Texture Mapping

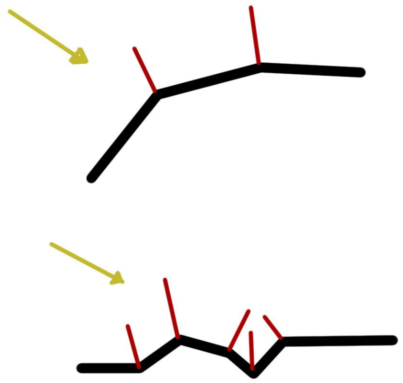
Team Mayans

Bump Mapping

Introduction

- Simulates bumps or wrinkles without needing to alter the model
- Shadow of objects won't change since the actual geometry of the object stays the same
- When a bump map is applied, the renderer will calculate the normal of the vertices and project the light rays

Theory

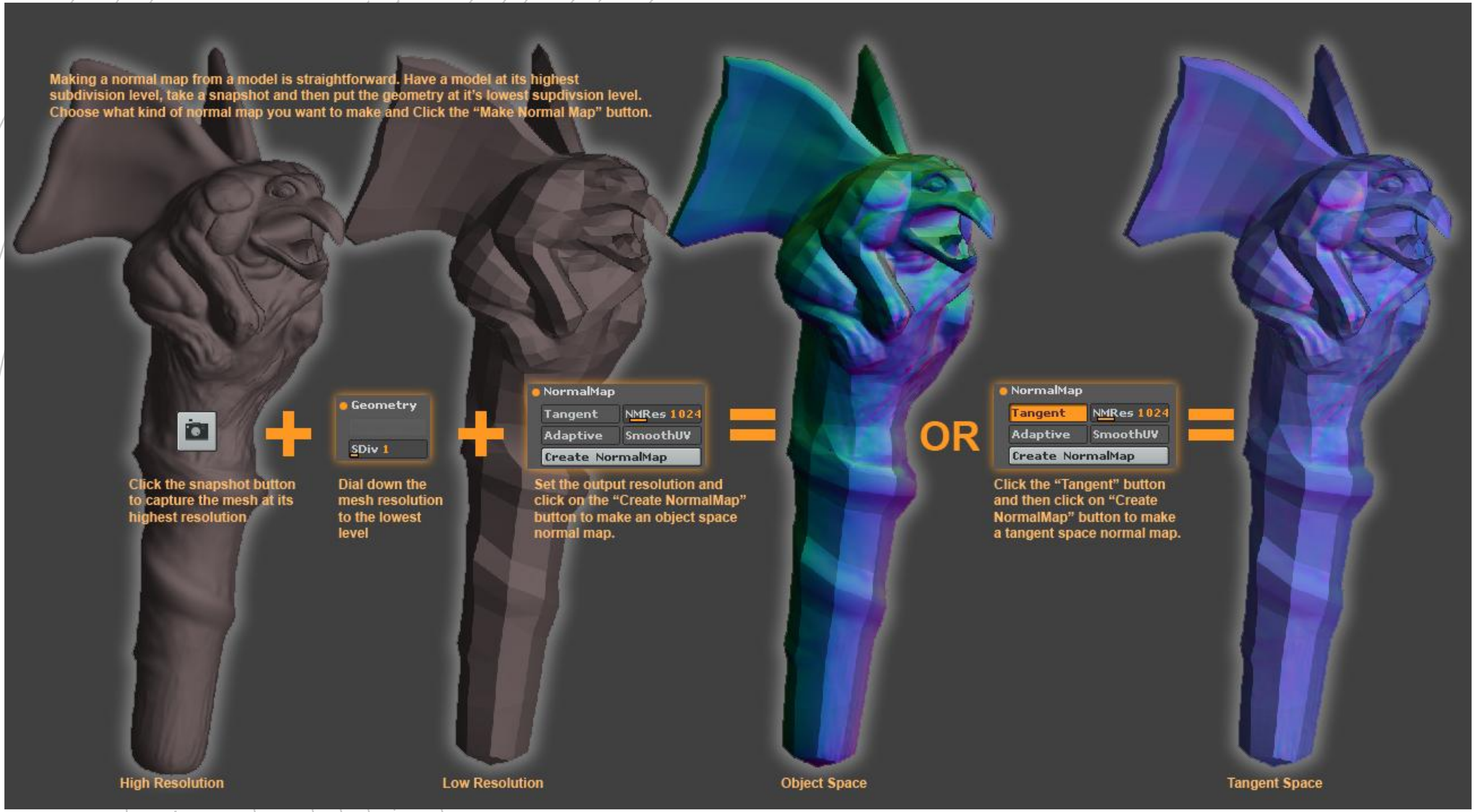


- The light ray (yellow line) hits the mesh (black line) and will bounce off at a certain angle calculated with the normal
- Adding the bump map (2nd image), the light that bounces is changed based on the combination of the two normal maps, therefore the render will look more detailed

Normal Mapping

Introduction

- It is an implementation of bump mapping
  - It gives the appearance of a high-resolution model on an otherwise low-resolution (low polygon) model
  - Simulates the appearance of very detailed wrinkles, indentations, etc on the surface by creating the illusion of light reflecting off of where the extruding parts of the surface, or shadows created depending on where light is angled on the surface.
- Uses the direction of normal vectors in the x, y, z coordinate planes of points on the model's surface to determine each point's R, G, B values.
  - A point's X, Y, Z normal direction determines its R, G, B values, respectively.



