Visual Appearance (2): Surface Mapping

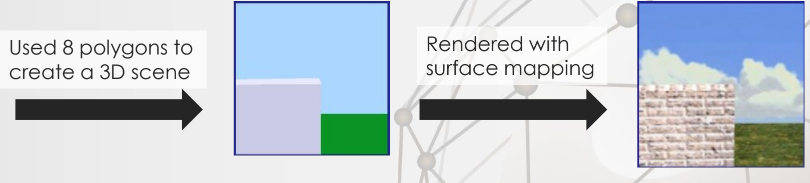
Lesson objectives

By the end of the module, you should be able to:

- Understand texture mapping
- Perform computation using parametric texture mapping
- Understand bump mapping
- Understand displacement mapping

1. Introduction

- The appearance of real life surfaces may not be just colored but also have textures, patterns, displacements, or bumps.
- To enrich the appearance, add surface details and achieve the impression of natural colors, textures, or other visual effects, various surface mapping techniques have been used in computer graphics, which can efficiently and effectively improve the visual realism of the generated images.

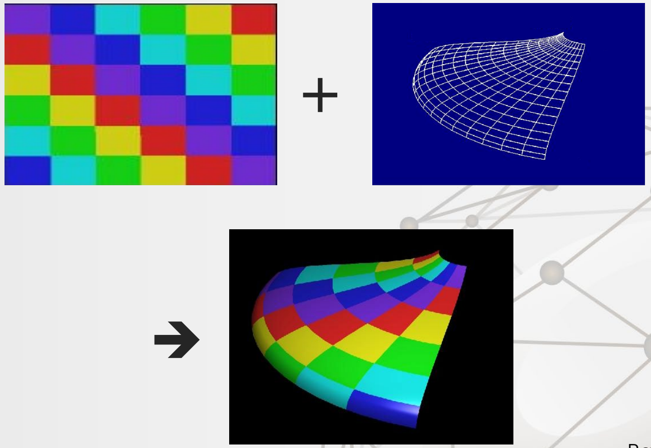


Used 8 polygons to create a 3D scene

Rendered with surface mapping

2. Texture mapping

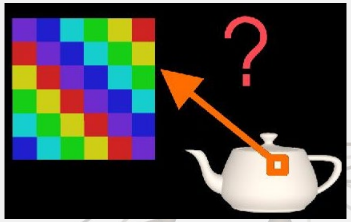
What is texture mapping?



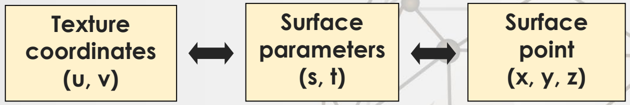
2.1 Parametric texture mapping

- **Task:** Given the following input:
  - A texture defined with a rectangular color pattern
  - A surface defined by parametric equation

  How can we find color for each point (x, y, z)?



- The basic idea of parametric texture mapping is to establish the mappings below:



Texture coordinates (u, v) ⟷ **Surface parameters (s, t)** ⟷ **Surface point (x, y, z)**

Main steps of parametric texture mapping

Step 1: Parameterize texture with (u, v) coordinates.

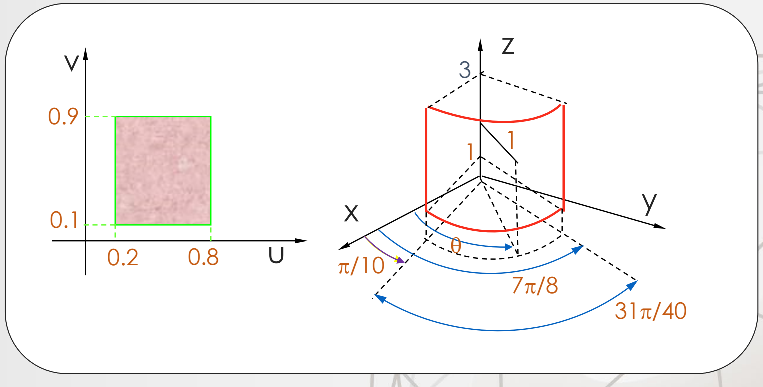
Step 2: Parameterize the surface with (s, t) coordinates.

