Projective Texture Mapping
Projective Textures

- Use to simulate effects:
  - Slide projectors
  - Spotlight illuminations
  - Shadows
  - Reprojection of images
Texture Projection

- Very similar to projecting the objects to the screen
- But now we will project the scene to the light source (or the slide projector, for example)
- The key is to calculate the texture coordinates of the projection receiver
Calculate the Texture Coordinates

- Main idea – Project the objects to the “image plane” of the projector and use the projector’s NDC to calculate the texture coordinates
Calculate the Texture Coordinates

- Use OpenGL’s texgen function and set it to GL_OBJECT_LINEAR
- Construct the texture matrix as a concatenation of \( M_t = T \times P \times M \)
  - M: Modelview (set the projector’s view using gluLookAt)
  - P: Projection (set the projector’s “camera” parameters)
  - T: Bias and Scale: transform from projector’s NDC \([-1,1]\) to \([0,1]\) to look up texture
Things to pay attention:

- Need to make sure that only the objects meant to receive the projected texture will render with the texture (not every object in the scene)
- Need to use 4D texture coordinates to ensure correct perspective projection
Example Code

Note that in this example we are using GL_EYE_LINEAR, but it is okay
As long as you remember to transform the eye space back to object space first

```c
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, (int) GL_EYE_LINEAR);
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, (int) GL_EYE_LINEAR);
glTexGeni(GL_R, GL_TEXTURE_GEN_MODE, (int) GL_EYE_LINEAR);
glTexGeni(GL_Q, GL_TEXTURE_GEN_MODE, (int) GL_EYE_LINEAR);
// These calls initialize the TEXTURE_MAPPING function to identity. We will be using
// the Texture matrix stack to establish this mapping indirectly.
float [] eyePlaneS = { 1.0f, 0.0f, 0.0f, 0.0f };
float [] eyePlaneT = { 0.0f, 1.0f, 0.0f, 0.0f };
float [] eyePlaneR = { 0.0f, 0.0f, 1.0f, 0.0f };
float [] eyePlaneQ = { 0.0f, 0.0f, 0.0f, 1.0f };
glTexGenfv(GL_S, GL_EYE_PLANE, eyePlaneS);
glTexGenfv(GL_T, GL_EYE_PLANE, eyePlaneT);
glTexGenfv(GL_R, GL_EYE_PLANE, eyePlaneR);
glTexGenfv(GL_Q, GL_EYE_PLANE, eyePlaneQ);
```

(next page to continue)