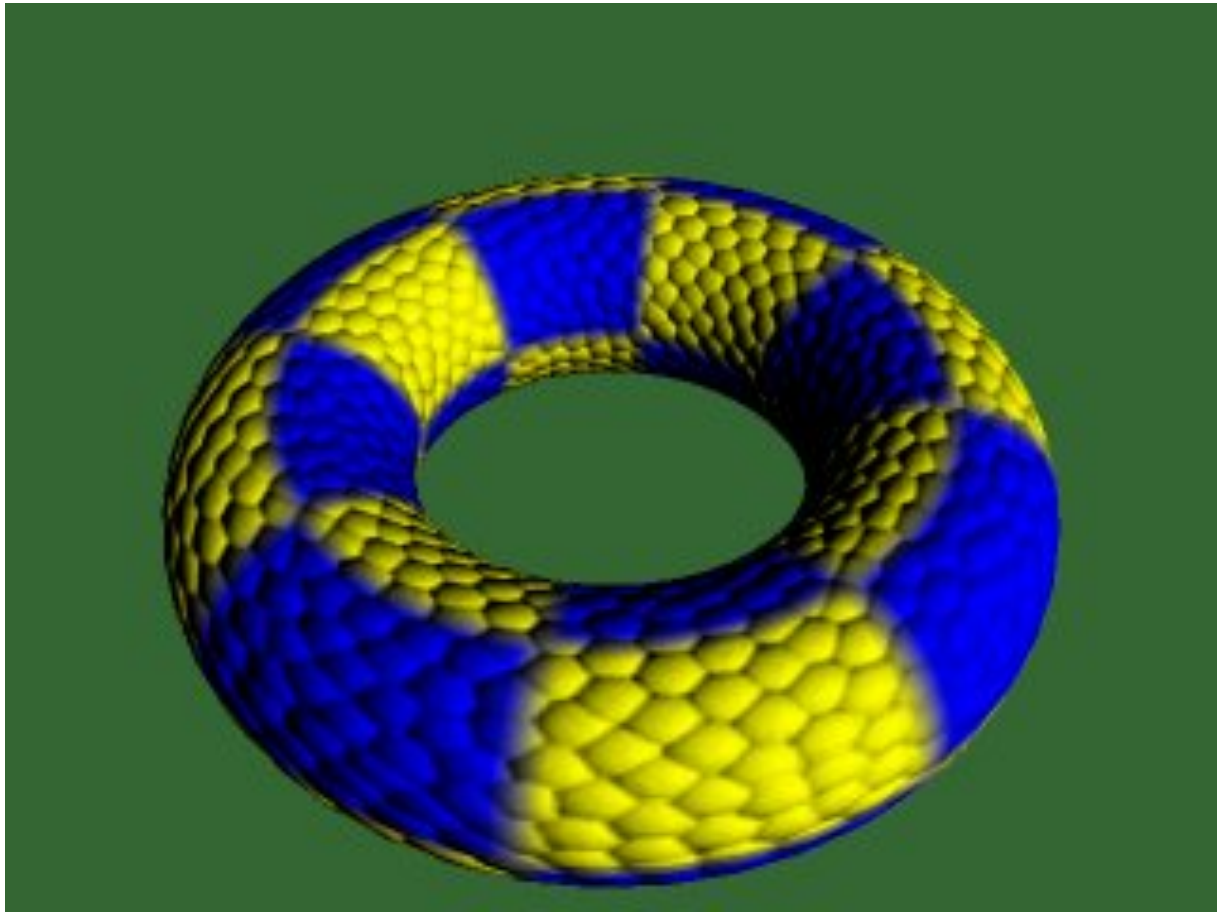
DigiPen
INSTITUTE OF TECHNOLOGY

CS300
Bump Mapping & Normal Mapping

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved.

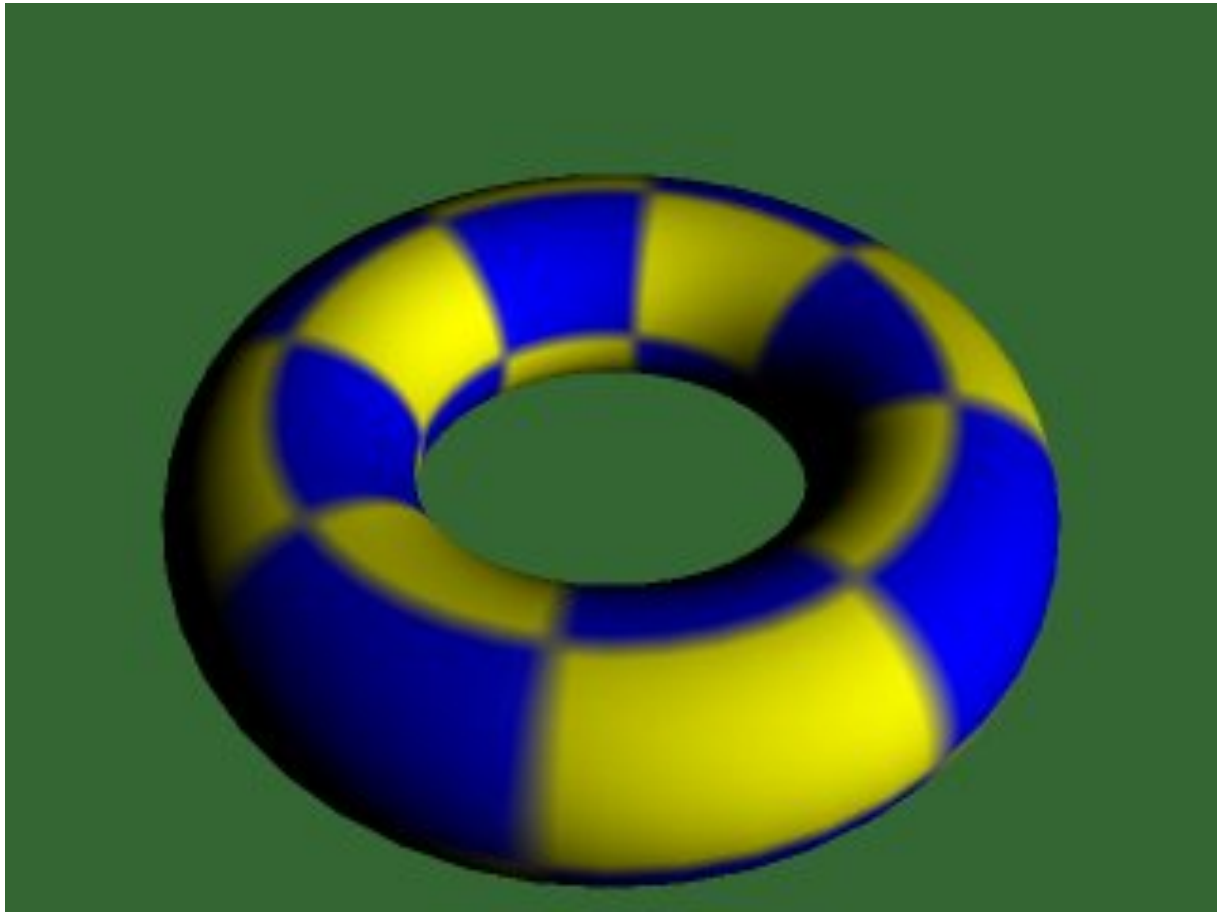
DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping



DigiPen
INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping



Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping

- Bump mapping is a technique developed by James Blinn (1978)
  - "Simulation of wrinkled surfaces," SIGGRAPH 1978
- The method allows a surface to appear wrinkled or dimpled without the need to model these depressions geometrically.
- To achieve this effect, the surface normal at every pixel is modified according to the information supplied in a height map.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*

