

GPU and GPGPU Programming

Shuai Lu 170742

Assignment 3

Task

1. Start by setting up a simple shader in order to see the preloaded scene on the screen (you should see 21 balloons).
2. Setup your framebuffers and implement the geometry rendering-pass.
You will need buffers for colors, normals, and depth. The depth should be normalized.
3. Setup the geometry (i.e., a rectangle) that fills the entire screen and render it in the second pass.
4. Implement at least four variations of the second render-pass.
 - 4.a) Compute phong shading in the second render-pass (you need to store light parameters as uniforms, and use the color and the normal from the buffers).
 - 4.b) Implement edge detection on the depth buffer and display the result.
 - 4.c) Implement blurring using a variable sized box (or mean) filter.
 - 4.d) Implement the Depth of Field effect.
The idea of the Depth of Field technique is to imitate the blurring of objects that are out of focus when using real cameras.
You can implement it by defining a focus plane (uniform variable in the range $[0..1]$). Compute the distance between the focus plane and the normalized depth value in the depth buffer (computed in the first pass). Use the blur filter implemented in 4.c) with a variable kernel size. Map zero distance to a kernel of size 1×1 , and the highest distances to a kernel of size 15×15 .
5. Provide key mappings to allow the user to switch between modes and to change the focal-plane for 4.d).
6. Submit your program and a report including screenshots of the results.

BONUS: - Implement Screen Space Ambient Occlusion.

Outcome:

I setup the simple transformation shaders. The outcome is in fig.1.

I setup the framebuffers and implement the geometry rendering-pass. I setup the quad that fills the entire screen and render it in the second pass.

I implement the Phong shader(Fig.2), the edge detection(Fig.3), the blurring(Fig.4), and the Depth of Field effect(Fig.5) in the second render-pass.

I also implement a key_callback function to allow the user to switch between modes and to change the focal-plane for 4.d).



