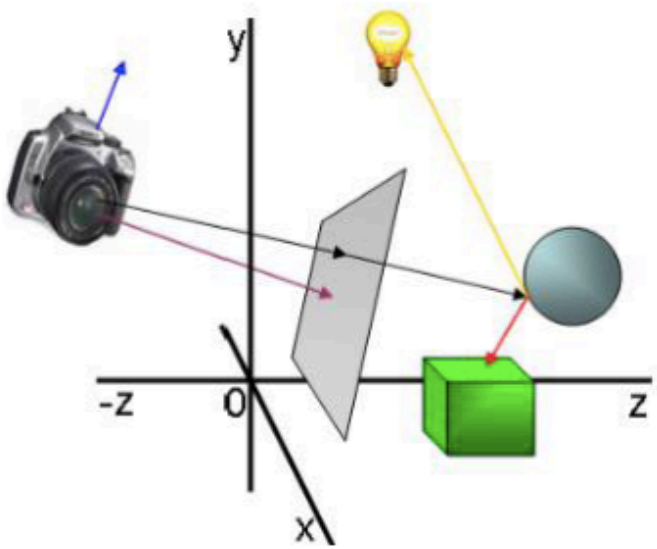
Example

Courtesy of Taku Komura www.inf.ed.ac.uk/teaching/courses/cg

Environment mapping: Background

Precisely simulating such phenomena is computationally costly

- Requires ray tracing, which can be expensive
- Tracking the rays, finding out where they collide, and doing another lighting computation



Courtesy of Taku Komura www.inf.ed.ac.uk/teaching/courses/cg

Environment Mapping

Also called reflection mapping.

Approximating the appearance of a reflective surface by means of a precomputed texture image.

- A quick way to simulate the effects of reflecting the surrounding world on the surface of a glossy object.

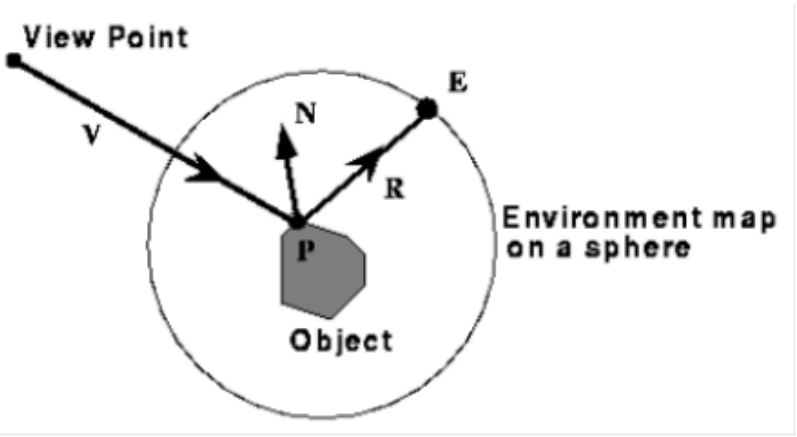
The texture is used to store the image of the distant environment surrounding the rendered object.

Two approaches:

- Sphere mapping (older)
- Cube mapping

Environment mapping

- Simple yet powerful method to generate reflections
- Simulate reflections by using the reflection vector to index a texture map at "infinity".

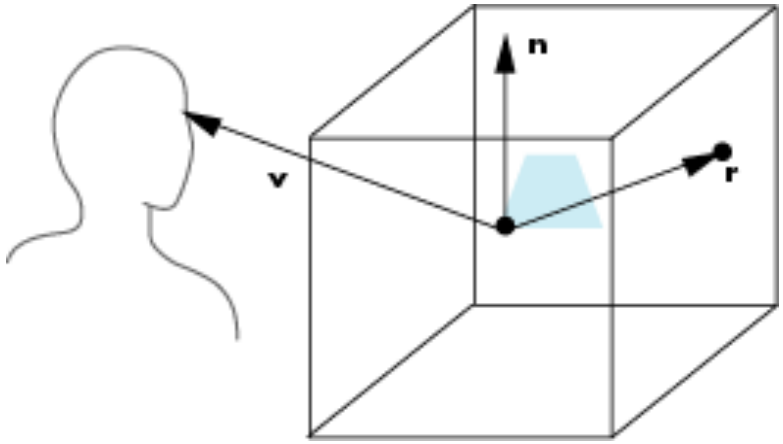


The original environment map was a sphere [by Jim Blinn '76]