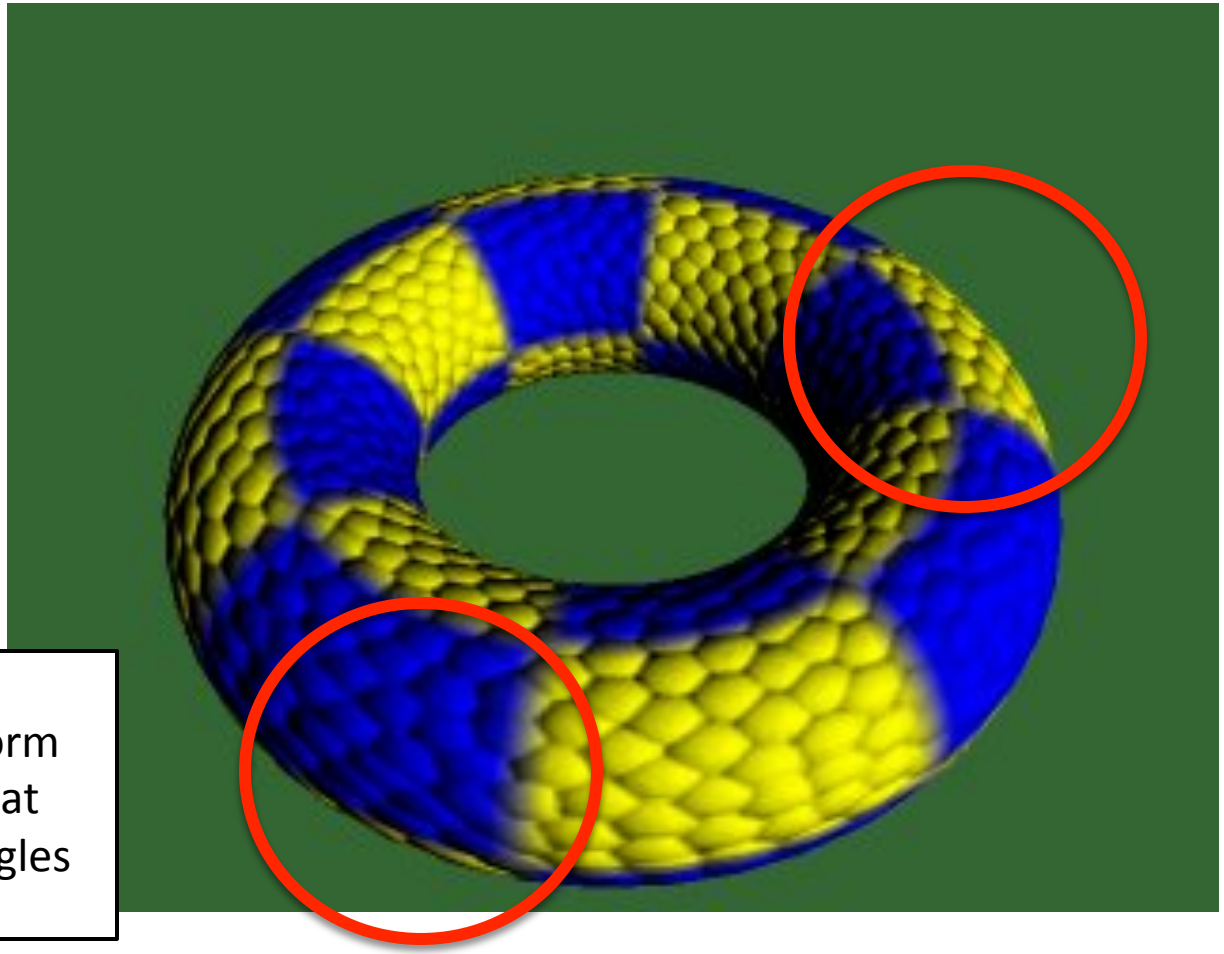
- Key Idea
  - Surface variations **do not exist in the model**, causing the illusion to break around the silhouette of objects.

- At these edges the viewer notices that there are no real bumps but just smooth outlines.

- Good to use when the bump is 'small' relative the surface size.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping

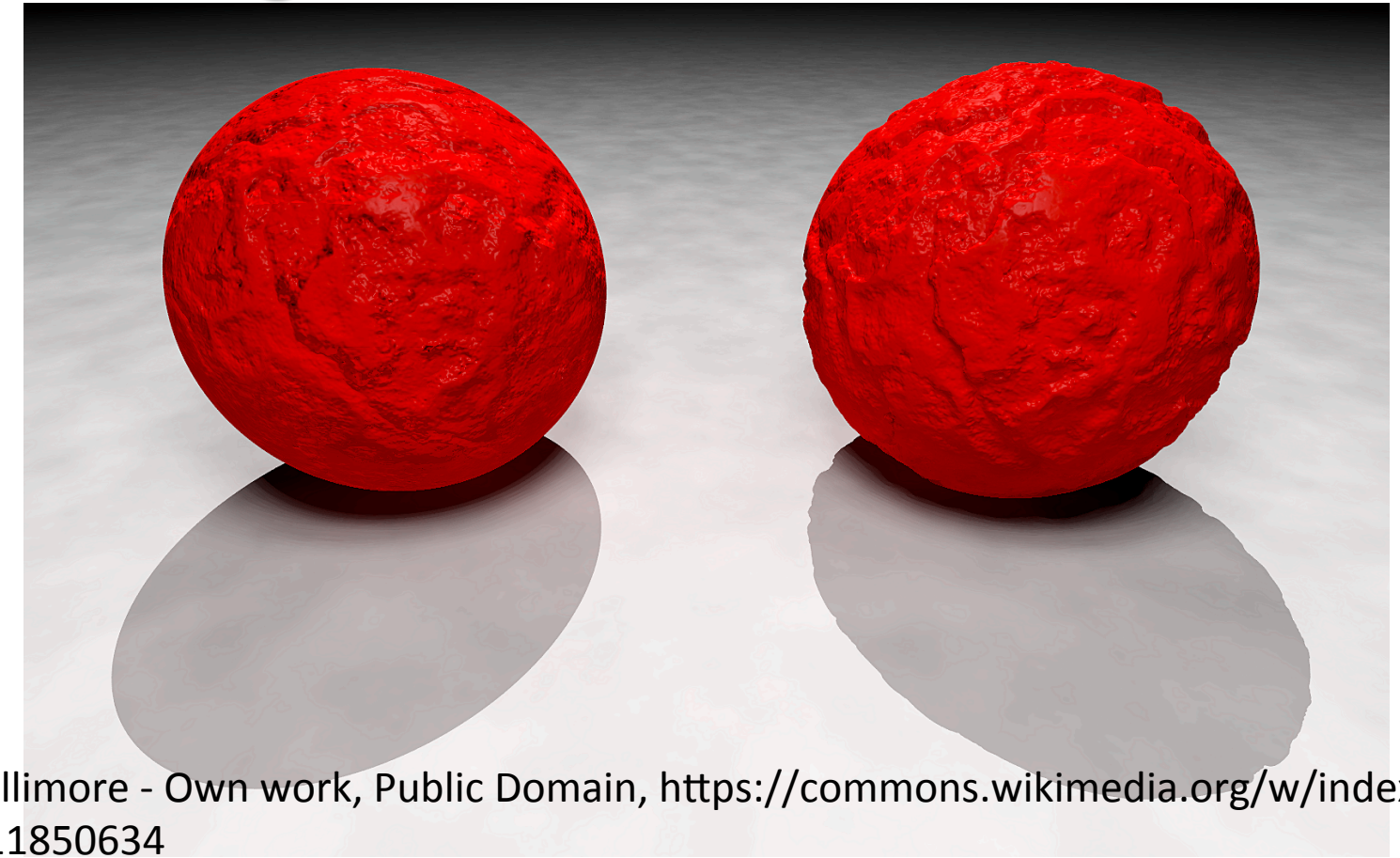


Smooth Silhouette

Non-uniform features at grazing angles

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping Vs Volume-based Rendering



By GDallimore - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=11850634

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

OpenGL Lighting Model

- What components of the light model determine the color at a point on the surface?
  - Light vector (position of the light)
  - Surface normal (local surface features)
- We trick the user by manipulating the lighting model components
  - Which one can we (reliably, consistently & efficiently) change?

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping Calculations

- What do we need?
  - Light position (defined per scene - uniform)
  - Surface normal (defined per vertex – attribute/out)
  - View vector (defined per fragment - out)

