Discussion 7: Discrete Techniques

Discussion Topic:

How does texture mapping involve wrapping 2D images (textures) onto 3D surfaces, enhancing objects' visual detail and realism in a scene? Share your understanding of how texture mapping contributes to creating immersive virtual environments.

My Post:

Hello class,

In computer graphics, texture mapping is a technique used to apply 2D images to the surface of 3D models. This technique allows the addition of complex patterns, detailed color schemes, and surface characteristics to 3D models' surfaces without the overhead of implementing complex geometric to mimic intricate surfaces. Therefore, reducing computational cost while still enhancing the visual detail of objects and realism in scenes.

On a side note, a similar technique is applying 3D textures onto 3D models; this technique is usually referred to as volume texturing; however, this could also be called texture mapping. However, the main difference between the two techniques is that 2D texture mapping applies flat images onto the surface of a 3D model, while 3D texture mapping applies volumetric textures (3D Textures) throughout the 3D space of the model.

When using APIs such as WebGL, texture mapping is applied in several steps, such as creating a texture object and binding it to the 3D model, loading the texture image, assigning coordinates to the texture object, and then applying it to the object or model.

Below are examples of JavaScript and GLSL WebGL code that illustrate the steps mentioned above.

Step-1 Creating and binding the texture object. The texture object is created and stored in the GPU memory. WebGL uses functions like gl.createTexture() and gl.bindTexture() to create and bind the texture object.
JavaScript

```
var texture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_2D, texture);
```

Step-2 Loading the Texture Image. The image is loaded into RAM by storing it in an image object, usually, the image is a PNG or JPEG. Then the image data is transferred to the GPU memory.
JavaScript

```
var image = new Image();
image.src = 'texture.png';
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);
```

Step-3 Assigning Texture Coordinates (UV Mapping). In this step, the vertexes of the 3D model are assigned the texture telling the renderer what part of the texture corresponds to each point on the object's surface. The shape bellow is a quadrilateral

JavaScript

```javascript
var texCoords = [
    vec2(0, 0), // bottom-left corner
    vec2(0, 1), // top-left corner
    vec2(1, 1), // top-right corner
    vec2(1, 0)  // bottom-right corner
];
```

Step-4 Applying the Texture. This step is done in the fragment shader. The texture is applied to the surface of the object based on the interpolated texture coordinates.

```glsl
uniform sampler2D uTextureMap; // sampler for a 2D texture, holds the texture data

in vec2 vTexCoord; // Input texture coordinates passed from the vertex shader
                   // used to look up the color from the texture

// The final color.
out vec4 fragColor;

void main() {
   /*
     samples the texture at vTexCoord and
     assigns the resulting color to fragColor.
     The texture() function fetches the color
     from the uTextureMap.
   */
   fragColor = texture(uTextureMap, vTexCoord);
}
```

In a few lines of code, APIs such as WebGL allow programmers to implement texture mapping which enhances significantly the detail of object surfaces without the overhead of complex geometry. It enhances the realism of models by simulating details like wood grain, bricks, or fabric patterns with an image. For example, a plain, untextured cube would look flat, but the same cube with texture could appear to have detailed brick patterns, with color variations and even surface roughness.

Additionally, texture mapping allows for advanced effects such as environment mapping simulating reflections on shiny surfaces like metal or water. "Highly reflective surfaces are characterized by specular reflections that mirror the environment. [...] We can, however, use variants of texture mapping that can give approximate results that are visually acceptable through environment maps or reflection maps" (Angle & Shreiner, 2020, p197). Note that environment mapping is a technique that converts 3D environment information into a 2D texture format

Furthermore, bump mapping, a 2D mapping technique used to create the illusion of surface irregularities and roughness by altering the normal vectors of a surface during shading. It gives the illusion of depth and texture without modifying the geometry. Bump mapping will show changes in shading as the light source or object moves, making the object appear to have variations in surface smoothness by using a

grayscale texture (a bump map) to adjust the surface normals during lighting calculations (Angle & Shreiner, 2020).

To summarize, in computer graphics, texture mapping is a technique that enhances the visual realism of 3D objects by applying 2D images to their surfaces. Furthermore, advanced 2D texture mapping techniques like environment mapping and bump mapping allow for further intricate surface detail, all while keeping the computational costs low by avoiding additional geometric complexity. All of these can be achieved with just a few lines of code using abstraction layers provided by APIs such as WebGL.

-Alex

References:

Angel, E., & Shreiner, D. (2020). Chapter 7: Texture mapping. *Interactive computer graphics. 8th edition*. Pearson Education, Inc. ISBN: 9780135258262