lecture 4

projections

- orthographic
- parallel
- perspective + vanishing points

view volume (frustum)

Orthographic projection

How to map 3D scene point (say in camera coordinates) to a 2D image point?

The simplest method: just drop the z coordinate.

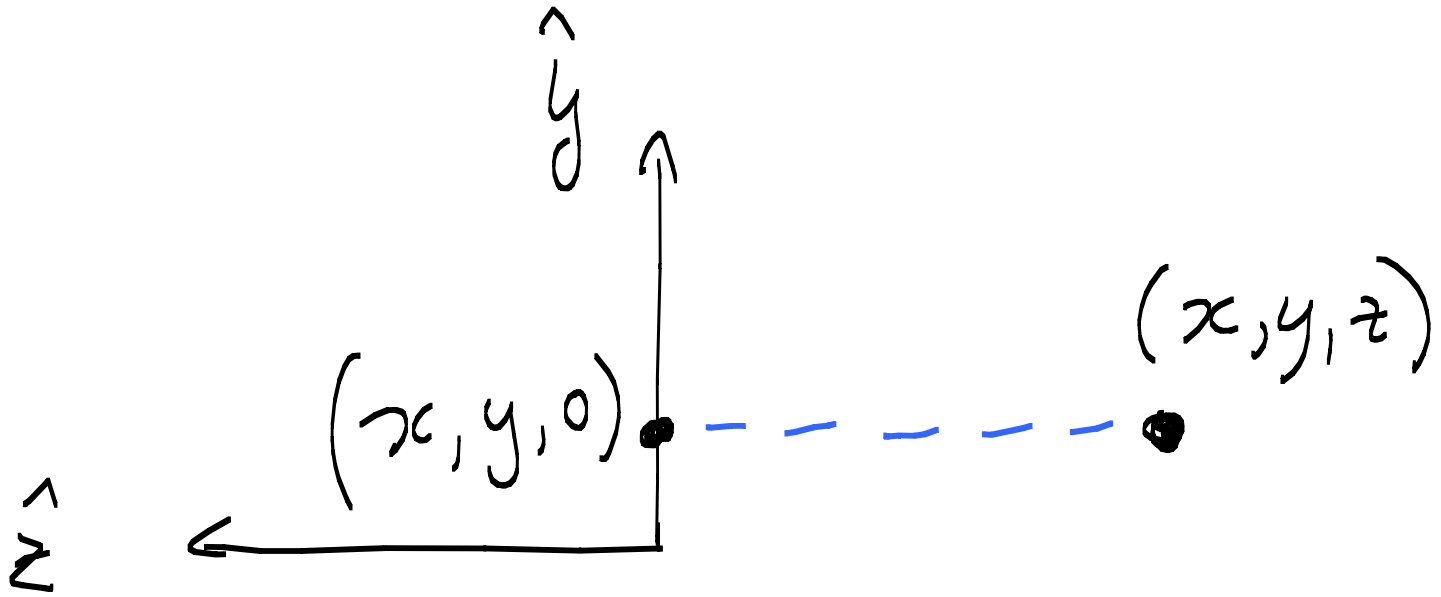
$$(x, y, z) \rightarrow (x, y)$$

Similar method: project to z=0 plane.

