

Steps for Hardware Shadow Mapping Using OpenGL

1. Create an empty depth texture
2. Set it up with an internal format of `GL_DEPTH_COMPONENT`
3. Set the `GL_DEPTH_TEXTURE_MODE` to `GL_LUMINANCE` (so it stores the depth internally with a single luminance value)
4. Enable the depth buffer
5. Render scene from the light
6. Copy the depth buffer into the texture using `glCopyTexImage2D(...)`
7. (Optional) Display texture to check that everything so far has worked
8. When we project the shadow map onto the scene, we need to compare the texture with the distance to the light we'll compute at each pixel. Tell OpenGL how to do this comparison by setting (using `glTexEnvf(...)`) the parameter `GL_TEXTURE_COMPARE_FUNC` to `GL_LEQUAL`.
9. Tell OpenGL what we want to compare on a per-pixel basis. Set `GL_TEXTURE_COMPARE_MODE` to `GL_COMPARE_R_TO_TEXTURE`.
10. Tell OpenGL what sort of texture generation to use:
 - `glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_EYE_LINEAR);`
 - `glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, GL_EYE_LINEAR);`
 - `glTexGeni(GL_R, GL_TEXTURE_GEN_MODE, GL_EYE_LINEAR);`
 - `glTexGeni(GL_Q, GL_TEXTURE_GEN_MODE, GL_EYE_LINEAR);`
11. Compute the matrix used to generate texture coordinates. This should be $\mathbf{SP}_{\text{light}}\mathbf{L}^{-1}$, where \mathbf{S} is the scale/bias matrix, $\mathbf{P}_{\text{light}}$ is the light's projection matrix (the `gluProjection(...)` matrix you used when rendering the light view), and \mathbf{L}^{-1} is the light's view matrix (the `gluLookAt(...)` matrix you used when rendering the light view).
12. Inside your display function *right after you call `gluLookAt(...)` for your eye's viewpoint*, setup your texture planes using:
 - `glTexGenfv(GL_S, GL_EYE_PLANE, plane_s);`
 - `glTexGenfv(GL_T, GL_EYE_PLANE, plane_t);`
 - `glTexGenfv(GL_R, GL_EYE_PLANE, plane_r);`
 - `glTexGenfv(GL_Q, GL_EYE_PLANE, plane_q);`Define `GLfloat plane_s[4], plane_t[4], plane_r[4], plane_q[4];` and initialize the planes as
$$\mathbf{SP}_{\text{light}}\mathbf{L}^{-1} = \begin{pmatrix} \text{plane_s}[0] & \text{plane_s}[1] & \text{plane_s}[2] & \text{plane_s}[3] \\ \text{plane_t}[0] & \text{plane_t}[1] & \text{plane_t}[2] & \text{plane_t}[3] \\ \text{plane_r}[0] & \text{plane_r}[1] & \text{plane_r}[2] & \text{plane_r}[3] \\ \text{plane_q}[0] & \text{plane_q}[1] & \text{plane_q}[2] & \text{plane_q}[3] \end{pmatrix} \quad (1)$$
13. Enable texture generation for all four texture coordinates:
 - `glEnable(GL_TEXTURE_GEN_S);`
 - `glEnable(GL_TEXTURE_GEN_T);`
 - `glEnable(GL_TEXTURE_GEN_R);`
 - `glEnable(GL_TEXTURE_GEN_Q);`