Nicola Frachesen

Richard Li

Natalie Rumak

Xiaoxi Zheng

CAP 3027

Bump Mapping Script

Basic Structure of Presentation:  
 *-Introduction to Bump Mapping*

*-Different Bump Mapping Techniques*

*-Bump Mapping  
 -Normal Mapping  
 -Displacement Mapping  
 -Relief Mapping*

*-Applications, Advances, and In-Depth Properties*

For Reference:

-Nicola == Ni

-Richard == R

-Natalie == Na

-Xiaoxi == X

## Start of Presentation

*Slide 1*

Ni: Hello, everyone. My name is Nicola…

Na: I’m Natalie.

X: I’m Xiaoxi.

R: And I’m Richard.

Ni: We are the Texture Meowpers, and we are your final group of presenters.

Na: Today, We’ll be talking a look at Bump Mapping.

R: No need to worry if you have no clue what that is, as we’ll introduce the basics for each subject.

X: Also, we’ll be presenting a number of visuals and live demos to show everyone how it works.

## Introduction to Bump Mapping

*Slide 2*

Ni: So what is bump mapping anyway? Bump mapping refers to a number of different techniques used in 3D computer modeling to give flat 2D surfaces the illusion of depth. But why is it that we need to create this illusion of depth? 3D models are already, as their name suggests, three dimensional, so why do we need to bother trying to make flat things look like they have depth? Also, how exactly can a simple texture give the illusion of depth, and how good can it really make something look? We will be answering these questions and more in our introduction to the principles of bump mapping.

*Slide 3*

Na: The reason bump mapping was created has to do with the processing power of computers, and wanting to add even more little details to 3D models. 3D models are primarily created using polygons that are stitched together to form a three dimensional object. It can take a lot of processing power for a computer to show what exactly a complex 3D model polygons looks like, a process known as rendering. This mean that, if a model is made up of a lot of polygons, it can take huge amounts of time for a computer to render an image of it. Thus, we use techniques such as bump mapping to put less of a strain on our machines, and get the images we want more quickly.

*Slide 4*

X:

Bump mapping is a technique used in 3D computer modeling to give objects the illusion of depth by altering their appearance [1] - [5]. A number of different techniques exist for creating bump maps, including normal mapping, displacement mapping, and relief mapping. These techniques will be described later in greater detail. Bump mapping typically works by modifying the normal angle of a surface, thus affecting how it is shaded. What this means is that the light which is hitting the surface is made to reflect in a way in which it appears as though the object has a more uneven or complex shape than it actually has [2]. This proves to often be much less intensive for computers to render than a highly detailed model.

Information to be added:

## Bump Mapping

Typically, bump maps are grayscale images that are limited to 8-bits of color information. Thus, only 256 variations of black, gray or white can be calculated. For example, when values in a bump map are close to 50% gray, there’s little to no detail that comes through on the surface. When values get brighter (closer to pure white), details appear more and the surface seems to pop out. Applying the same logic, when values get darker and closer to black they appear to be pushing into the surface. These techniques mean that bump mapping works best for creating tiny details on a model such as pores or wrinkles on skin. However, with few exceptions, the silhouette of the geometry that the bump map is applied to will be unaffected by it.

## Normal Mapping

Normal Mapping is an implementation of bump mapping. It is commonly stored as regular RGB images where the RGB components correspond to the X, Y, and Z coordinates, respectively, of the surface normal. The RGB information tells us the exact direction of the surface normal and how it should be oriented for each and every polygon, and tells an application how the polygon should be shaded.