$$(x, y, z) \rightarrow (x, y, 0)$$

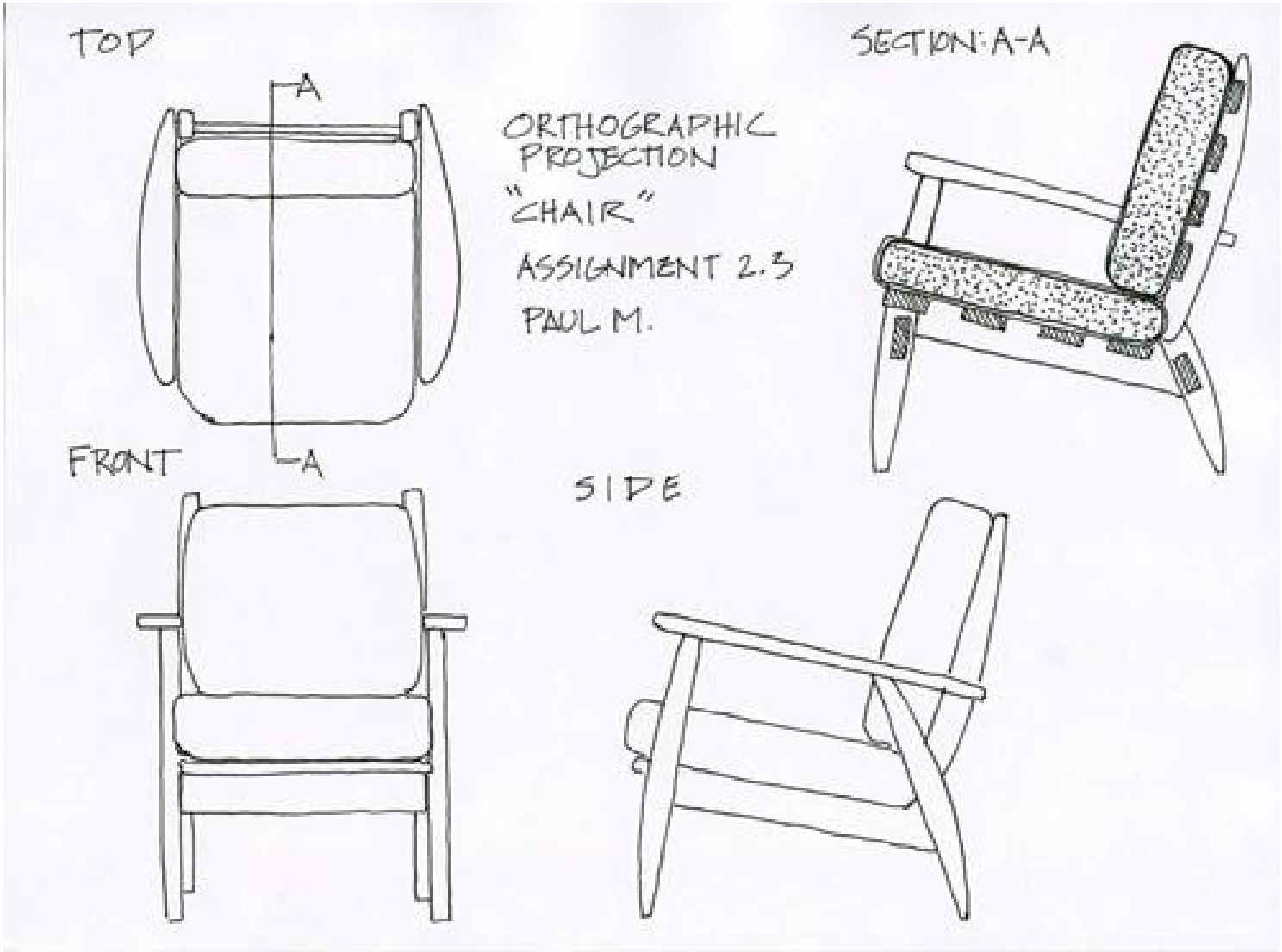
Orthographic projection to z=0 plane

