

Interactive Computer Graphics (exercise sheet 3)

Assignment 3 [10 Points] Shadow Mapping

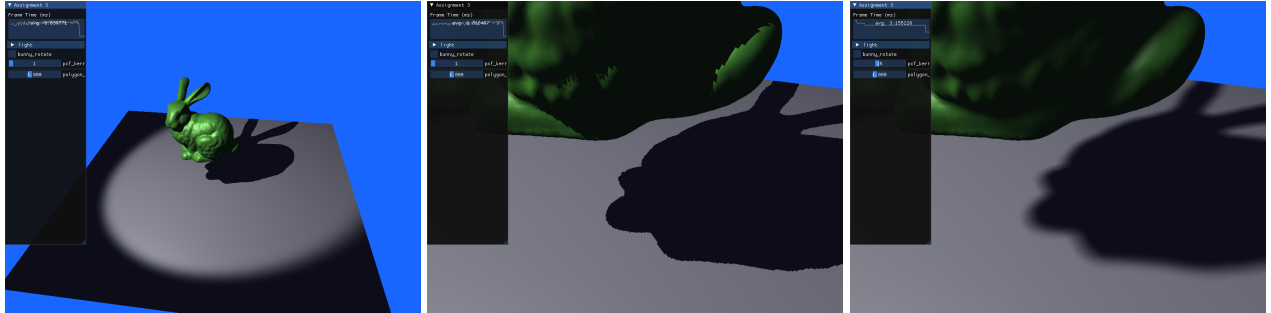


Figure 1: Left to right, bunny with shadow, closeup on hard shadow and closeup on pcf soft shadow.

When running the code provided for this assignment you should see a simple scene: a bunny illuminated by a spotlight. Your job will be to add shadows to this scene.

a) Basic Shadow Mapping [5 Points] Shadow mapping has been covered in great detail in the lecture: for any given point in the scene, if the depth value stored in the shadow map is smaller than the point's depth value (projected into light space), then the point is occluded by some other geometry and therefore not lit. Otherwise, the light source is visible from this point and we can apply diffuse and specular lighting.

The depth texture is almost set up by the code provided to you. However, there are still some texture parameters missing to enable the automatic comparison between the depth value sampled from the depth texture and the current fragment's depth. Additionally, you still need to bind the texture to the frame buffer object (FBO) to be able to render into it. To do so, complete the method `initShadowMap`.

The next step is to actually render to the depth texture. Do this by extending the method `renderShadowMap` as follows: bind the FBO, set the appropriate viewport size and render from the point of view of the light source. The code to bind the shadow-map afterwards is already there in method `render`. It does, however, need a small adjustment of the shadow projection matrix, so the additional transformation can be done using only matrices in the shader.

Now the only thing left is to use the shadow map in the fragment shader. Extend the vertex shader to compute the correct projection and the fragment shader to compute the value of `is_lit` (1.0 if visible from the light, 0.0 if shadowed).

- b) Shadow-Acne** [1 Points] The shadows on the bunny should now work, but there are still some artefacts on it, and the ground is mostly black. This problem is called Shadow-Acne. What is Shadow-Acne and how do you usually deal with it? Add a new element (e.g. Slider + variable in .hpp) to the GUI, to make your solution configurable. Use this variable to fix the problem.
- c) Percentage Closer Filtering** [4 Points] Extend your implementation by Percentage Closer Filtering (PCF) with a square kernel (side-length: `pcf_kernel_size` in pixels) centered at the projected coordinate. To do so, set `is_lit` to the percentage of lit pixels in the kernel. As a result, the edges of the shadow should become smooth (compare the center and right reference image on top).

Files to submit: `a03.cpp`, `a03.hpp`, `shader/phong.vs.glsl`, `shader/phong.fs.glsl`
The assignment is due until Wednesday, 25.06.2024, 11:55pm.

Happy Hacking! :)