# CS-411 – Assignment 6 (5%)

Normal Mapping

Due by: December 2, 2017

In this assignment you need to implement normal mapping to map normals specified in a normal map image to a surface. The object to be rendered (cube) is read from an input file containing vertex and face information. After rendering the object keyboard keys may be used to move the camera around it. Answer the following questions and implement the necessary code. Submit the assignment using the submission instructions of the first assignment. A skeleton program and sample object models will be provided. You are not required to use the skeleton program and may modify it as needed.

1. General questions (use JavaScript and/or manual computations ):

   (a) Explain the difference between forward and inverse texture mapping. Explain which is the preferred (forward or inverse mapping) method for performing texture mapping.

   (b) Let $v_1 = [1, 2]$, $v_2 = [4, 2]$, $v_3 = [4, 6]$ be texture coordinates (i.e. vertices in a texture image) and $v'_1 = [1, 2]$, $v'_2 = [1, 4]$, $v'_3 = [4, 2]$ be the corresponding coordinates in the rendered image that is being produced. Write the equation for computing the forward map between texture and image coordinates, then compute the forward map. Write the equation for computing the inverse map, then compute the inverse map.

   (c) Explain the advantage of bump-mapping over standard texture mapping in increasing the level of realism. Explain how bump-mapping creates the illusion of bumps without modifying the geometry of the surface.

   (d) Given a bump image $b(x, y)$ explain how the normal $n = (n_x, n_y, n_z)$ at location $(x, y)$ is changed using the bump image.

   (e) Let the intensity of the image at location $(4, 5)$ be 20 and the intensity at location $(5, 5)$ be 25. Compute the displacement that will be added to the $x$ coordinate of the normal using bump mapping.

   (f) Given that the R,G,B intensity you read from a normal map is $(0.2, 0.3, 0.9)$, what is the normal vector that will be decoded from this intensity.

   (g) Given a triangle with vertices $p_0 = [1, 2, 1]$, $p_1 = [4, 2, 2]$, $p_2 = [4, 6, 3]$ and texture coordinates of $(u_0, v_0) = [1, 2]$, $(u_1, v_1) = [1, 4]$, $(u_2, v_2) = [4, 2]$, compute the normal $N$, tangent $T$, and bi-tangent $B$.

   (h) Given the vectors $T, B, N$ from the previous question, write the transformation matrix that will transform the normals from tangent space (normal map coordinate system) to the world coordinate system.

   (i) Given the transformation matrix from the previous question, write the transformation matrix that will transform the light direction vector $L$ to the normal map coordinate system.

2. Program implementation (use JavaScript and WebGL):

   (a) *Input file format:* The input text file describes one 3D object. Each line in that file describes either a vertex or a face. Empty lines, and lines beginning with a # should be ignored. The following are possible lines in the input file:

```
# Specify a VERTEX:
v <float-coord-x> <float-coord-y> <float-coord-z>

# Specify a VERTEX normal:
vn <float-coord-x> <float-coord-y> <float-coord-z>

# Specify a VERTEX texture coordinates:
vt <float-coord-x> <float-coord-y>

# Specify a FACE (triangle):
f <vertex-index-1> <vertex-index-2> <vertex-index-3>
```

   (b) *Load the normal map:* Load the normal map as a texture and convert the texture values to normal vectors (multiply by 2 and subtract 1).

   (c) *Determine the TBN matrix:* For each triangle solve a system of equations using the texture coordinates to compute tangent and bi-tangent vectors. Using the normal N, tangent T, and bi-tangent B, compose the TBN matrix that will transform between tangent and woorld coordinates.

   (d) *Perform light computations in world coordinates:* Using the TBN matrix perform light computations in world coordinates (apply TBN to transform the normals from tangent space) and render the object.

   (e) *Perform light computations in tangent coordinates:* Using the transpose of the TBN matrix transform the illumination vectors to tangent coordinates and perform light computations there to render the object.

   (f) *Texture and normal mapping:* Use two texture maps, one for texture, and one for normals, and render the the object with texture and normal maps.

   (g) *Bonus question:* Instead of loading the normal map from a file, use a single texture map to compute image gradients and use them as normal displacements to produce bump mapping.

3. Submit via bitbucket using the submission procedure of assignment 1.