Courtesy of Taku Komura www.inf.ed.ac.uk/teaching/courses/cg

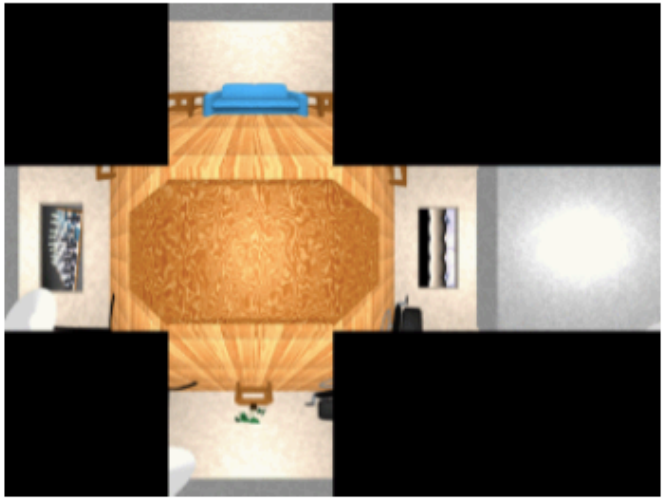
Cube Mapping

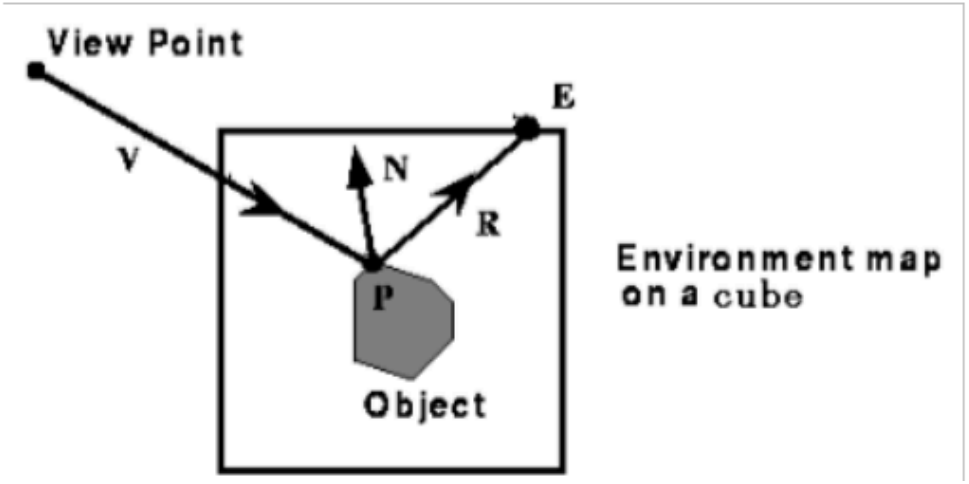
Use reflection vector to locate texture in cube map



Cube Mapping

- The map resides on the surfaces of a cube around the object
- Align the faces of the cube with the coordinate axes