Step 3: Define a mapping between (u, v) and (s, t).

Step 4: When shading a surface at point (x, y, z), find the corresponding (s, t) and use the mapping function to find (u, v), from which the appropriate texel in the texture is accessed and used to affect the final color.

Example 1

Q: Propose a method that maps the texture shown on the left of the figure below onto the portion of the cylinder bounded by four red lines.

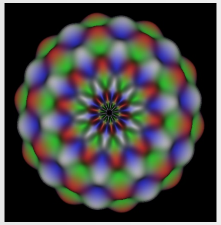


2.2 Something to think about

- How to texture an object that is not defined by parametric representation or has very complicated shape and representation?
- **Approach 1:**
  Parameterize the object, and then use the parametric texture mapping.
- **Approach 2:**
  Find the correspondence between the object and some simple shape (called the map shape) such as cylinder, sphere, or box first, and then find the mapping between the texture and the simple shape.

2.3 Function-based extension

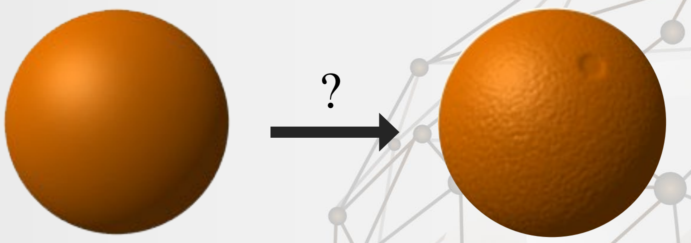
- Texture can be viewed as a function defined on discrete grids of a rectangular domain.
- If texture can be defined or simulated by a continuous function, this may make the representation more compact.
  - In FVRML, this can be implemented in diffuseColor using parametric representation (ie., r = r(u,v,w,t); g = g(u,v,w,t); b = b(u,v,w,t)). For example,



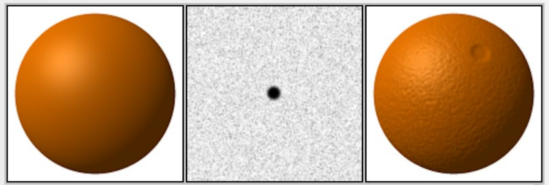
```
appearance FAppearance {
        material FMaterial {
              diffuseColor " r=0.3*sin(10*u*pi)*sin(10*v*pi)+0.5;
                                   g=0.3*sin(10*v*pi)+0.5;
                                   b=0.3*sin(10*u*pi)+0.5;"
         }
}
```

3. Bump mapping

- Texture mapping adds color detail to the object, but the object still looks **smooth**.
- **Problem:** How do you make the object look "rough"?



An effective solution: Bump mapping



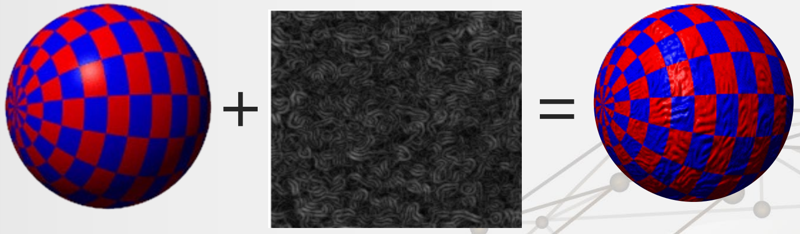
Smooth sphere + Bump map ➔ Mottled sphere

- Bump mapping (invented by James Blinn) uses texture (**bump map**) to perturb the normal at each point on the surface during the rendering stage, which results in the change of appearance.
  - The surface geometry is not modified.
  - Only the surface normal is modified as if the surface had been displaced.