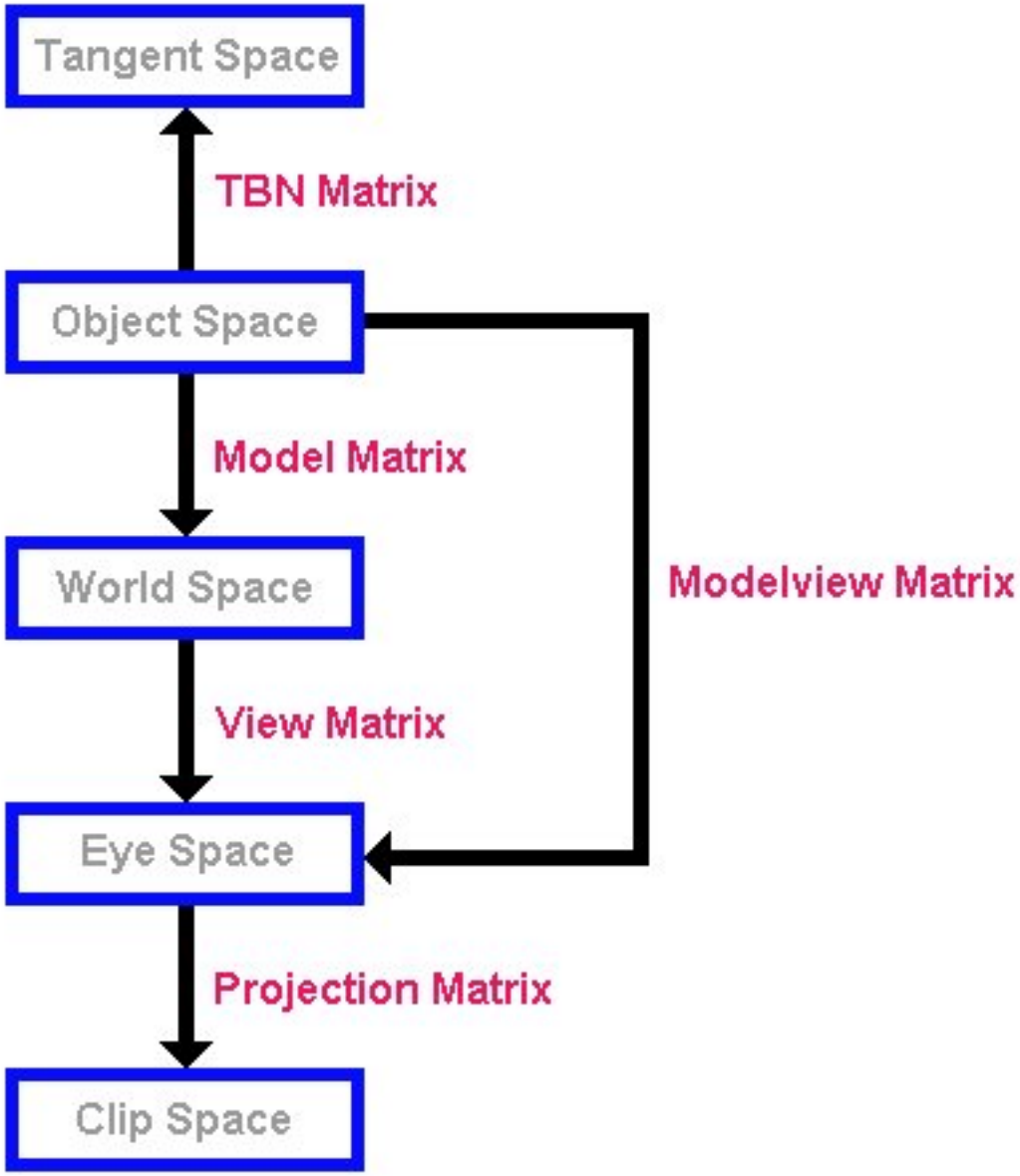
- Let us try the naïve approach
  - Change the normal in the fragment shader
  - Use this normal in the lighting equation

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Lighting Model Redux

- Phong lighting model relies on dot products to simulate illumination distribution across a surface
- Our vectors are all in the same space when we compute the fragment color (Eye space)

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved



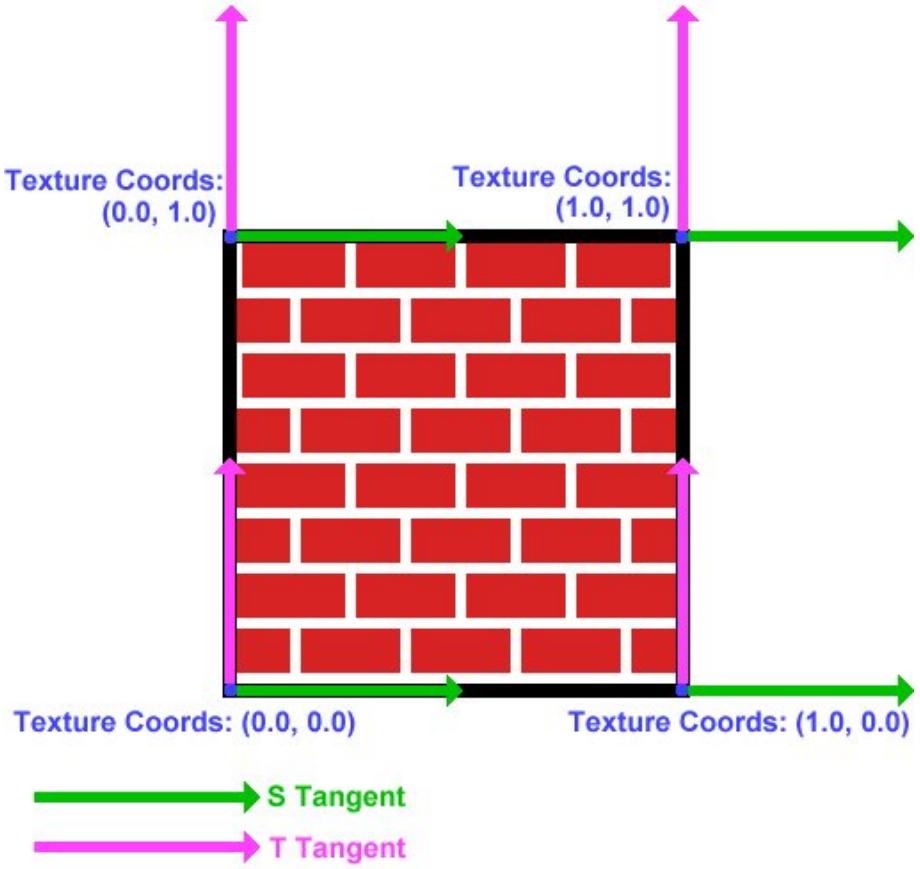
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Tangent Space

- Is uniquely defined for EVERY POINT on the surface
- Is represented by 3 axes in 3D
  - Similar to X, Y and Z axes in Object Space
- Simpler interpretation
  - Z = N, the normal (the outward direction)
- What about the corresponding “X” and “Y” axes?

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Tangent Space Representation



DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Calculating lighting equation in Tangent Space

- Convert the Light vector (L) and View direction (V) to the tangent space
- Apply the transformations to the normal
  - Or lookup a new value for the normal
- Perform the lighting calculations with the new normal

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Transforming L into the TBN space

$$\begin{bmatrix} L'x \\ L'y \\ L'z \end{bmatrix} = \begin{bmatrix} Sx & Sy & Sz \\ Tx & Ty & Tz \\ Nx & Ny & Nz \end{bmatrix} \begin{bmatrix} Lx \\ Ly \\ Lz \end{bmatrix}$$

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*

- The technique converts a two-dimensional height field B(u, v), called the bump map, which represents some desired surface displacements, into appropriate perturbations of the local surface normal.

- When this surface normal is used in the shading equation, the reflected light calculations vary as if the surface had been displaced.

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*

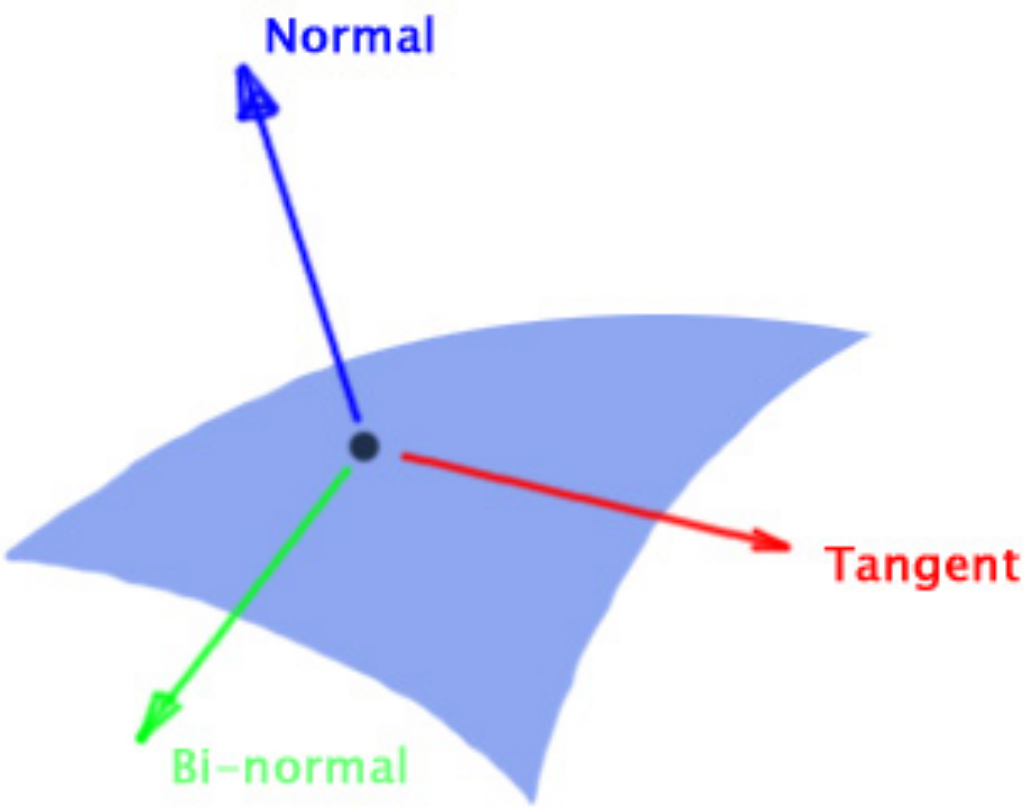
- Consider a point P(u, v) on a parameterized surface, we define the surface normal at the point to be

$$N = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} \qquad N = P_u \times P_v$$

