ADDING TEXTURE TO A GRAPHIC

Jefwa ngombo nicholas

ngombonj@gmail.com

0722641884

Function of texture

- Adds shininess or natural appearance to the surface
- Affects the amount of light reflected from surface
- applied in texture mapped rendering technique or application
- An alternative rendering technique to photorealistic rendering which uses ray tracing(next session)

Texture representation

- 2D textures are represented as 2D images- array of pixels
- textures can store a variety of properties, i. e.
  - positions of texture pixels, i. e. texel
    - characterized by texture coordinates (u, v) in texture space
  - colors,
  - normals –direction vector for calculation of illumination

texture mapping

- Associating points on a graphic with texels on a texture
- texture mapping is a transformation from object space to texture space (x, y, z) (u, v)
- texture coordinates (u, v) are assigned to a vertex (x, y, z)
- texture mapping is generally applied per fragment

texture lookup

- rasterization determines fragment positions and interpolates texture coordinates from adjacent vertices
- texture lookup is performed per fragment using interpolated texture coordinates

mipmaps

A series of pre-filtered texture maps of varying resolutions

Used to assign texture to geometry objects in transformation such as scaling and viewing

Texturing pipeline

object space location

- texture coordinates are defined for object space locations
- if the object moves in world space, the texture moves along with it

projector function

- maps a 3D object space location to a (2D) parameter-space value

corresponder functions

- map from parameter space to texture space
- texture-space values are used to obtain values from a texture

value transform

a function that transforms obtained texture values

Texture mapping steps

1. Create a texture object and specify a texture for that object.
2. Indicate how the texture is to be applied to each pixel.
3. Enable texture mapping.
4. Draw the scene, supplying both texture and geometric coordinates.

Create a Texture Object and Specify a Texture for That Object

a texture object is created to maintain a texture.

Typically, textures are read from an image file, since specifying a texture programmatically could take hundreds of lines of code.

Header and variables definition part

```
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>
#include <stdlib.h>
#include <stdio.h>
/* Create checkerboard texture */
#define checkImageWidth 64
#define checkImageHeight 64
static GLubyte checkImage[checkImageHeight][checkImageWidth][4];
static GLuint texName;
```

Create texture

```
void makeCheckImage(void)
{
int i, j, c;
for (i = 0; i < checkImageHeight; i++) {
for (j = 0; j < checkImageWidth; j++) {
c = ((((i&0x8)==0)^((j&0x8))==0))*255;
checkImage[i][j][0] = (GLubyte) c;
checkImage[i][j][1] = (GLubyte) c;
checkImage[i][j][2] = (GLubyte) c;
checkImage[i][j][3] = (GLubyte) 255;}}}
```

Function to Generate texture

void init(void)

{glClearColor (0.0, 0.0, 0.0,
0.0);glShadeModel(GL_FLAT);glEnable(GL_DEPTH_TEST);

makeCheckImage();glPixelStorei(GL_UNPACK_ALIGNMENT, 1);//glPixelStorei *called because the data in the example isn't padded at the end of each texel row*

glGenTextures(1, &texName);glBindTexture(GL_TEXTURE_2D, texName);

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);//specify how the texture is to be wrapped and how the colors are to be filtered if there isn't an exact match between pixels in the texture and pixels on the screen

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);

glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MAG_FILTER,GL_NEAREST);

glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MIN_FILTER,GL_NEAREST);

glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth,

checkImageHeight, 0, GL_RGBA, GL_UNSIGNED_BYTE,checkImage);
//Defines a two-dimensional texture}

Function to apply texture as texture object

void display(void){glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);glEnable(GL_TEXTURE_2D);

glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);//sets the drawing mode to GL_DECAL so that the textured polygons are drawn using the colors from the texture map

glBindTexture(GL_TEXTURE_2D, texName) // references the texture textureName for use, both creates and uses texture objects;glBegin(GL_QUADS);

glTexCoord2f(0.0, 0.0); glVertex3f(-2.0, -1.0, 0.0);// associates texture coordiniates with vertex coodinates of geometry object that follow

glTexCoord2f(0.0, 1.0); glVertex3f(-2.0, 1.0, 0.0);

glTexCoord2f(1.0, 1.0); glVertex3f(0.0, 1.0, 0.0);

glTexCoord2f(1.0, 0.0); glVertex3f(0.0, -1.0, 0.0);

glTexCoord2f(0.0, 0.0); glVertex3f(1.0, -1.0, 0.0);

glTexCoord2f(0.0, 1.0); glVertex3f(1.0, 1.0, 0.0);

glTexCoord2f(1.0, 1.0); glVertex3f(2.41421, 1.0, -1.41421);

glTexCoord2f(1.0, 0.0); glVertex3f(2.41421, -1.0, -1.41421);