

# CS 300 |

## Assignment 3 | Normal Mapping

### Files (submit folder) due

- Week 10
- By midnight

Continuing with the previous assignment, you are to add the following functionalities:

1. Generate normal map from a height map
  - Load a height map from disk (metal\_roof\_spec\_512x512.tga). Use only RED channel as height values.
  - Generate the normal map from the height map using discrete derivatives (central difference formula).
  - Store the normal map as RGB texture.
  - This operation needs to be performed on application start-up. Write the code for generating the normal map in your “Init()” (or similarly named) function that initializes the application setup.
2. Calculate tangent (T) and bi-tangent (B) for each vertex of each shape (on CPU)
  - Calculate T and B for each shape’s triangle.
  - Calculate T and B from each triangle’s vertex’s position and texture coordinate, i.e. do NOT calculate T and B from the shape parametric equation.
  - The T and B of each vertex is the average of the Ts and Bs of all triangles that shares that vertex.
  - **Note that calculated T and B are in model space.**
3. Use normal from normal map for lighting calculation
  - Pass in calculated T and B to vertex shader. Note that T and B are per-vertex attribute, so, use attribute variable to pass them in.

### Copyright Notice

Copyright © 2010 DigiPen (USA) Corp. and its owners. All rights reserved.

No parts of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language without the express written permission of DigiPen (USA) Corp., 9931 Willows Road NE, Redmond, WA 98052.

### Trademarks

DigiPen® is a registered trademark of DigiPen (USA) Corp.

All other product names mentioned in this booklet are trademarks or registered trademarks of their respective companies and are hereby acknowledged.

- In vertex shader, transform T, B and N vectors to view space by multiplying them with ModelView matrix. These vectors form the TBN matrix to move entities from Tangent-Space to View-Space and vice versa.
  - In fragment shader
    - Setup sampler for the normal map
    - Calculate appropriate (u, v) values depending on the projector function
    - Sample the normal map and transform it to view space using the TBN matrix
    - Use the calculated normal for your lighting calculation
4. Scene setup:
- Same setup as assignment 2
5. In addition to inputs in assignment 2, have inputs to:
- Toggle the normal mapping on/off, i.e. use geometry normal or normal from the normal map.
  - Debug-draw: Draw TBN coordinate frame over the model for every vertex (See Figure 1)
    - **Hint:** Use the same method that you have used to draw Face and Vertex normals!
  - Toggle T and B (one at time) rendering as color. Render the T and B after model space to view space transformation. (See Figure 2)
  - Toggle normal map (use the normal map (R, G, B) as fragment color) rendering on/off.
6. Note:
- Make sure that at least one point (or spot) light is working correctly. In particular, the specular must work correctly.
  - T and B need to be calculated for each shape.

### Assignment Submission Guideline

Please refer to the syllabus for assignment submission guideline. Failure to the submission guidelines correctly might cause you to lose point.