Seymour, M. (2012, July 24). Founders Series: industry legend Jim Blinn. Retrieved August 25, 2017, from https://www.fxguide.com/featured/founders-series-industry-legend-jim-blinn/

Bump mapping example

Bump mapping example:



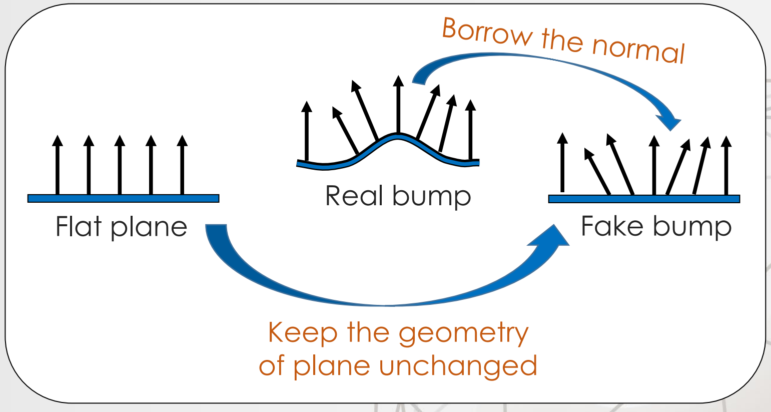
Sphere with color texture

Bump map

Sphere with color texture & bump map

Basic idea

- If we use “fake” normals, we can trick the renderer into thinking there are bumps on the surface:



Borrow the normal

Flat plane

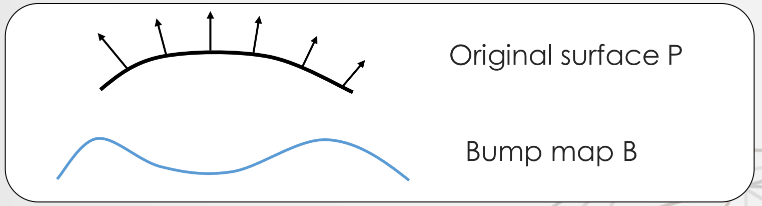
Real bump

Fake bump

Keep the geometry of plane unchanged

1D illustration of bump mapping

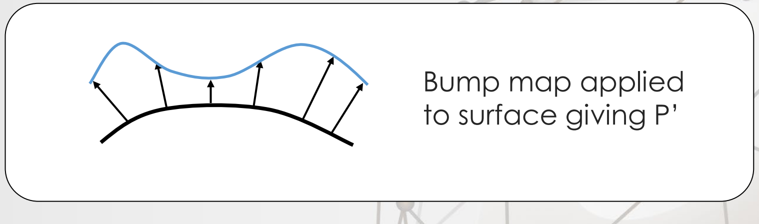
- Inputs:



Original surface P

Bump map B

- Combine the original surface with bump map:



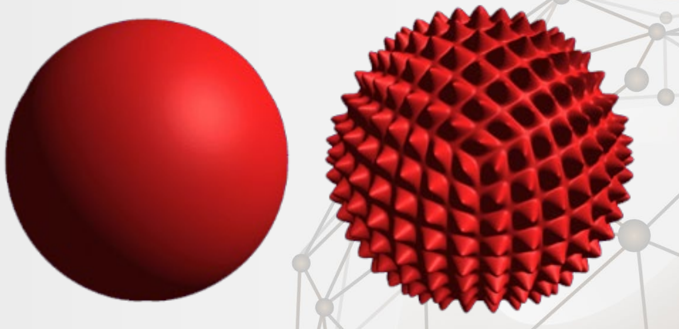
Bump map applied to surface giving P’

Bump mapping algorithm

- Bump mapping performs a calculation for each visible point on the object’s surface before lighting. This involves several steps:
  - Look up the values in the bump map that correspond to the position on the surface
  - Compute two partial derivatives of the bump map
  - Use the two partial derivatives to perturb the true (“geometric”) surface normal
  - Calculate the intensity of the surface using, for example, Phong illumination model, with the perturbed normal