$$\begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



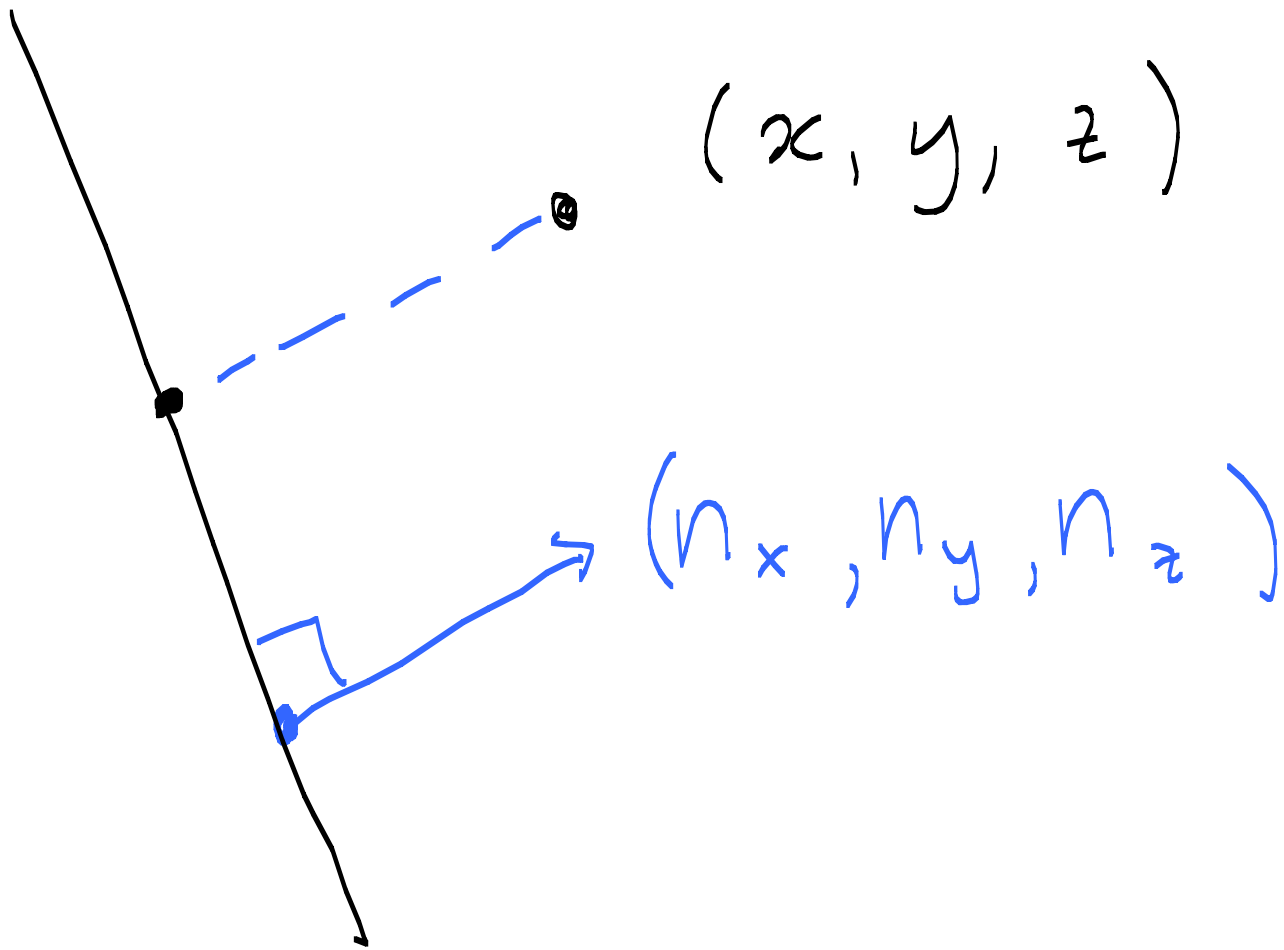
Orthographic projection can be in any direction.

