CSE 333/533: Computer Graphics
Lab 7: Normal Mapping

Instructor: Ojaswa Sharma , TAs: Vishwesh Vhavle , Aadit Kant Jha

Due date: 23:59, 10 November 2023

Introduction

Normal mapping is a technique used in 3D computer graphics to simulate the fine details of a surface without using more polygons. By using a normal map with a texture, small bumps, and dents can be simulated by altering the surface normal during lighting calculations, making a flat surface appear complex and detailed when lit. This is an incredibly efficient method as compared to increasing the polygon count to achieve a similar level of detail.

Fig. 1: *Texture File*

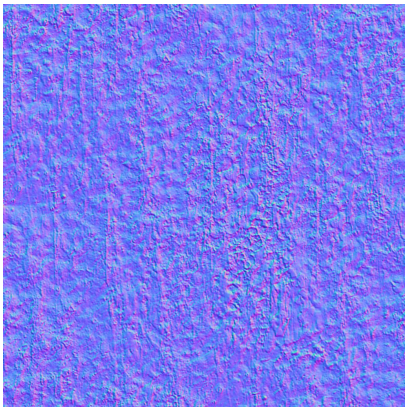

Fig. 2: *Tangent-space Normal Map*

The TBN frame is crucial for normal mapping because it allows the normals in the normal map to be correctly transformed into the same space as the model's geometry. Subsequently, the TBN matrix refers to a matrix composed of the three vectors: **Tangent**, **Bitangent**, and **Normal**. These vectors are perpendicular to each other and form an orthonormal basis used to transform normals from object space to tangent space.

$$M = \begin{bmatrix} t & b & m \end{bmatrix}^T$$

The normal from the normal map is sampled at the given texture coordinate, `TexCoord`. This normal is in the range of [0, 1], so it is scaled to the range of [-1, 1] by multiplying by 2 and subtracting 1. This scaled normal is then transformed by the TBN matrix to bring it into tangent space, aligning it with the surface of the mesh.

Next, the light vector (`l`) and view vector (`e`) are transformed into tangent space using the TBN matrix as well. This is necessary so that lighting calculations can be performed in the same space as the normals from the normal map.

Deliverables

You are provided with the tangent, bitangent and normal vector in the *fragment shader.* The task for the lab is to implement the TBN matrix, to load the normals from the normal map, and finally to use the TBN matrix to transform the loaded normal, eye vector, and light position in the *fragment shader*. Submit a screenshot of the output along with the code for evaluation.

Name the zip file as lab07_<name>_<roll number>.zip
Example: lab07_vishwesh_2020156.zip

References

https://www.opengl.org/documentation/
https://www.khronos.org/opengl/wiki/Rendering Pipeline Overview
https://www.khronos.org/registry/OpenGL-Refpages
https://www.glfw.org/documentation.html
https://www.khronos.org/opengl/wiki/Framebuffer
https://learnopengl.com/Getting-started/Textures
Some Examples: https://github.com/fpaut/my_antons_opengl_tutorials_book

---

Note: Your code should be written by you and be easy to read. You are NOT permitted to use any code that is not written by you. (Any code provided by the instructor/TA can be used with proper credits within your program). Theory questions need to be answered by you and not copied from other sources. Please refer to IIIT-Delhi's Policy on Academic Integrity here.