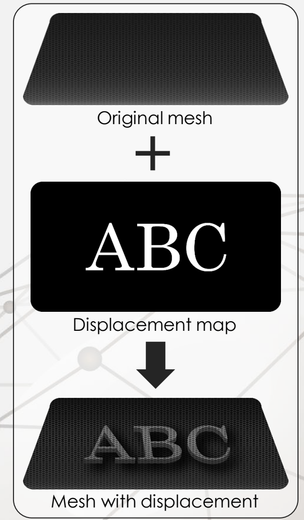
4. Displacement mapping

- Texture can also be used to displace each point on the surface. This is displacement mapping.
- In contrast to texture mapping and bump mapping, the surface is actually modified in displacement mapping.



4.1 Conventional displacement mapping

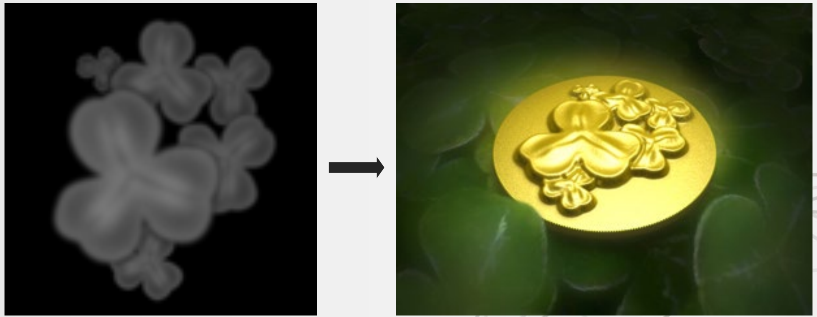
- Displacement mapping with height map:
  - Inputs: Original surface, height map (displacement map).
  - Step 1: Find the correspondence between the surface and the map.
  - Step 2: Displace the position of each point on the surface along its surface normal by amount according to the value in the corresponding position in the map.
  - Output: Displaced surface.



Features of displacement mapping

- The actual geometry is altered in order to allow for a more “bumpy” look
- Gives a more sense of depth and detail, thus a more realistic look. In particular:
  - Realistic silhouettes
  - Permitting self-shadowing or self-occlusion
- More costly than the other techniques

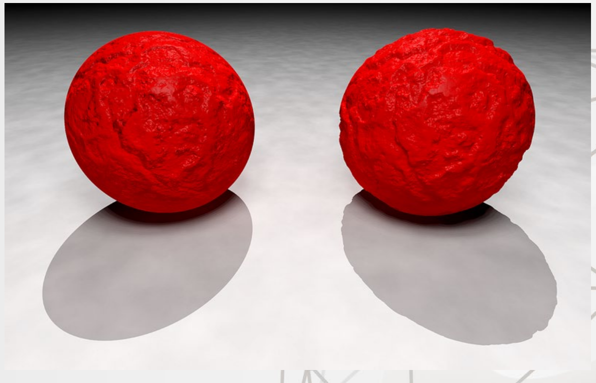
Displacement mapping example



The displacement map: Higher intensities indicate greater displacement

Question

In the figure below, which one is created by bump mapping and which one is created by displacement mapping? Why?



4.2 Function-based extension

- The displacement map actually defines geometric texture. The geometric texture can also be specified by explicit or parametric functions.
- There are four situations:
  - Parametric texture + parametric surface
  - Parametric texture + implicit surface
  - Explicit texture + parametric surface
  - Explicit texture + implicit surface

Parametric functions for geometric texture

- Geometric texture can be defined by parametric functions:

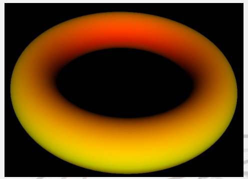
$$\begin{cases} x = x(u, v, w) \\ y = y(u, v, w) \\ z = z(u, v, w) \end{cases}$$

- These functions are used to displace the original surface. In general, the value range of the functions is very small compared to the size of the object.

