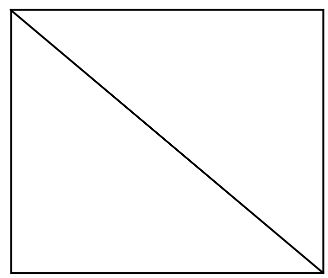
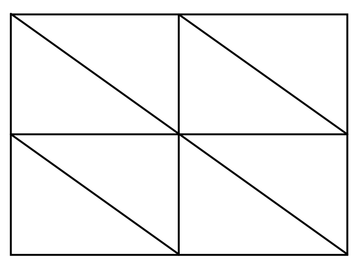
CS 370 - Assignment 3

1. For large polygons (particularly flat ones), the system will perform simple tessellation resulting in large fragments (which will then be shaded via interpolation of the normals between the original vertices), e.g.

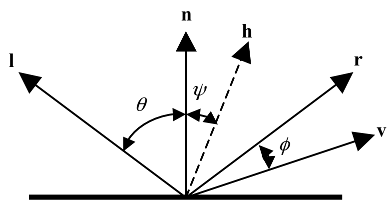


In order to force the system to more accurately perform shading, the application can divide the surface up into smaller polygons (e.g. using recursive subdivision) such that the tessellation will produce smaller fragments, e.g.



In order to deal with obscured light sources (i.e. blocked from reaching one object because of another) and also light reflected from other objects, we would have to keep track of **global** information about the position and characteristics of all the objects in the scene and then simultaneously solve for **all** lighting effects at once. This does not fit with the pipeline architecture of the graphics system which processes each object independently (i.e. no interaction of objects other than possibly clipping via depth buffering.) Techniques to perform this type of more realistic lighting include ray-tracing and radiosity which are performed at the application level rather than in hardware.

2. Using the diagram



the angle between $\mathbf{l}$ and $\mathbf{v}$ (i.e. between the light source and viewer) is given by (note that the angle between $\mathbf{l} \cdot \mathbf{r} = 2\theta$ by angle of incidence equals angle of reflection)

$$\mathbf{l} \cdot \mathbf{v} = \cos(2\theta + \phi)$$

By the definition of the half angle (i.e. half the angle between $\mathbf{l}$ and $\mathbf{v}$)

$$\mathbf{l} \cdot \mathbf{v} = \cos(2(\theta + \psi))$$

Equating these two relationships gives

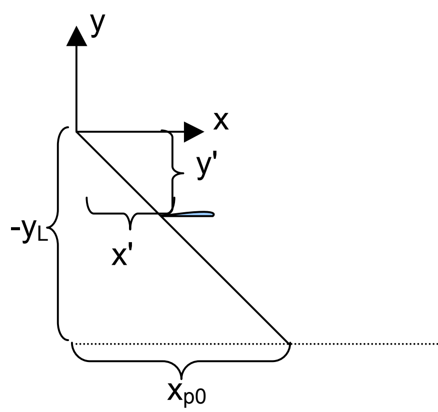
$$\begin{aligned} 2\theta + \phi &= 2(\theta + \psi) = 2\theta + 2\psi \\ \Rightarrow \phi &= 2\psi \end{aligned}$$

This angle avoids the need to compute $\mathbf{r}$ in the specular term $(\mathbf{r} \cdot \mathbf{v}) = \cos(\phi)$, instead replacing it with $(\mathbf{n} \cdot \mathbf{h}) = \cos(\psi) = \cos(\frac{\phi}{2})$. However if $\mathbf{v}$ is not in the same plane, then the computation only gives the angle of the projection which will be less than the true angle (which can be partially compensated for via adjusting the exponent e in the specular term.)

3. We first note that the projection plane becomes

$$y_{p_0} = -y_L$$

then looking at the side view, i.e. looking down the *z*-axis, gives



Using similar triangles we have

$$\frac{x'}{x_{p_0}} = \frac{y'}{-y_L} \quad \Rightarrow \quad x_{p_0} = \frac{x'}{y'/-y_L}$$

Similarly from a front view, i.e. looking down the *x*-axis, the *z* component is given by

$$z_{p_0} = \frac{z'}{y'/-y_L}$$

We will then define the *homogeneous* coordinates of the above terms (recalling that the *w* homogeneous component *divides* the other coordinates)

$$x_s = x' \quad y_s = y' \quad z_s = z' \quad w_s = \frac{y'}{-y_L}$$

Then the *matrix* that converts the original *homogeneous* vertex $(x', y', z', 1)$ to the *projected homogeneous shadow* vertex (x_s, y_s, z_s, w_s) is

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ w_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{y_L} & 0 & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

and thus the final coordinates are

$$x_{p_0} = \frac{x_s}{w_s} \quad y_{p_0} = \frac{y_s}{w_s} \quad z_{p_0} = \frac{z_s}{w_s}$$

The initial translation of the light source to the origin is given by

$$x' = x - x_L \quad y' = y - y_L \quad z' = z - z_L$$

and so final translation of the light source back to the original position is given by

$$x_p = x_{p_0} + x_L = \frac{x_s}{w_s} + x_L = \frac{x'}{y'/-y_L} + x_L = x_L - \frac{x - x_L}{(y - y_L)/y_L}$$

$$y_p = y_{p_0} + y_L = \frac{y_s}{w_s} + y_L = \frac{y'}{y'/-y_L} + y_L = y_L - y_L = 0$$

$$z_p = z_{p_0} + z_L = \frac{z_s}{w_s} + z_L = \frac{z'}{y'/-y_L} + z_L = z_L - \frac{z - z_L}{(y - y_L)/y_L}$$