Computer Graphics, Task 5 - Texturing

Paweł Aszklar
P.Aszklar@mini.pw.edu.pl

Warsaw, May 31, 2016

1 Guidelines

Before going any further, make sure to read basic triangle mesh drawing guidelines available here. The following document is based on information provided there.

1.1 Scene modeling

For each textured mesh we need to select a texture, which is a two-dimensional image, read from file, which will be overlaid over the faces of the mesh. Additionally in each vertex of the mesh we need to define texture coordinates, corresponding to a certain point on the texture image. Texture coordinates of triangle vertices define the section of the texture image the triangle will be filled with. Texture coordinates defined in this document will be independent of the texture image size, and will be expressed as real numbers in range of $[0, 1]$. Those should be mapped to actual pixel coordinates so that $(0, 0)$ maps to the top-left and $(1, 1)$ maps to bottom-right corner of the image.

As a result each vertex (besides its position $\mathbf{p}$ and normal vector $\mathbf{n}$ defined in local coordinate system) will have an additional texture coordinates attribute t:

$$\mathrm{V} = (\mathbf{p}, \mathbf{n}, \mathbf{t})$$

$$\mathbf{t} = \begin{bmatrix} t_u \\ t_v \end{bmatrix}$$

Next we will show how to modify the previously described meshes of basic solids to include texture coordinates. Only texture coordinates for each vertex will be given, positions and normal vectors do not change.

1.1.1 Cuboid mesh

Let us assume that we have a texture image divided into six parts. Each part will cover one side of the cuboid, as shown in figure 1.

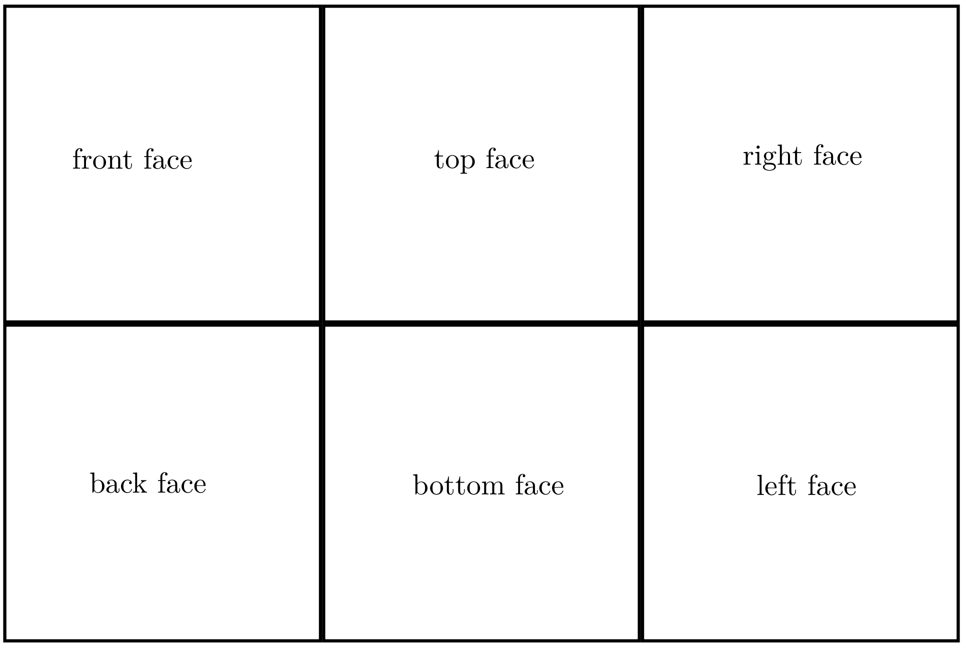

Figure 1: Cuboid texture

Texture coordinates of the front face vertices are as follows:

$$\mathbf{t}_1 = \left[0, \frac{1}{2}\right]^T$$

$$\mathbf{t}_2 = \left[\frac{1}{3}, \frac{1}{2}\right]^T$$

$$\mathbf{t}_3 = \left[\frac{1}{3}, 1\right]^T$$

$$\mathbf{t}_4 = \left[\frac{1}{3}, 1\right]^T$$

Texture coordinates for the remaining five faces can be defined analogously.

1.1.2 Cylinder texture

Let us assume that the cylinder texture is divides as shown in figure 2.

Top half of the image will be wrapped around the cylinder side. In the bottom half we have two circular regions which should fill the top and bottom base.