Parametric texture + parametric surface

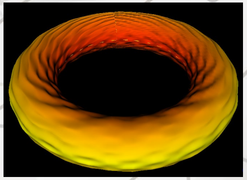
- A torus is defined by parametric equations:

$$\begin{aligned} x &= \cos(\pi v)(4+\cos(\pi u)) \\ y &= \sin(\pi v)(4+\cos(\pi u)) \quad -1 \le u, v \le 1 \\ z &= \sin(\pi u) \end{aligned}$$



- We can add the following geometric texture to the torus:

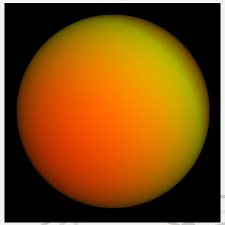
$$\begin{aligned} x &= 0.03\sin(12\pi u)\sin(120\pi v) \\ y &= 0.03\sin(15\pi v) + 0.03\sin(15\pi u) \\ z &= 0.03\sin(30\pi uv) \end{aligned}$$



Parametric texture + implicit surface

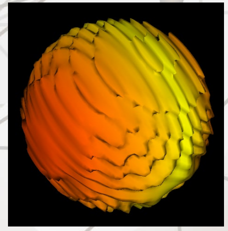
- A sphere is defined implicitly by:

$$6 - x^2 - y^2 - z^2 = 0$$



- We can add the following geometric texture to the sphere:

$$\begin{aligned} x &= 0.03\sin(12\pi u)\sin(120\pi v) \\ y &= 0.03\sin(15\pi v) + 0.03\sin(15\pi u) \\ z &= 0.03\sin(30\pi uv) \end{aligned}$$



Single function for geometric texture

- Geometric texture can be defined by a single function:

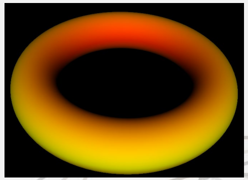
$$f(x, y, z)$$

- The value of this function is used to displace the original surface along its local normal. The value range of the function is generally **small** compared to the size of the object.

Explicit texture + parametric surface

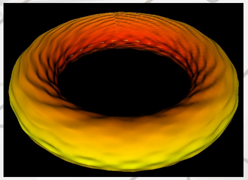
- A torus is defined by parametric equations:

$$\begin{aligned} x &= \cos(\pi v)(4+\cos(\pi u)) \\ y &= \sin(\pi v)(4+\cos(\pi u)) \quad -1 \le u, v \le 1 \\ z &= \sin(\pi u)) \end{aligned}$$



- We can add the following (explicit) geometric texture to the torus:

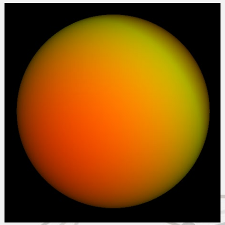
$$\begin{aligned} f(x, y, z) = 0.1(&\sin(2\pi x)\sin(2\pi y) \\ &+ \sin(2\pi x)\sin(2\pi z)); \end{aligned}$$



Explicit texture + implicit surface

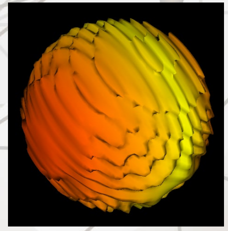
- A sphere is defined implicitly by:

$$6 - x^2 - y^2 - z^2 = 0$$



- We can add the following explicit (geometric) texture to the sphere:

$$\begin{aligned} f(x, y, z) = 0.1(&\sin(2\pi x)\sin(2\pi y) \\ &+ \sin(2\pi x)\sin(2\pi z)); \end{aligned}$$



5. Summary

- Texture mapping:
  - Used to change the color
- Bump mapping:
  - Used to change the normal of the surface
- Displacement mapping:
  - Uses geometric texture to change the surface