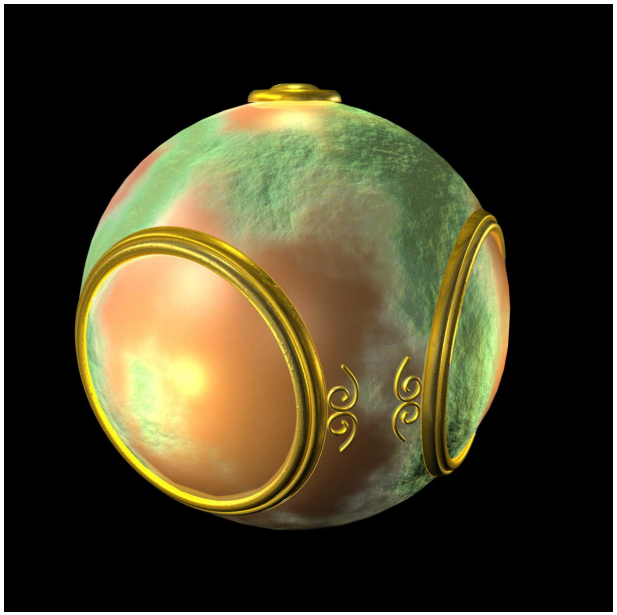


Environment map on a cube

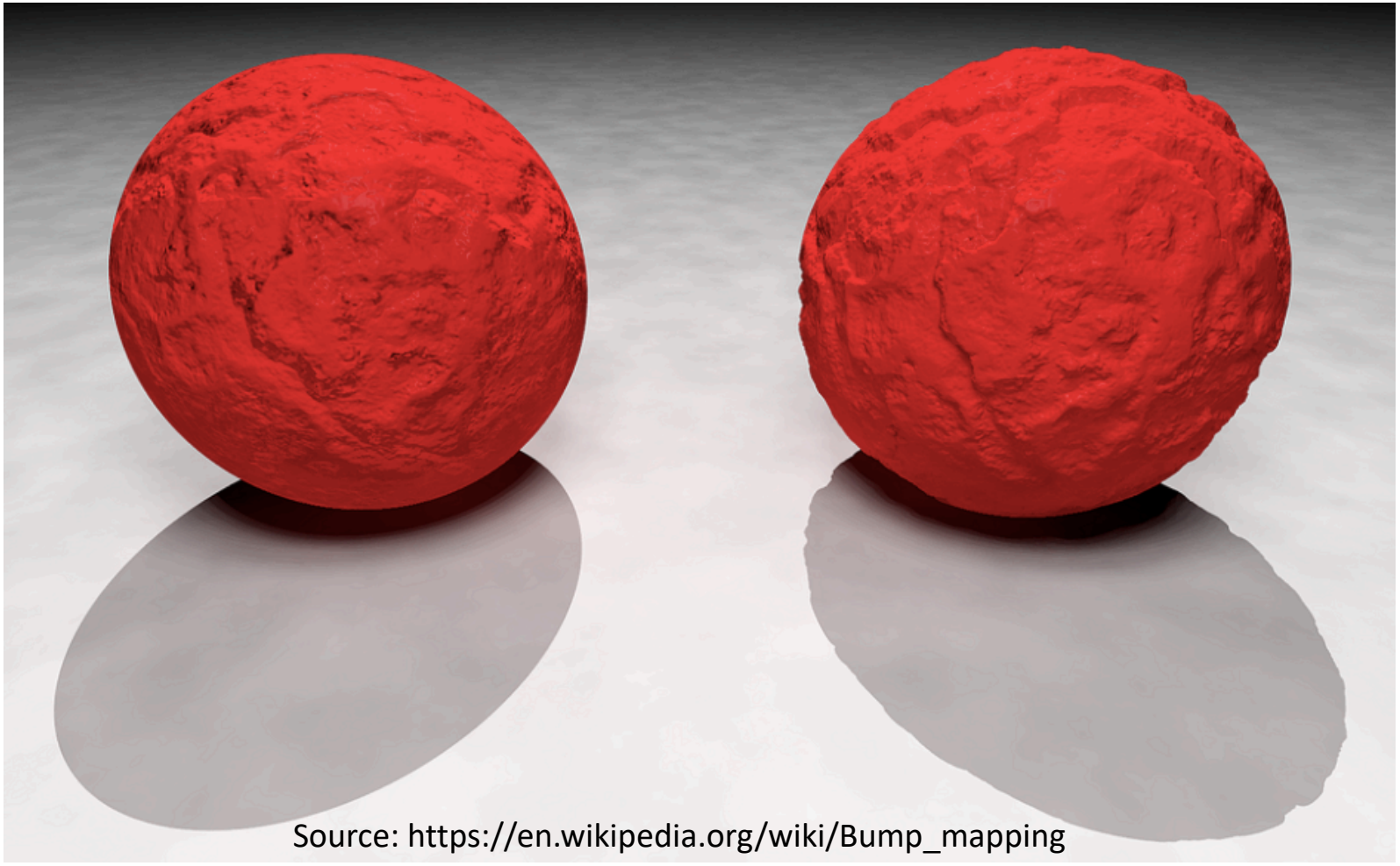
Courtesy of Taku Komura www.inf.ed.ac.uk/teaching/courses/cg

Bump Mapping

- Perturb normal for each fragment
- Store perturbation as textures



Limitations of bump mapping



Source: https://en.wikipedia.org/wiki/Bump_mapping

Bump mapping (left) does not modify the shape of the object. The image on the right shows the effect of modifying the surface with the same function.