- Where P_u and P_v are the partial derivatives lying in the tangent plane to the surface at point P(u, v)

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

TBN Space



DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*

- What we want to do is to have the same effect as displacing the point P along the surface normal at that point by an amount B(u, v)

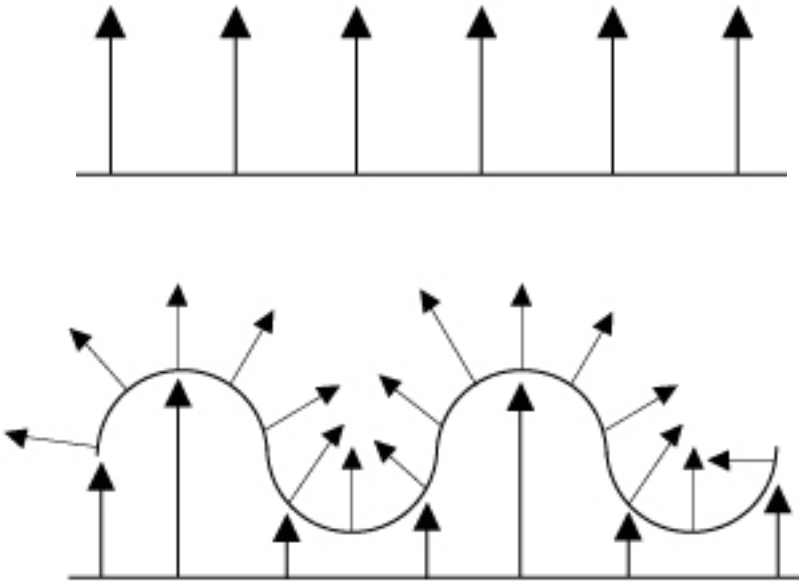
- We get the new surface P' defined by

$$P'(u,v) = P(u,v) + B(u,v)N$$

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

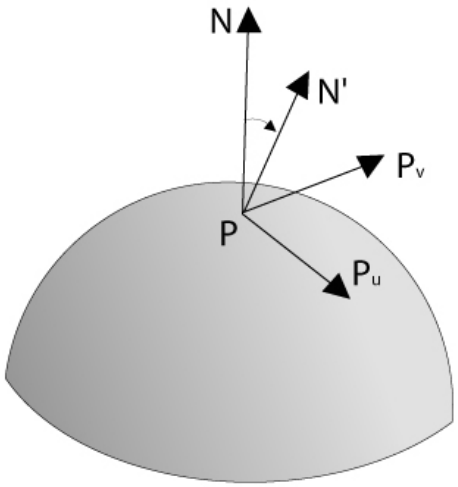
Bump Mapping *(cont'd)*

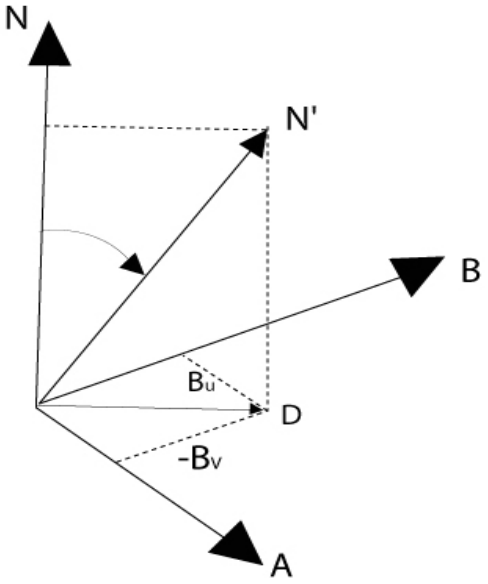
- The figure below shows a one-dimensional analogue to the process



DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*





DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*

- Now we need to find the new normal N' to the surface P' at every point (u, v) of the surface, this is done as follows

$$N' = P'_u \times P'_v$$

$$P'_u = P_u + B_u N + B(u,v) N_u$$

$$P'_v = P_v + B_v N + B(u,v) N_v$$

- N_u and N_v is assumed to be zero because we assumed that the normal to vary very little on the surface.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*

$$N' = N + B_u N \times P_v + B_v P_u \times N$$

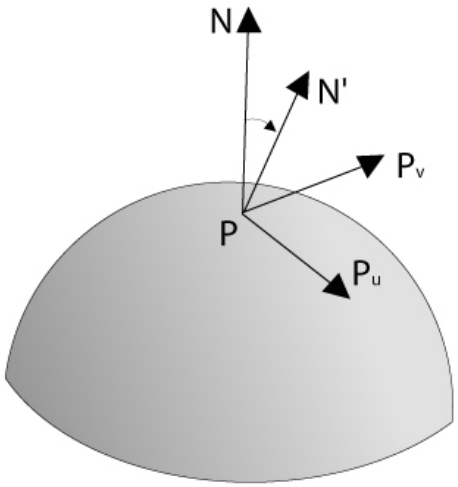
$$N' = N + B_u N \times P_v - B_v N \times P_u$$

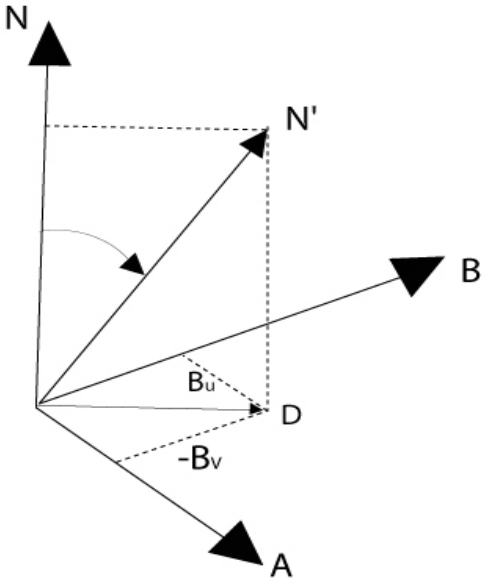
$$N' = N + (B_u A - B_v B)$$

$$N' = N + D$$

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*





DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Bump Mapping *(cont'd)*

- For every point (u, v) on the surface represented by P, we use the new normal N' in the illumination calculations.

- This will create the bumpy surface we are looking for.

DigiPen
INSTITUTE OF TECHNOLOGY

CS300
Normal Mapping

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved.

DigiPen
INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Normal Mapping

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Normal Mapping

- Normal mapping is just another form of Bump Mapping.
- Instead of storing heights in a bump map, we will store normals in a normal map.
- Depending on how the normals are measured and stored, normal mapping comes in two varieties:
  - Object-space normal mapping
  - Tangent-space normal mapping

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Normal Mapping

I. Constructing normal maps from height maps

II. Constructing normal maps from high polygon models

III. Tangent Space

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Constructing normal maps from height maps

- Given a height map H(i,j), where H represents the height at any pixel (i, j), a normal map can be derived by considering the tangents or partial derivatives in the u and v directions.

- The normal to the pixel is formed by taking the vector resulting from the cross product of the two tangents.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Constructing normal maps from height maps *(cont'd)*

- First we need to find the two tangents at any pixel (i, j).
- Let S(i, j) be the tangent in the u direction and T(i, j) be the tangent in the v direction

$$S(i,j) = \langle 1, 0, aH(i+1,j) - aH(i-1,j) \rangle$$

$$T(i,j) = \langle 0, 1, aH(i,j+1) - aH(i,j-1) \rangle$$

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Constructing normal maps from height maps *(cont'd)*

- The constant "a" is a scale factor that can be used to vary the range of the height, making the perturbed normals more or less pronounced.
- To calculate the normal vector N at the point (i, j) we take the cross product of S and T at that point:

$$N(i,j) = \frac{S(i,j) \times T(i,j)}{\| S(i,j) \times T(i,j) \|}$$

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Constructing normal maps from height maps *(cont'd)*

- The normalized vector N needs to be stored within the normal map.
- We know that each components of the vector vary between -1 and 1.
- And the texture (assuming it is an RGB texture with 8-bits per component) can store a number between 0 to 255 per component.
- We need to map the [-1,1] range to [0,255].

