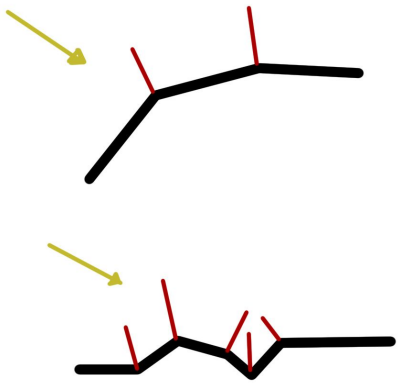
Texture Mapping

Team Mayans

Bump Mapping

- 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



- 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

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

- Decomposes the definition of its surface to the **macrostructure** geometry
- Height map describes the difference of the **mesostructure** and **macrostructure** surfaces in the direction of the **macrostructure** normal vector
- The **macrostructure** surface is assumed to be a triangle mesh

Relief Mapping

The method

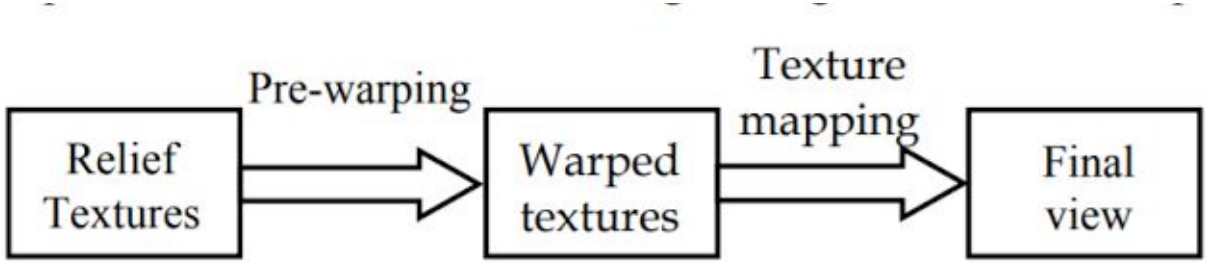
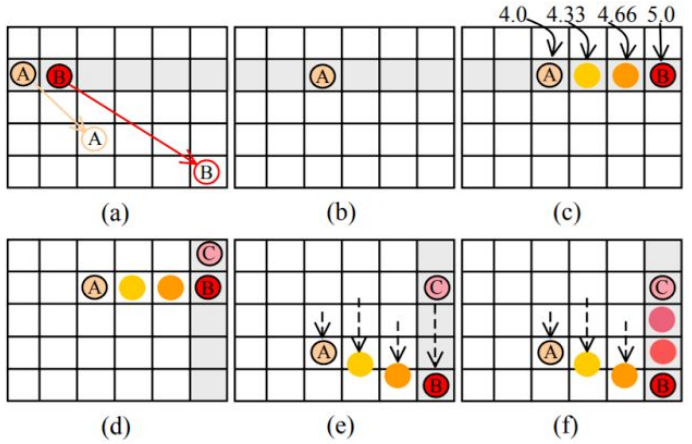


Figure 3. Relief texture mapping: pre-warping followed by standard texture mapping.

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

The method

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



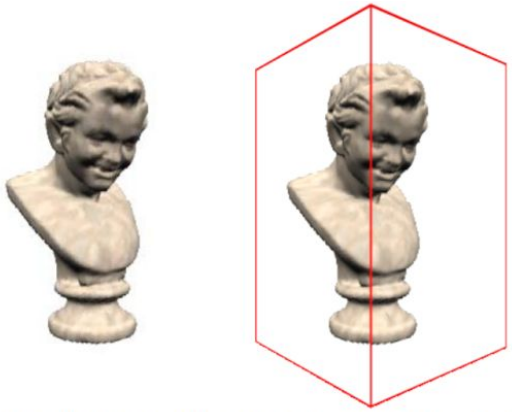


Figure 16. View of the statue (left) obtained by texture mapping two quads, whose boundaries are shown to the right.