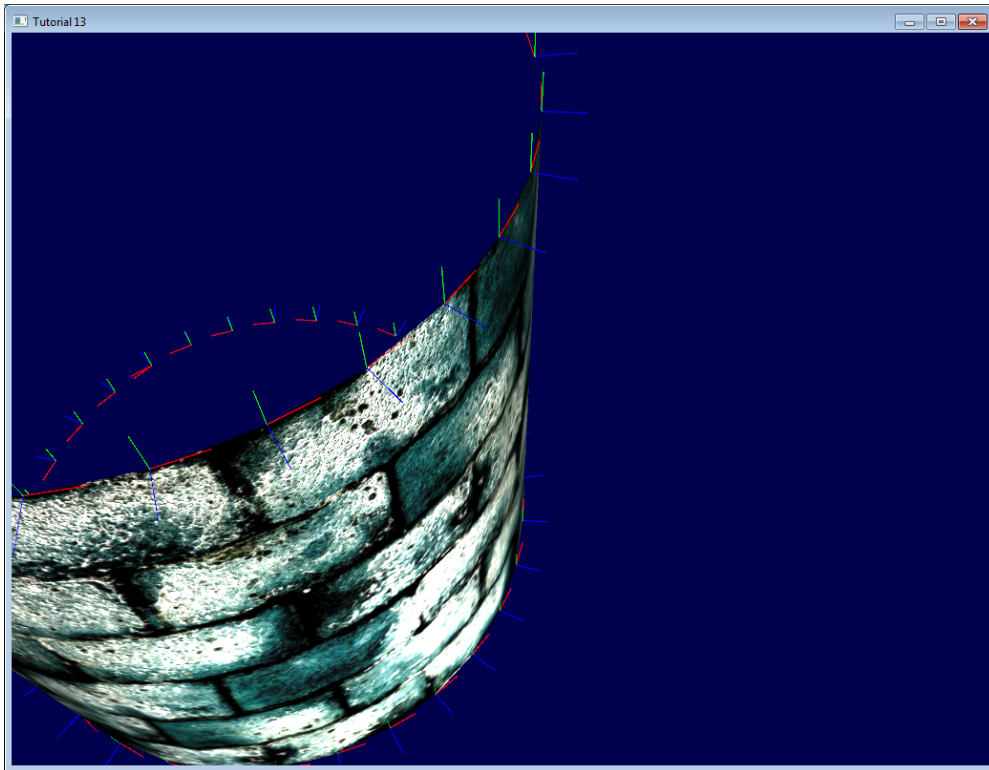


Figure 1: Debug draw TBN vectors

<http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-13-normal-mapping/>

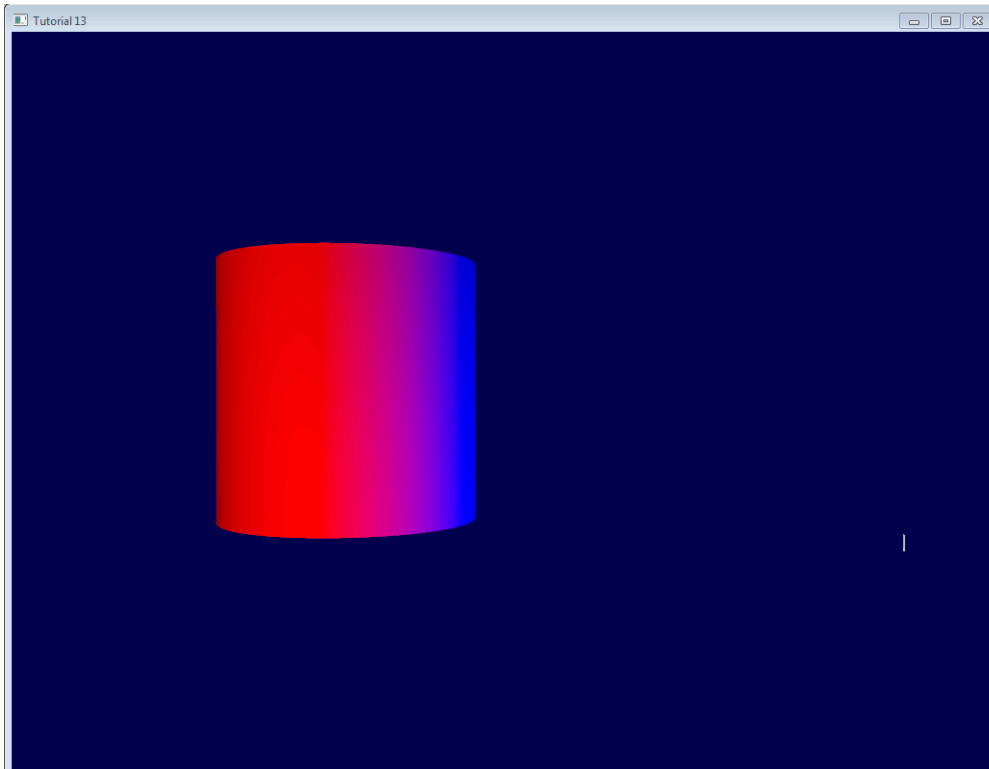


Figure 2: Draw T/B/N as RGB fragment colors

<http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-13-normal-mapping/>

## GRADING SHEET

Implementation Point	Grade	Points obtained	Comments
<b>Normal Map generation</b>	<b>15%</b>		
Correct normal calculated for the height map texels	5		
Correct handling of edge case	3		
Correct range mapping to store the normal components in RGB	5		
Output of the normal map generation passed correctly as texture	2		
<b>Tangent &amp; Bitangent calculation</b>	<b>35%</b>		
Correct calculation of T	7		
Correct calculation of B	7		
Correct handling for division by 0	7		
T & B for a vertex correctly calculated as average of all faces incident on the vertex	7		
T & B arrays are correctly passed to the shader as vertex attributes	7		
<b>Debug drawing functionality</b>	<b>15%</b>		
Normal map used as a texture over the geometry	5		<b>Failure to implement the debug drawing for the assignment will result in zero grade for the rest of the sections of the assignment</b>
Render the Tangent vectors per vertex	5		
Render the Bitangent vectors per vertex	5		
<b>Normal Map usage</b>	<b>25%</b>		
Transforming T, B and N vectors into the correct space	5		
Correct sampling of the normal map	5		
Correct calculation of the range values of the vectors from RGB values	5		
Correct transformation of the sampled normal in view space	5		
Correct usage of the new fragment normal in the lighting equation	5		
<b>Miscellaneous issues</b>	<b>10%</b>		
Missing information in README	4		<b>You will receive zero credit for the assignment if your application does not compile or is</b>
Application does not compile	2		
Application cannot be executed	2		
Scene setup incorrect	2		

	incorrectly set up.
--	---------------------