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Constructing normal maps from height maps *(cont'd)*

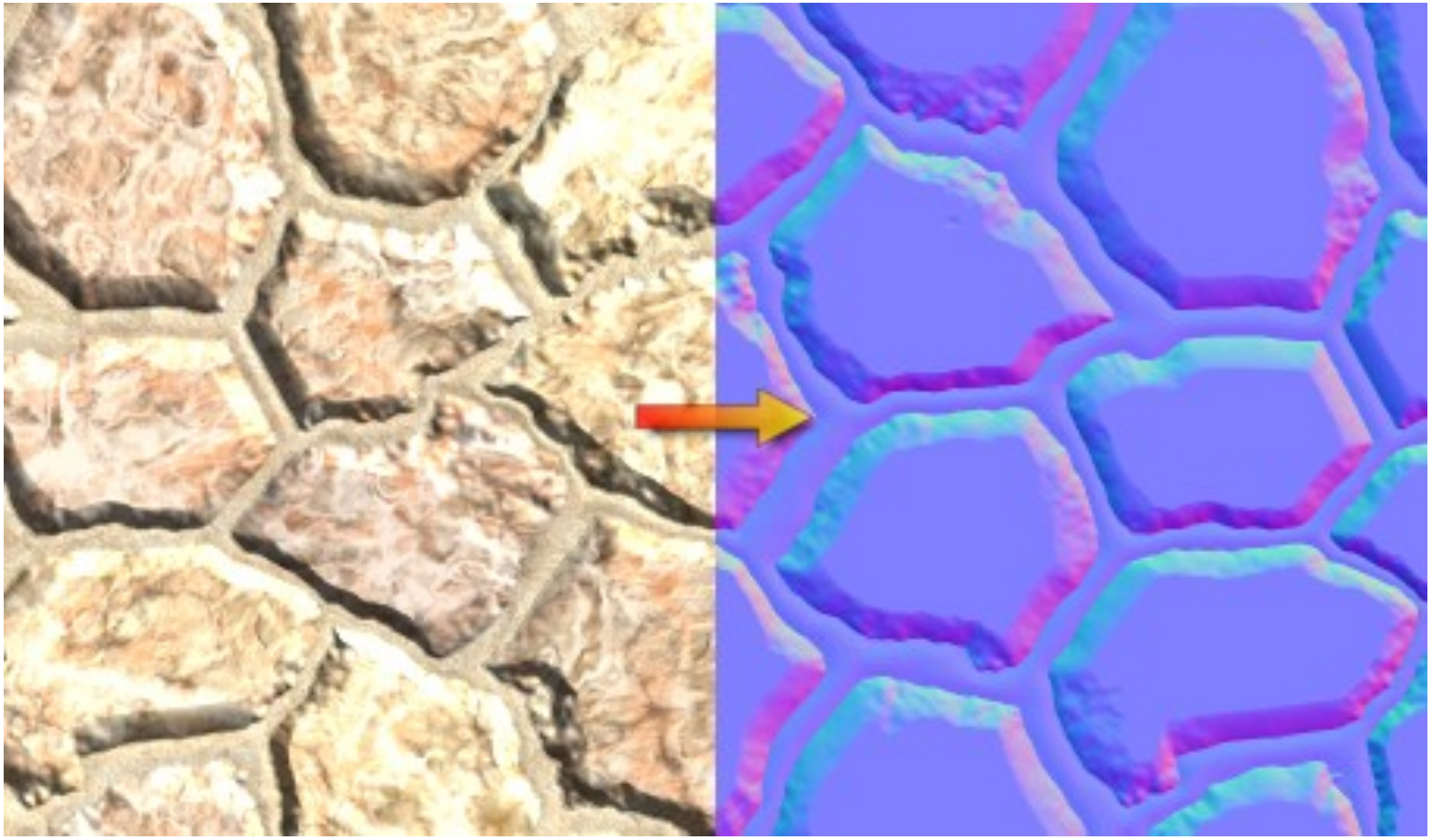
- The equation to map the x coordinate to the red channel:

$$R = 255 * 0.5 * (N_x + 1)$$

- It is customary to store the x coordinate in the red channel, the y coordinate in the green channel and the z coordinate in in blue channel.

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Example Normal Map



DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Using Maps to encode surface

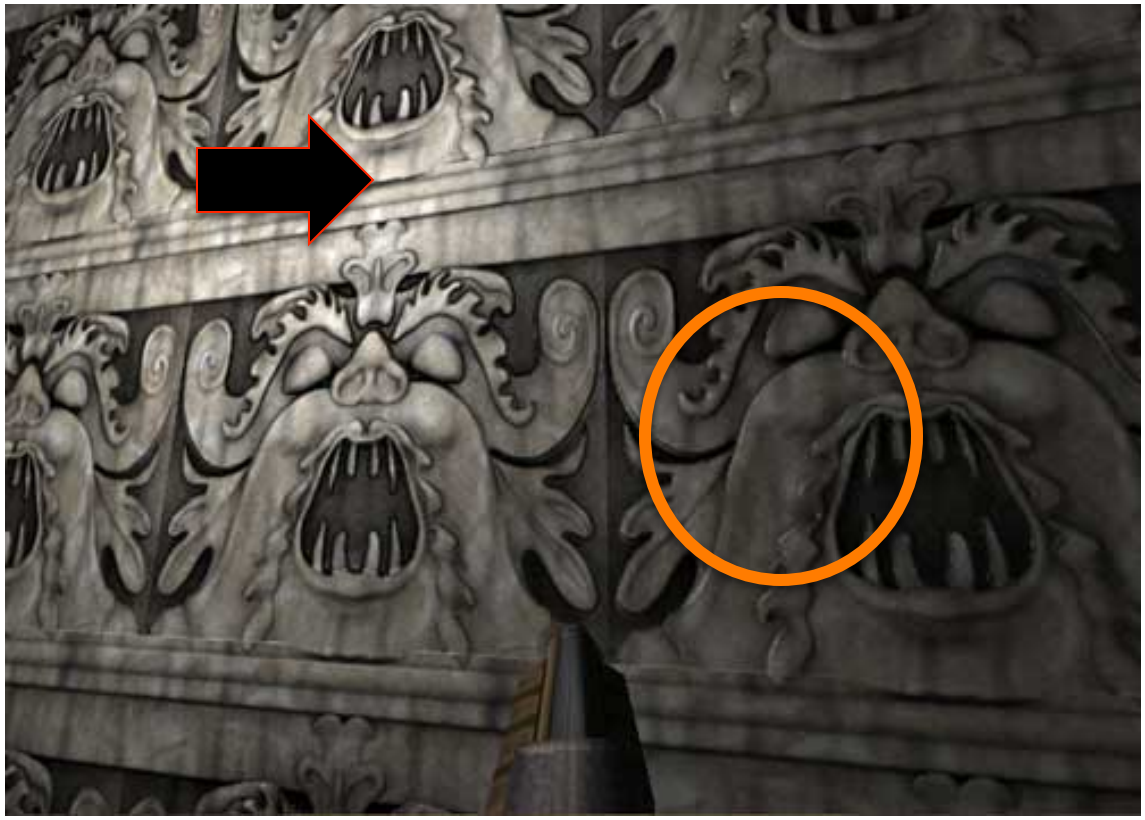
Texture
(shading and lighting)

Flat walls

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Building Height Maps (Bump Mapping)

Bump Map



Better lighting, but still feels flat

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Normal Map

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Normal Map Dissected

Red

Green

Blue

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Normal Map Vs Bump Map

http://www.zarria.net/nrmphoto/nrmphoto.html

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

II. Constructing normal maps from high polygon models

- The goal of normal mapping is to allow low polygon models to appear as though they were more detailed.
- We need to create two versions of the same model, a high polygon version and a low polygon version.

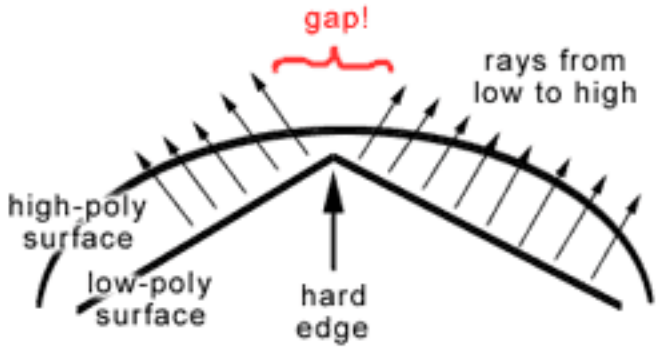
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

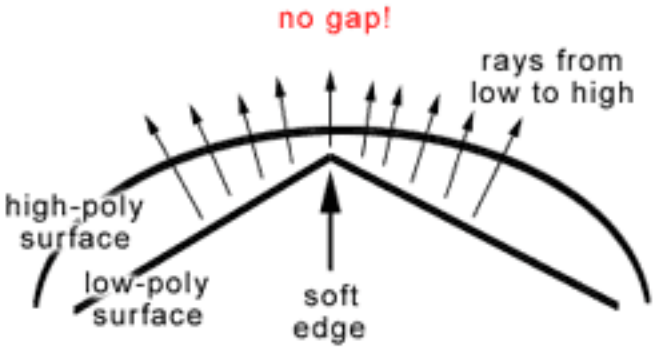
II. Constructing normal maps from high polygon models *(cont'd)*

- Afterwards, we cast rays from the low polygon version to the high polygon version, and we return the normal that collided with each ray.
- The normals are then mapped from the [-1, 1] range to the [0, 255] range and saved in the normal map.
- The number of rays sent is determined by the size of the normal map, which is usually the same size as the texture map.