Illustrates the process of Normal Mapping on a model

- A low-poly model is exported to another program to create the *normal map* image
- Once uploaded, the model would be subdivided into many small polygons
- The direction of the normal vectors on the surface would be taken in as x, y, z coordinates, which would assign the same values, respectively, for R, G, B
- The *normal map* would represent these colors on x, y, and z planes, and this image would be mapped on top of the low-poly model

Displacement Mapping

Introduction

- Renders bumps as true geometry
- Physically displaces the mesh to which they are applied
  - Creates triangles between texels
- Good for creating detail in low-resolution meshes
- More system intensive than bump maps

Theory

- Geometric position of points are displaced along the surface normal
- Decomposes the definition of its surface to a **macrostructure** geometry
- The **macrostructure** surface is assumed to be a triangle mesh
- Uses height map to displace the **macrostructure** surface in the direction of the normal **surface** vector

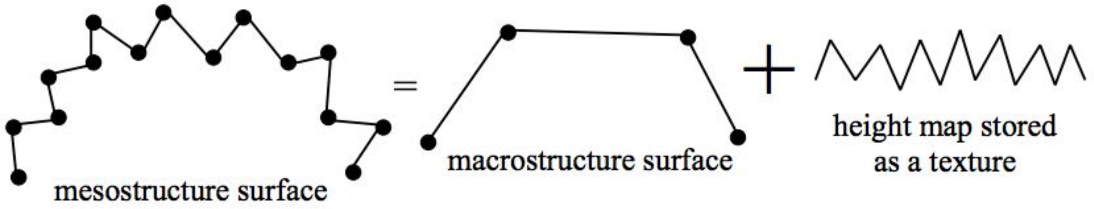


Figure 1: *The basic idea of displacement mapping*

Algorithm

$$\vec{r}(u,v) = \vec{p}(u,v) + \vec{N}^0(u,v)h(u,v).$$

- *r* - Mesostructure surface (result image)
- *p* - Vertex position on the surface (macrostructure - triangle mesh)
- *N* - Outward surface normal vector
- *h* - Height map function
- *u,v* - Texture Coordinates

Relief Mapping

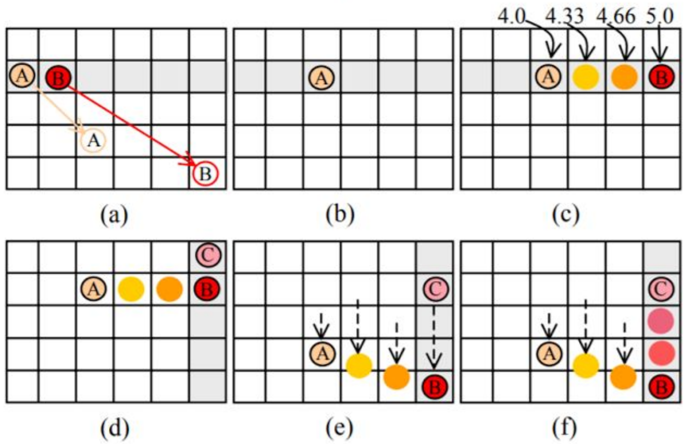
Method

Figure 3. Relief texture mapping: pre-warping followed by standard texture mapping.

Two steps to relief mapping: pre-warping and texture mapping

Method (cont.)

There are two phases in pre-warping: horizontal pass and vertical pass



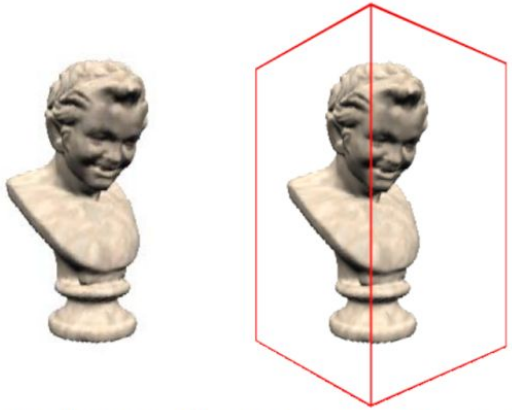


Figure 16. View of the statue (left) obtained by texture mapping two quads, whose boundaries are shown to the right.

