

Homework 5: Texture mapping

COSC6372 Interactive Computer Graphics (Spring 2023)

DUE: APRIL 13, 2023 AT 11:59 PM

1. Introduction

In this assignment, you will add the texture mapping functions to the Gz library. Note that we only consider the simple texture mapping without shading. So, you do not need to use the illumination in the previous assignment.

With your provided API functions, the main application will read series of triangles from the text file Tris.txt, load a texture from the image Texture.bmp, and draw the tea pot. This time, Tris.txt only provides you the coordinates of the vertices and the corresponding texture coordinates.

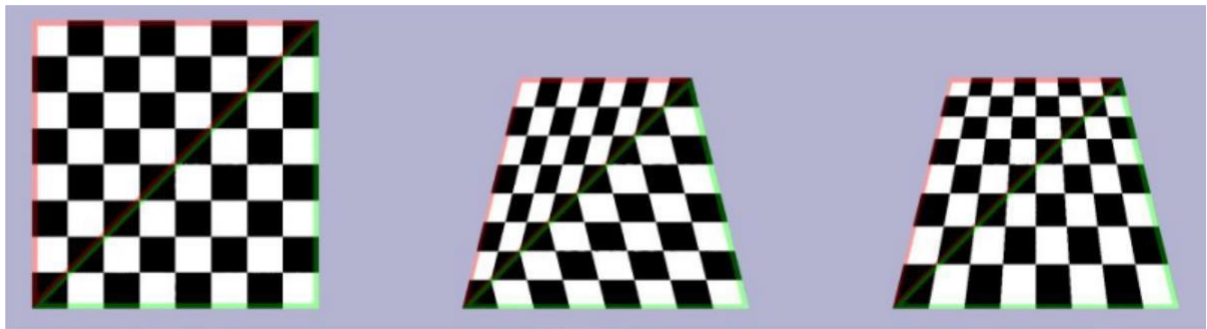
2. Files to update

File	Description
Gz.h	Updated to support texture mapping. Note that we do not support texture mapping with lights. You need to add some code to these files.
Gz.cpp	
GzFrameBuffer.h	Updated with the solution of HW4. You may put your rasterization with texture for HW5 in these files.
GzFrameBuffer.cpp	
Tris.txt	An input text file contains the list of triangles. You can figure out the format by reading the source code in file main.cpp or check the description bellow.
Texture.bmp	The texture file. In main.cpp, we use the class GzImage to load this texture.
TeaPot1.bmp	The sample bmp-format results. Note that you are supposed to generate a result looks like this file, but not exact pixel-by-pixel.
TeaPot2.bmp	

3. Note

Here are some details you may need to pay attention:

1. Since we do not use the lights, you can directly apply the color of the texture pixels for rendering. The only thing you need to figure out is for each rendered pixel, what is the corresponding one in the texture.
2. Your program needs to support both orthogonal projection and perspective projection. With the orthogonal projection, you can do the linear interpolation for texture coordinates as you did in previous assignments. However, for the perspective projection, you need to do the perspective correction to avoid the artifact likes following:



Texture

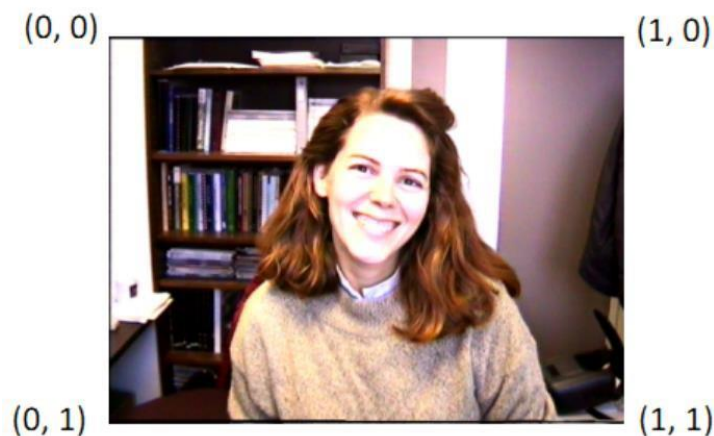
Artifact

Correct

Please read the follow link about how to do the perspective correction:

http://en.wikipedia.org/wiki/Texture_mapping#Perspective_correctness

3. The format of the Tris.txt has changed. After the first 3 numbers represent for vertex coordinate, we have 2 numbers u and v represent for texture coordinates. The ranges of u and v are $[0, 1]$. $(u, v) = (0, 0)$ corresponds to pixel $(0, 0)$ of the texture, $(u, v) = (1, 1)$ corresponds to pixel $(\text{width}-1, \text{height}-1)$ of the texture.



4. Requirements

1. Do the assignment independently.
2. You must submit all your source code, project files (MSVC project or Makefile), your results and your report. TA may test your implementation by changing some options, changing the source code of the main program or changing the data file.
3. You need to write a detailed report (50% points of the assignment, pdf format), you should state the assignment problem, explain the algorithm or method you use, explain details of implementation, discuss your results, etc.
4. Save your results as images.
5. Upload every necessary file to GitHub
6. In your GitHub readme file, put your name and student ID there, including the coding environment and compiling method (command) if necessary.

7. You can only use the libraries we provided.
8. You will lose points if violate any requirement above.