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Gotcha!

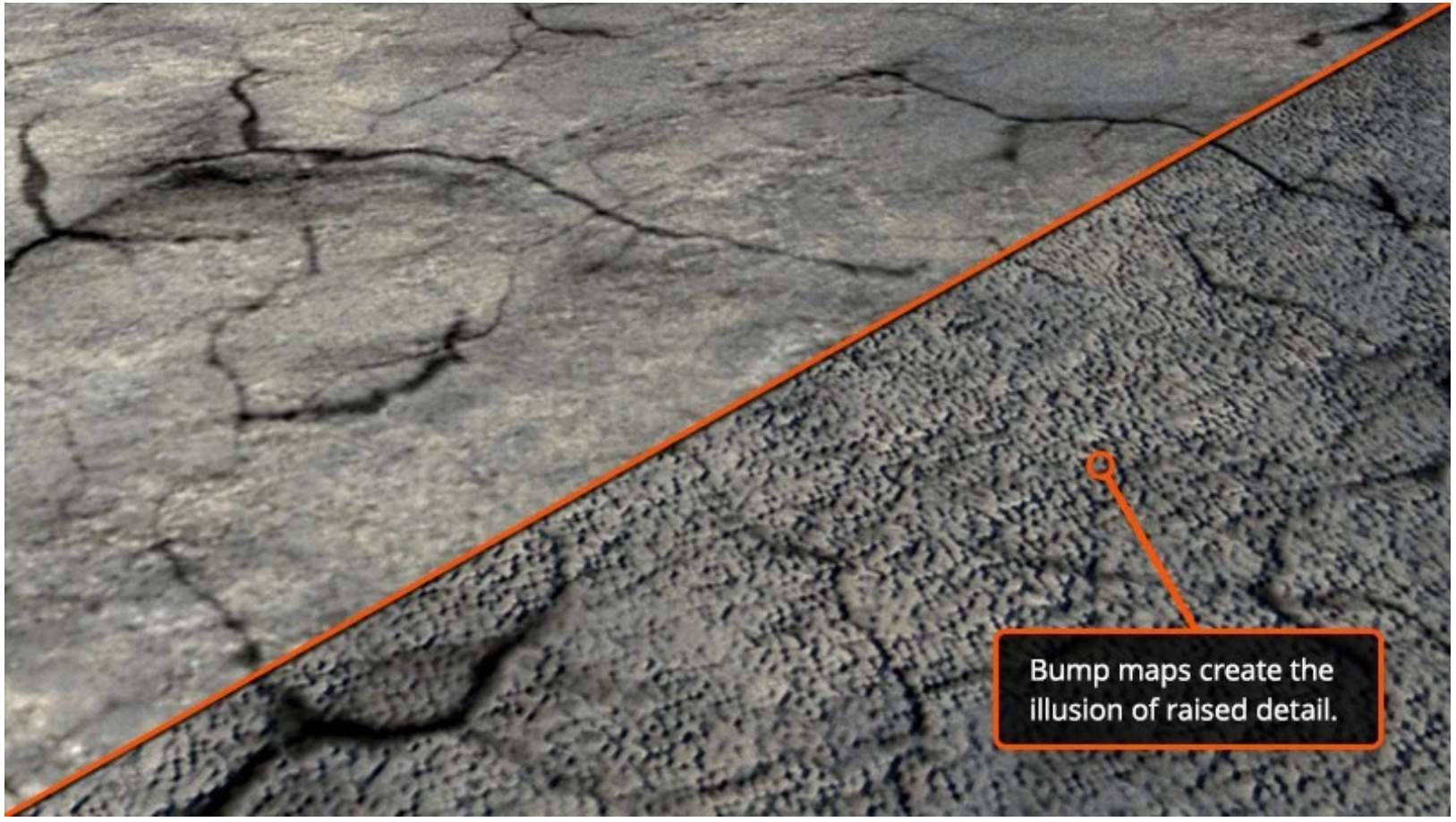




Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

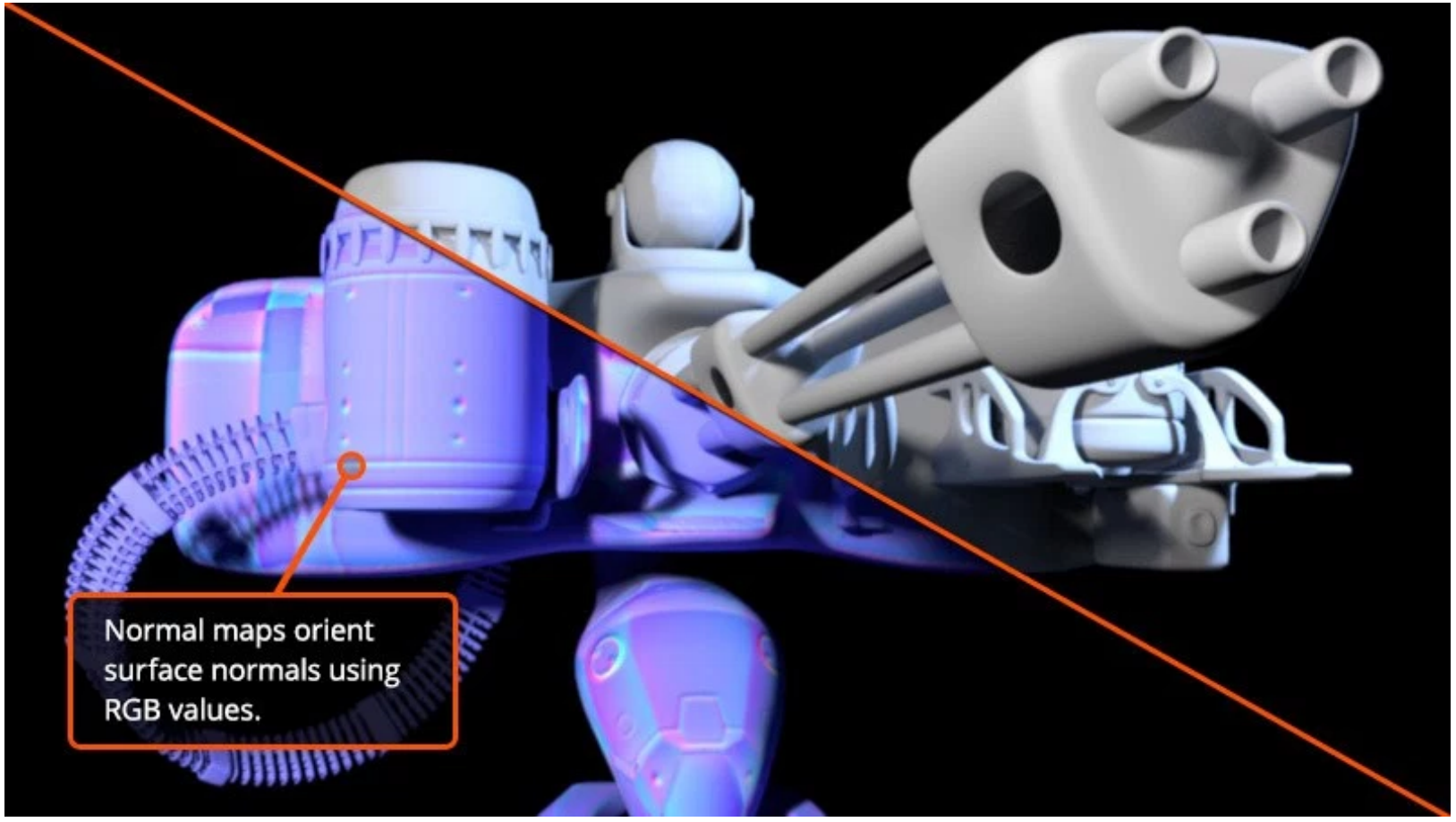
Bump Map (Summary)



http://blog.digitaltutors.com/bump-normal-and-displacement-maps/

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Normal Map (Summary)