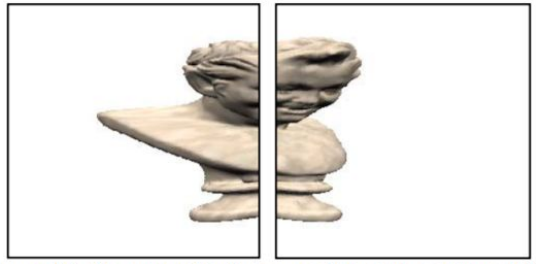
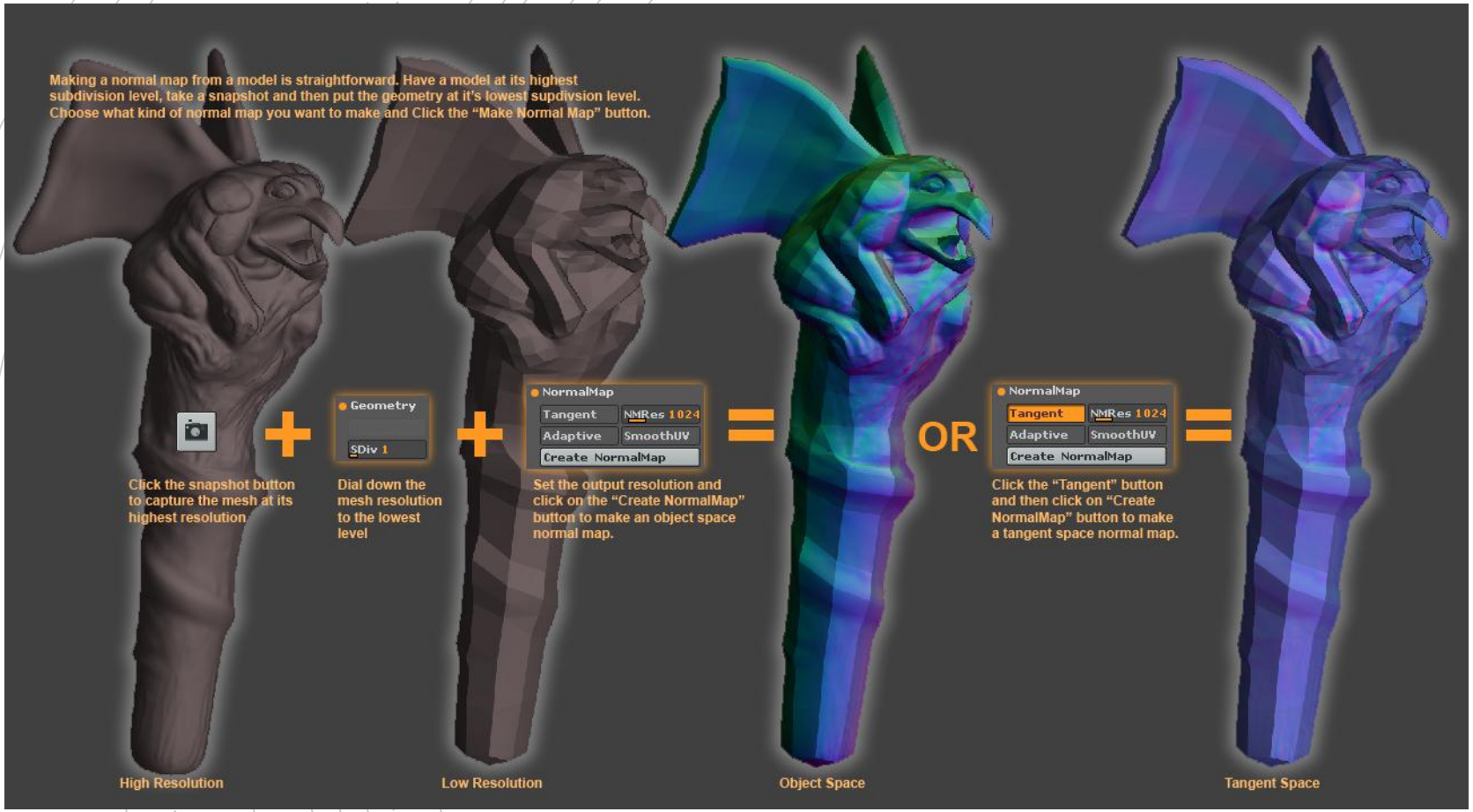


Figure 17. Pre-warped textures used to produce Figure 16.

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