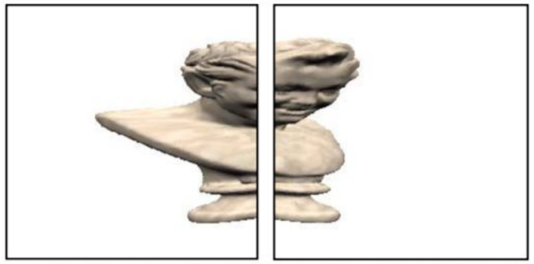
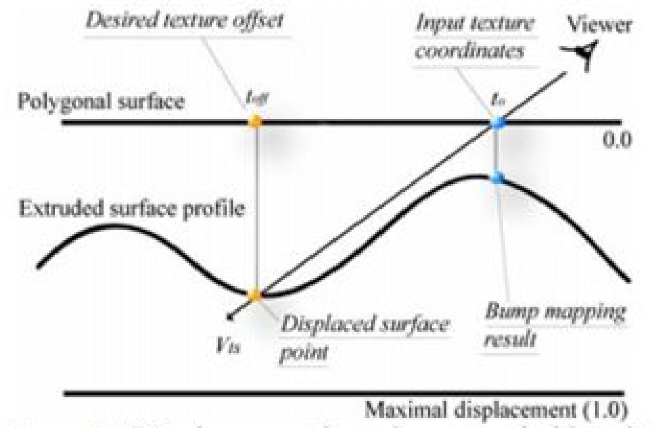


Figure 17. Pre-warped textures used to produce Figure 16.

Parallax Occlusion Mapping

What is parallax occlusion?



igure 3. Displacement based on sampled height ield and current view direction.

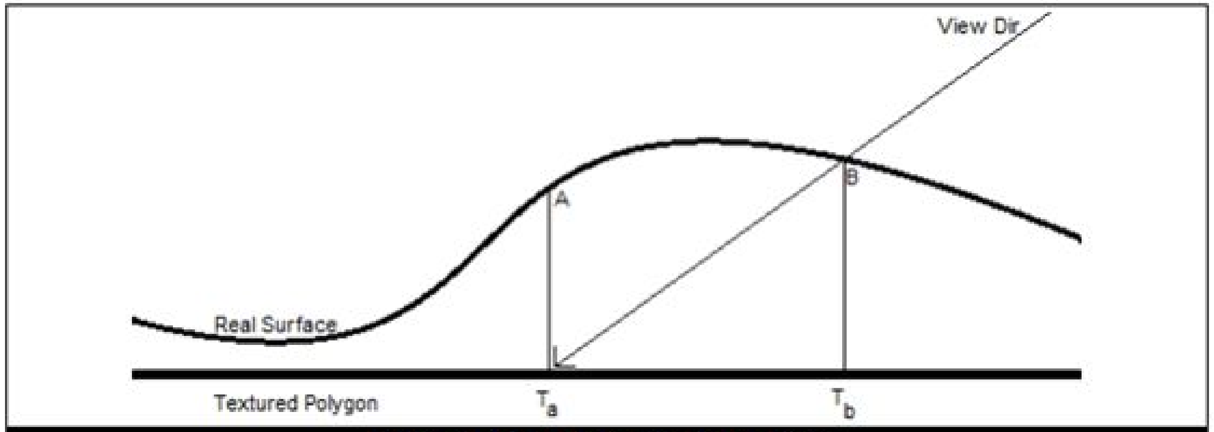
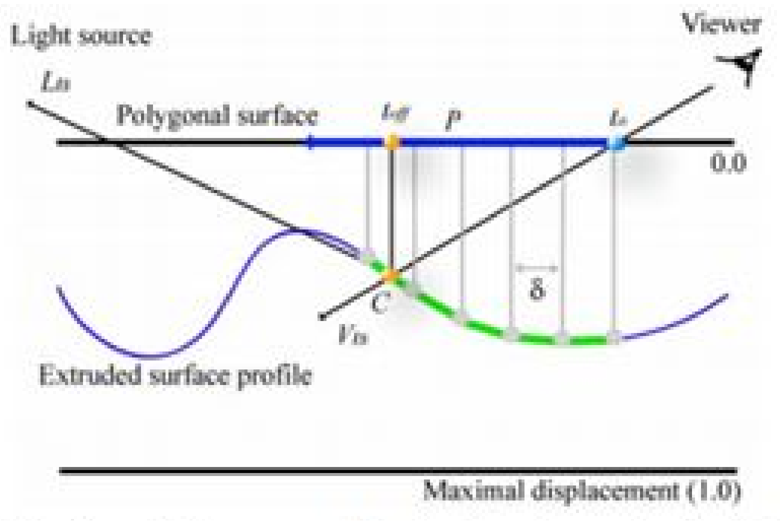


Figure 1. Observed texel vs. real texel

Algorithm



V_{ts} = tangent-space viewing angle

L_{ts} = tangent-space light direction

P = parallax offset vector

t_{off} = shifted texture coordinates

1. Compute V_{ts} and L_{ts}. Interpolate and normalize.
2. Compute P.
3. Find t_{off} by sampling height field and finding parallax offset amount.
4. Calculate any occlusion.
5. Shade according to pixel attributes.