Example: x (side), y (top), z (front)



Orthographic projection (in general) :

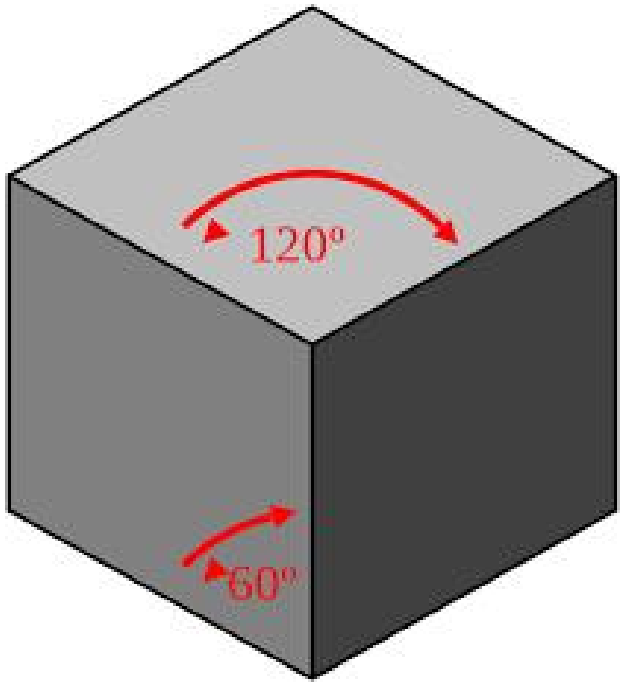
Project onto a plane, and in a direction of the plane's normal (i.e. perpendicular to plane)



Isometric projection:

orthographic projection onto $x + y + z = 0$.

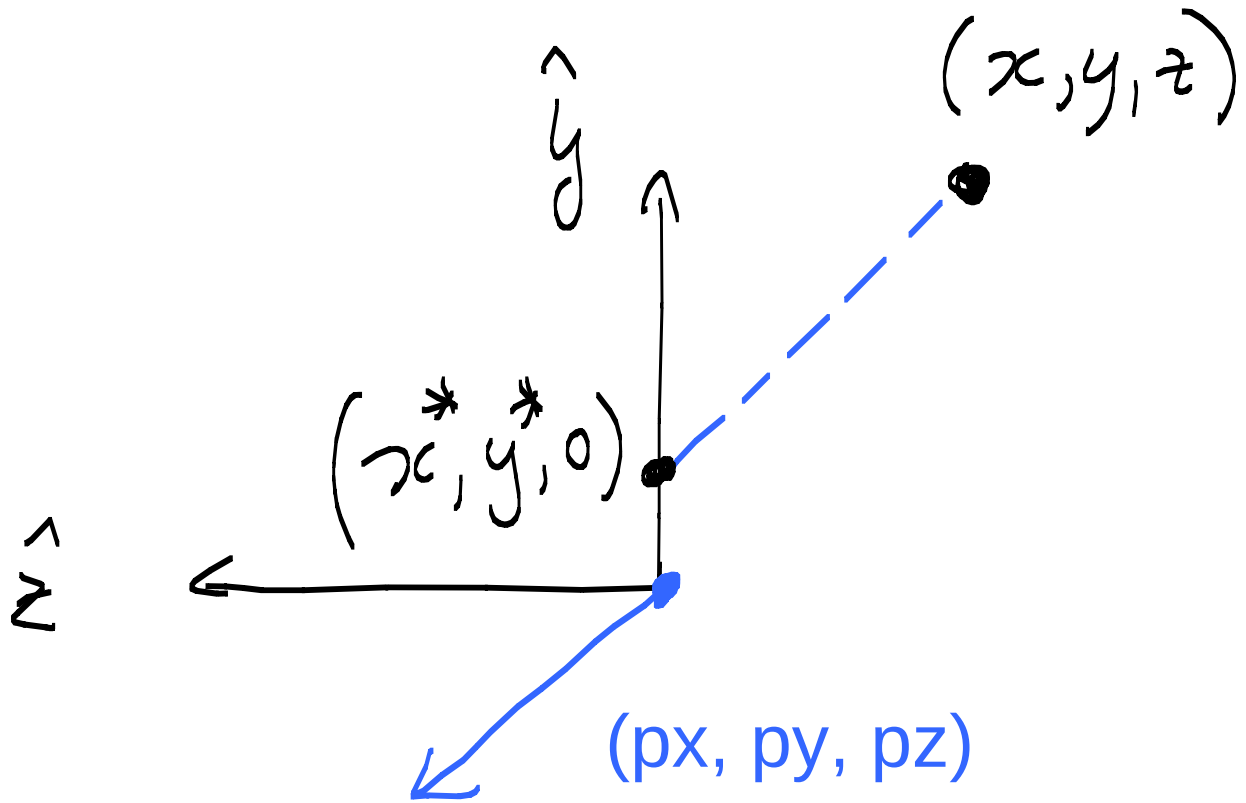
$\hat{x}, \hat{y}, \hat{z}$ all project to the same length in the image.