Texture coordinates for the top base vertices:

$$\mathbf{t}_0 = \left[\frac{1}{4}, \frac{1}{4}\right]^T$$

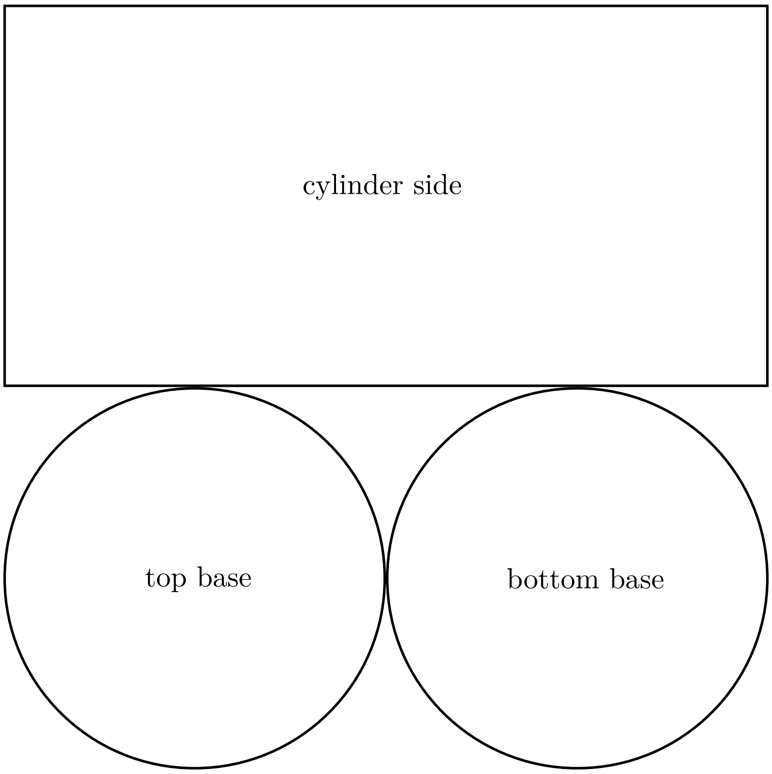

Figure 2: Cylinder texture

$$\mathbf{t}_i = \left[\frac{1}{4}\left(1+\cos\left(\frac{2\pi}{n}(i-1)\right)\right), \frac{1}{4}\left(1+\sin\left(\frac{2\pi}{n}(i-1)\right)\right)\right],\ i=1,\ldots,n$$

for the bottom base vertices:

$$\mathbf{t}_{4n+1} = \left[\frac{3}{4}, \frac{1}{4}\right]^T$$

$$\mathbf{t}_i = \left[\frac{1}{4}\left(3+\cos\left(\frac{2\pi}{n}(i-1)\right)\right), \frac{1}{4}\left(1+\sin\left(\frac{2\pi}{n}(i-1)\right)\right)\right],\ i=3n+1,\ldots,4n$$

for vertices of the cylinder side along the top base:

$$\mathbf{t}_{i+n} = \left[\frac{i-1}{n-1}, 1\right],\ i=1,\ldots,n$$

for vertices of the cylinder side along the bottom base:

$$\mathbf{t}_{i+2n} = \left[\frac{i-1}{n-1}, \frac{1}{2}\right],\ i=1,\ldots,n$$

1.1.3 Siatka sfery

In case of a sphere we will try to wrap the entire texture image along the equator, without subdividing the texture into separate fragments. Texture coordinates of the vertices will create an uniform grid on the texture image. The only two exceptions will be top and bottom pole. Since we can only store one set of texture coordinates in each, we will point them to the middle of top and bottom edge of the image.

Texture coordinates for the top and bottom pole:

$$\mathbf{t}_0 = \left[1, \frac{1}{2}\right]$$

$$\mathbf{t}_{mn+1} = \left[0, \frac{1}{2}\right]$$

for all other vertices

$$\mathbf{t}_{im+j} = \left[\frac{j-1}{m-1}, \frac{i}{n+1}\right],\ i = 1, \ldots, n;\ j = 1, \ldots, m$$

1.2 Scene drawing

Algorithm for drawing triangle meshes needs to be modified. Each vertex of a triangle contains now an additional attribute of texture coordinates. This attribute is not subject to any matrix transformations. However, whenever a triangle is clipped or filled, the value of texture coordinates along edges and along scanlines needs to be interpolated in the same manner as the vertex position $\mathbf{p}^G$ and normal vector $\mathbf{n}^G$ in 3D space.

Another modification is needed when calculating a color for a pixel. The interpolated texture coordinates should be used to select color of a pixel from the texture image. That color can be used as the color of the resulting pixel, or if Phong illumination model is used, can replace diffuse and ambient reflection coefficients of the surface material.