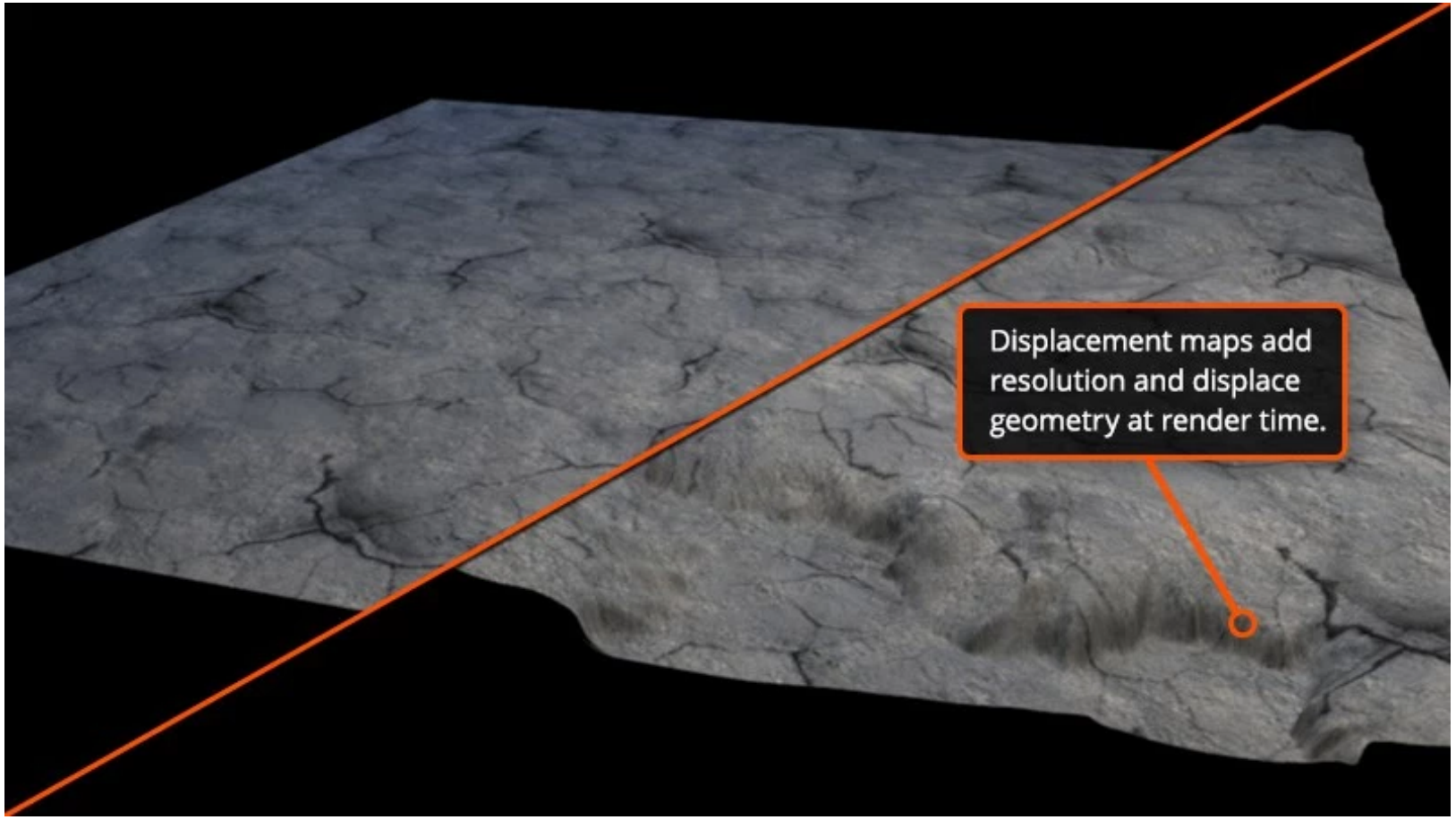
http://blog.digitaltutors.com/bump-normal-and-displacement-maps/

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Next step?

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

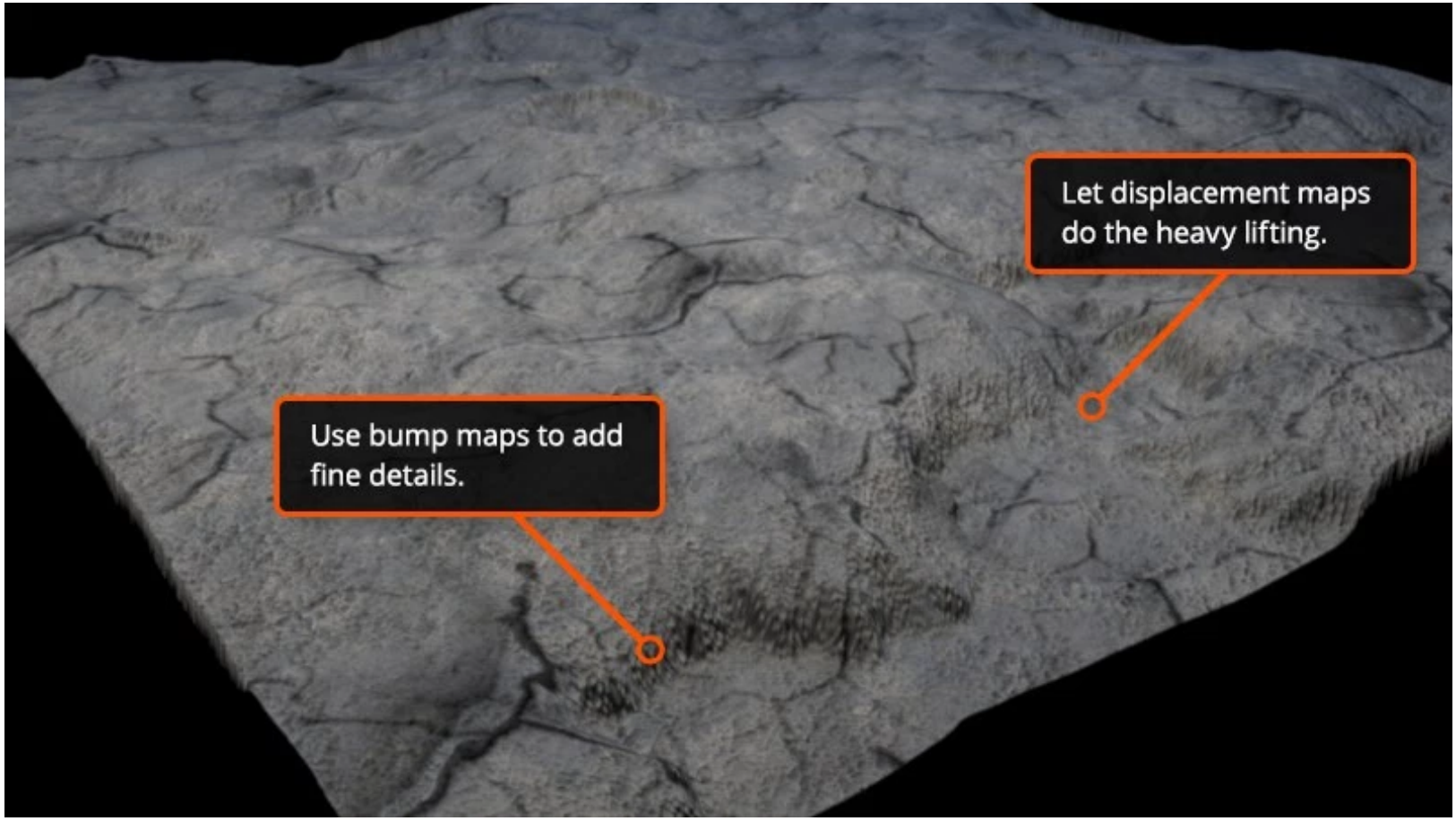
Displacement Maps (no fake geometry)



http://blog.digitaltutors.com/bump-normal-and-displacement-maps/

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

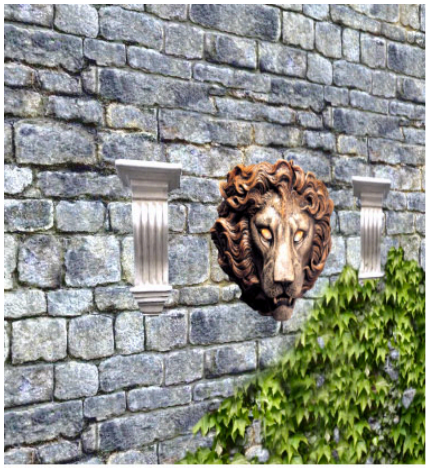
Combining maps for realistic details



http://blog.digitaltutors.com/bump-normal-and-displacement-maps/

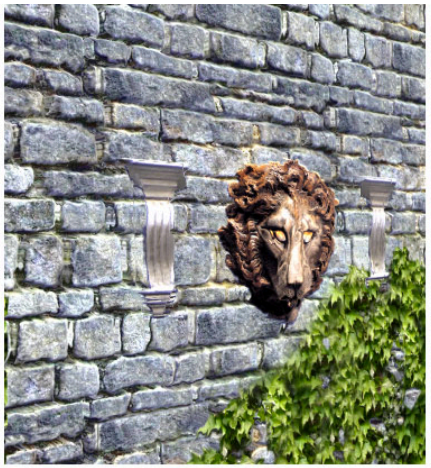
DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Comparison



