Casting (fake) shadows in X3D

Giorgio Bacci

Department of Mathematics and Computer Science
University of Udine, Italy

FIT Summer School

Novi Sad, 26 June 2009

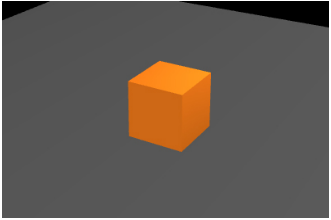
Presentation outline

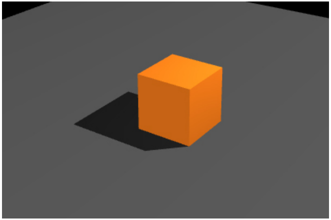
- Shadows on plane $y = 0$ **(simple)**
- Shadows on a general plane **(not so simple...)**
- Example in Mathematica
- ...finally in **X3D**

On the importance of shadows. . .

Shadows are important for several reasons:

- they are useful when the weather is hot!
- they convey additional spatial information to the 3D scene





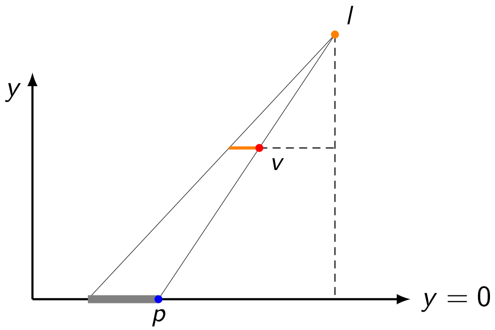
Loss of shadows. . .

In computer graphics objects are often rendered without shadows (because it costs!)

Usual 3D graphics rasterization-based pipeline does not have a stage where shadows are computed. . .

Shadows are computed separately as an optional rendering step!

Casting shadows onto the $y = 0$ plane



from the similar triangles...

$$\frac{p_x - l_x}{v_x - l_x} = \frac{l_y}{l_y - v_y}$$

$$\frac{p_z - l_z}{v_z - l_z} = \frac{l_y}{l_y - v_y}$$

hence: $$p_x = \frac{l_y v_x - l_x v_y}{l_y - v_y} \qquad p_z = \frac{l_y v_z - l_z v_y}{l_y - v_y}$$

Homogeneous projection matrix (plane $y = 0$)

The previous two equations can be converted into the **homogeneous** projection matrix:

$$\mathbf{M} = \begin{pmatrix} l_y & -l_x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -l_z & l_y & 0 \\ 0 & -1 & 0 & l_y \end{pmatrix}$$

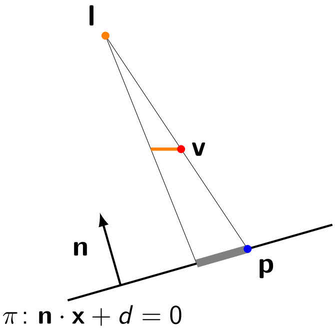
Casting shadow onto a general plane π

The problem can be generalized:

instead of using the $y = 0$ plane we cast shadows onto a plane

$$\pi : \mathbf{n} \cdot \mathbf{x} + d = 0$$

Casting shadow onto a general plane π (continued)



In order to find the projection $\mathbf{p}$ we have to solve this simple system of equations

$$\begin{cases} \mathbf{l} + \lambda(\mathbf{l} - \mathbf{v}) = \mathbf{p} \\ \mathbf{n} \cdot \mathbf{p} + d = 0 \end{cases}$$

i.e. the intersection between π and the line from $\mathbf{l}$ along $\mathbf{v}$.

Note: this approach works for any dimension n of the space $\mathbb{R}^n$

Casting shadow onto a general plane π (continued)

Substituting $\mathbf{p}$ with $\mathbf{l} + \lambda(\mathbf{l} - \mathbf{v})$ in $\mathbf{n} \cdot \mathbf{p} + d = 0$ we can recover λ

$$\mathbf{n} \cdot (\mathbf{l} + \lambda(\mathbf{l} - \mathbf{v})) + d = 0$$

$$\mathbf{n} \cdot \mathbf{l} + \lambda(\mathbf{n} \cdot (\mathbf{l} - \mathbf{v})) + d = 0$$

$$\lambda = -\frac{\mathbf{n} \cdot \mathbf{l} + d}{\mathbf{n} \cdot (\mathbf{l} - \mathbf{v})}$$

hence, from $\mathbf{p} = \mathbf{l} + \lambda(\mathbf{l} - \mathbf{v})$ we have

$$\mathbf{p} = \mathbf{l} - \frac{\mathbf{n} \cdot \mathbf{l} + d}{\mathbf{n} \cdot (\mathbf{l} - \mathbf{v})}(\mathbf{l} - \mathbf{v})$$

Homogeneous projection matrix (plane $\pi : \mathbf{n} \cdot \mathbf{x} + d = 0$)

This induces a generalization of
the **homogeneous** projection matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{n} \cdot \mathbf{l} + d - l_x n_x & -l_x n_y & -l_x n_z & l_x d \\ -l_y n_x & \mathbf{n} \cdot \mathbf{l} + d - l_y n_y & -l_y n_z & -l_y d \\ -l_z n_x & -l_z n_y & \mathbf{n} \cdot \mathbf{l} + d - l_z n_z & -l_z d \\ -n_x & -n_x & -n_z & \mathbf{n} \cdot \mathbf{l} \end{pmatrix}$$

Note: if $\mathbf{n} = (0, 1, 0)^T$ and $d = 0$ (plane $y = 0$) this matrix turns into the homogeneous matrix given before

Mathematica

X3D

Some remarks. . .

This projection technique works only for ideal infinite planes.

Less ingenuous rendering algorithms draw the shadow polygons using switching on/off the **Z-buffer** algorithm.