Figure 1: Transformation

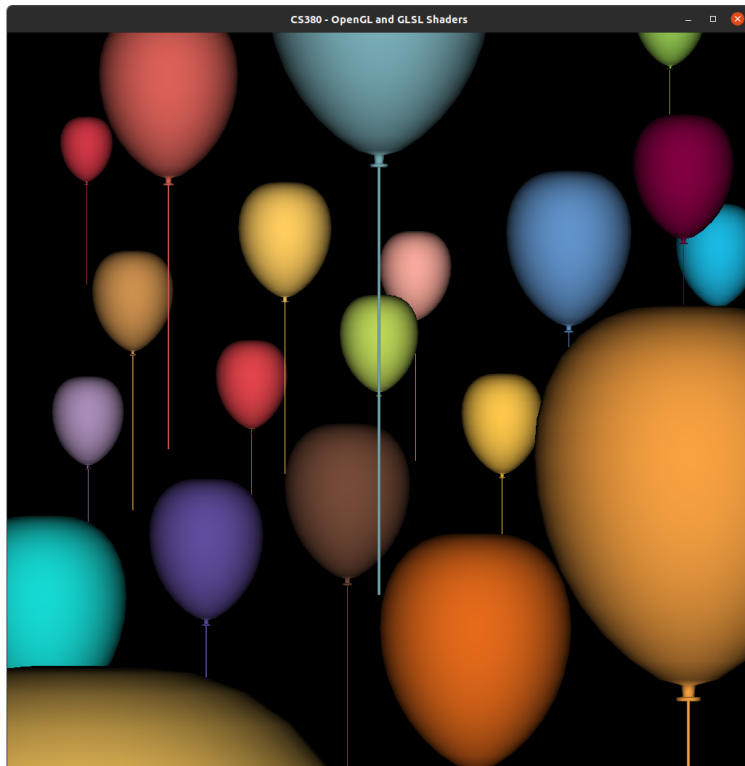


Figure 2: Phong shading

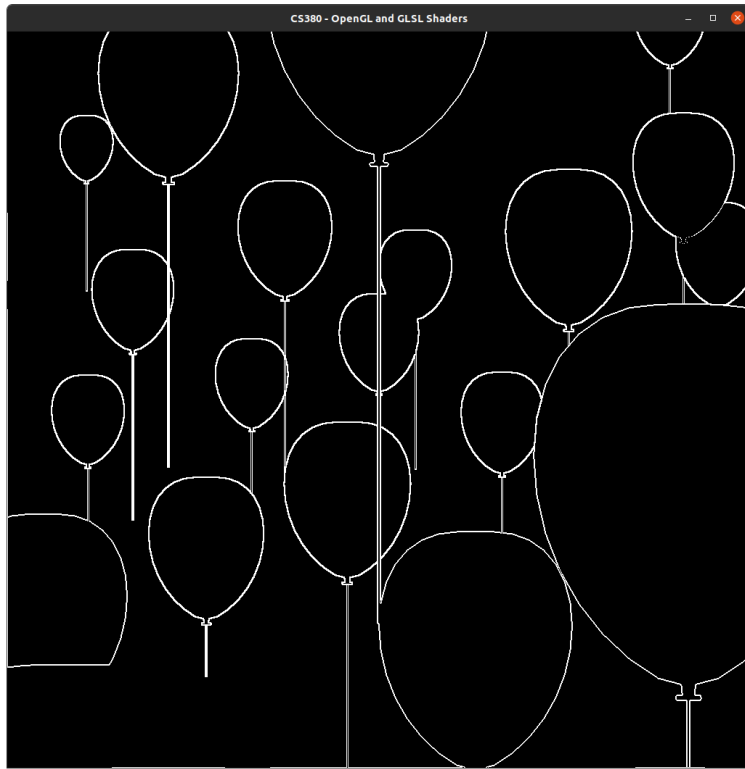


Figure 3: Edge detection



Figure 4: Blurring

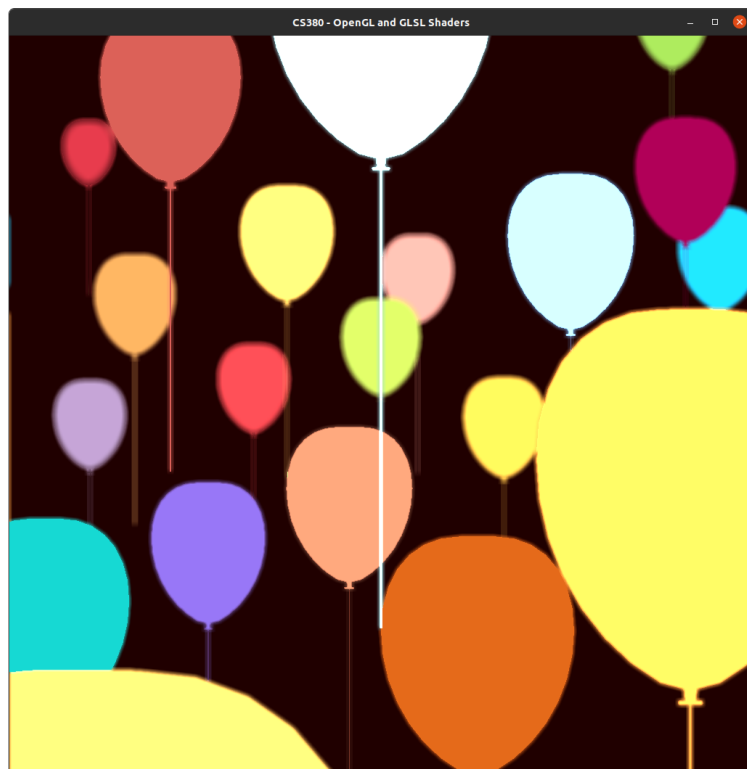


Figure 5: Depth of Field effect