Texture Mapped

Normal Mapped



Parallax Mapped

Steep Parallax Mapped

DigiPen
INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

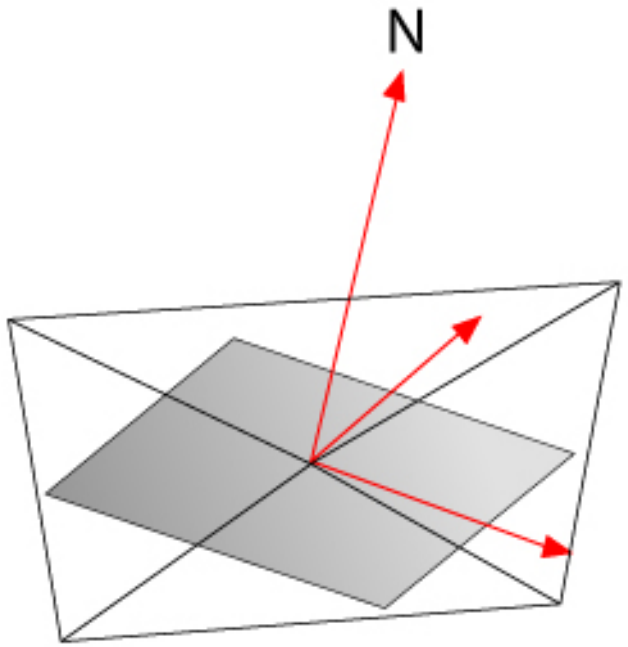
III. Tangent Space

- Given a vertex V having a normal N, we need to construct a coordinate system at that vertex such that the value (0,0,1) taken from a normal map corresponds to the normal N.
- This can be achieved by constructing a coordinate system in which the vertex normal always points along the positive z-axis.
- In addition to the normal vector, we need two vectors parallel to the surface and aligned to the texture space of the normal map.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- The resulting coordinate system is called tangent space and is shown in the figure below:



Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- Once the tangent-space coordinate system is established at each vertex of a triangle, the light vector L is computed and transformed into tangent space.

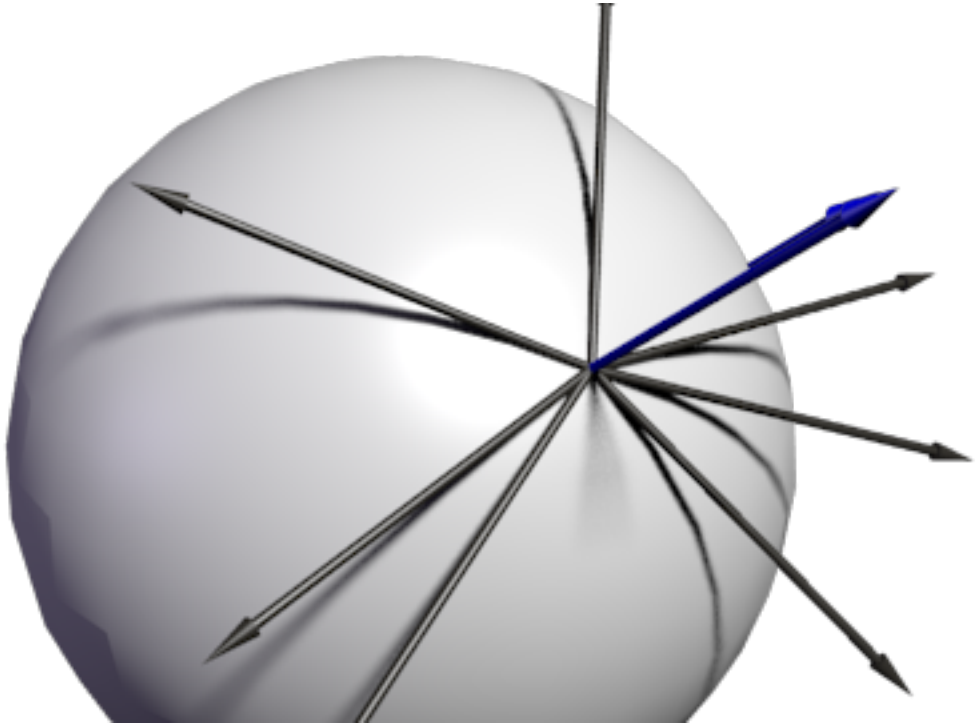
- Afterwards all what we need to do is to interpolate the vector across the surface and get the result of N.L where N represents the normal at the pixel fetched from the normal map.

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

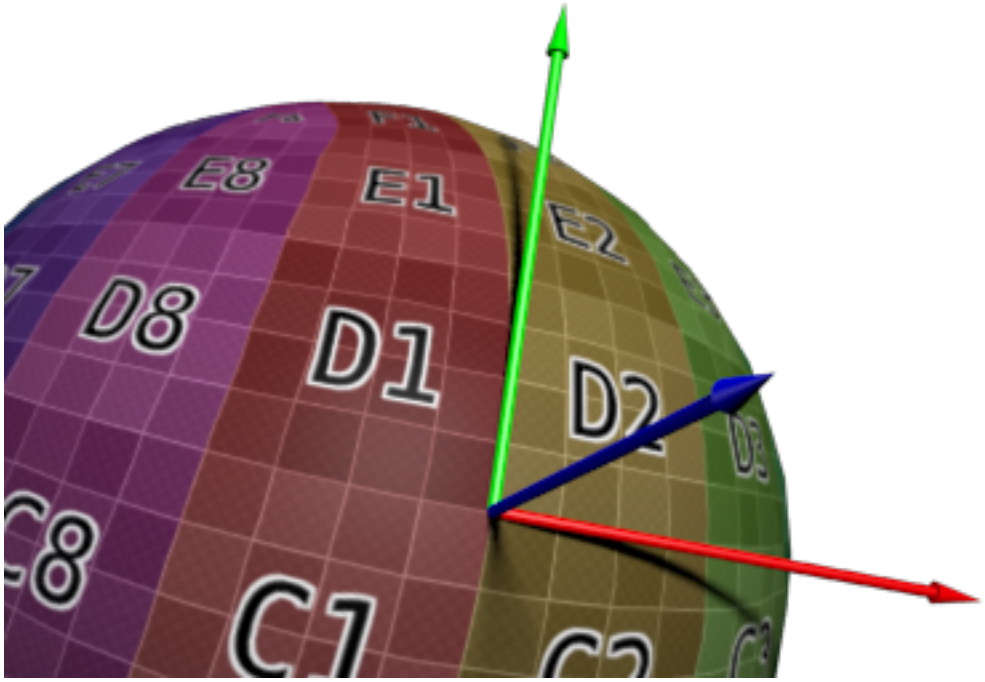
III. Tangent Space *(cont'd)*

- Our tangent space is formed by the three axes described by the vectors:
  - N is the normal to the surface
  - T (tangent) is parallel to the surface and in the direction of increasing s (or u)
  - B (bitangent) is parallel to the surface and in the direction of increasing t (or v).

DigiPen
INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved



DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved



DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- Any vector V (s, t) belonging to the face can be written as:

$$V = sT + tB$$

where (s, t) are the texture coordinates describing the vector

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- Given a triangle's vertices and texture coordinates, our goal is to find the vectors T and B in order to build our tangent space.
- Consider a triangle having as vertices E, F and G.
- The texture coordinates corresponding to those vertices are (s_e, t_e), (s_f, t_f) and (s_g, t_g) respectively.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- Let P and Q be the vectors formed by:

$$P = F - E$$

$$Q = G - E$$

and

$$(s_1, t_1) = (s_f - s_e, t_f - t_e)$$

$$(s_2, t_2) = (s_g - s_e, t_g - t_e)$$

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- We need to solve the following equations for T and B:

$$P = s_1 T + t_1 B$$
$$Q = s_2 T + t_2 B$$

- This is a linear system with six unknowns and six equations (corresponding to the x, y and z coordinates of T and B).

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- Solving the system will give us the tangent vector T and bitangent vector B corresponding to the triangle EFG.

- To find the tangent vector to a single vertex, we need to average the tangents of all the triangles sharing that vertex.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- The resulting tangent and bitangent are most likely not unit vectors and they should NOT be normalized as they contain the scaling (bumpiness) information.

- However, the normal vectors are typically normalized to ensure the height of the bump surface across the object surface is uniform.

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- The following matrix will transform a point in a tangent space back to the model space.

$$\begin{bmatrix} T_x & B_x & N_x \\ T_y & B_y & N_y \\ T_z & B_z & N_z \end{bmatrix}$$

- Use its inverse to transform from the model space to the tangent space

DigiPen INSTITUTE OF TECHNOLOGY
Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

III. Tangent Space *(cont'd)*

- Keep in mind that the light vectors are described in view space and the calculated T and B are in model space.
- So, transform T and B to view space before constructing the matrix. This way, the matrix will transform the sampled normal directly to view space.

Copyright © 2011, DigiPen Institute of Technology. All Rights Reserved

Thank You.