Parallel projection

Example: Project to z=0 plane.

But now project in general direction (px, py, pz).



How can we calculate the projection point ?

$$(x, y, z) + t(p_x\ p_y\ p_z) = (x^*, y^*, 0)$$

First, use z coordinate to solve for t. Then plug in:

$$\Rightarrow \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} x - \frac{p_x}{p_z} z \\ y - \frac{p_y}{p_z} z \end{bmatrix}$$

$$\begin{bmatrix} x - \frac{p_x}{p_z} z \\ y - \frac{p_y}{p_z} z \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} & & & \\ & & & \\ & ? & & \\ & & & \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

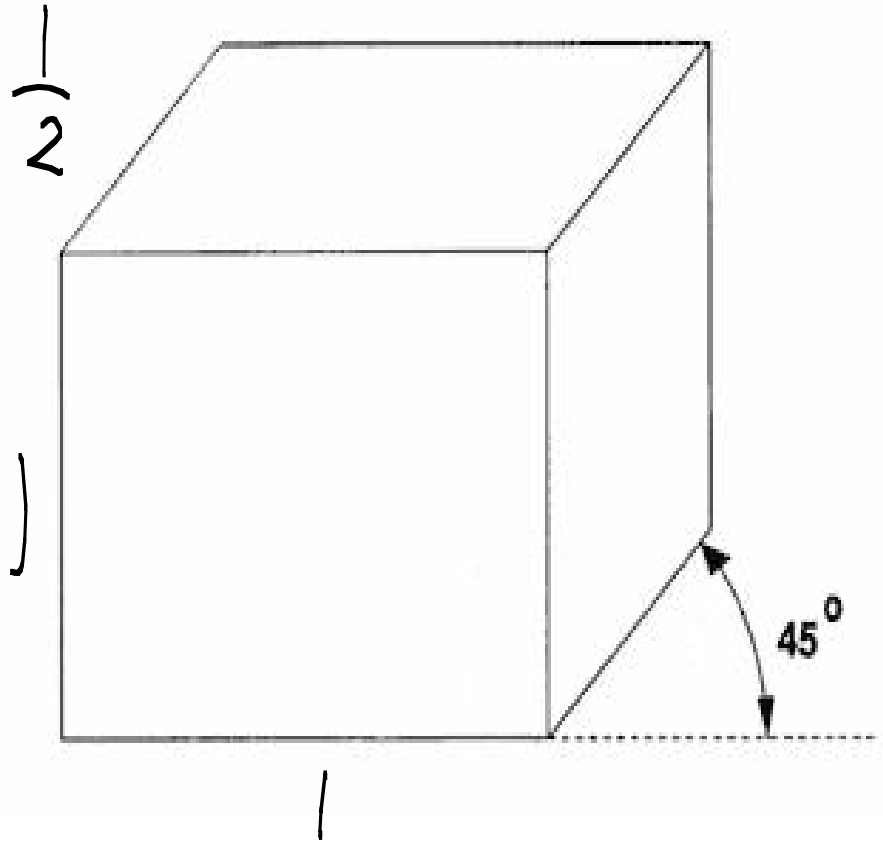
How can we write parallel projection using a 4x4 matrix ?

$$\begin{bmatrix} x - \frac{p_x}{p_z} z \\ y - \frac{p_y}{p_z} z \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -p_x/p_z & 0 \\ 0 & 1 & -p_y/p_z & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Example of parallel projection: cabinet projection

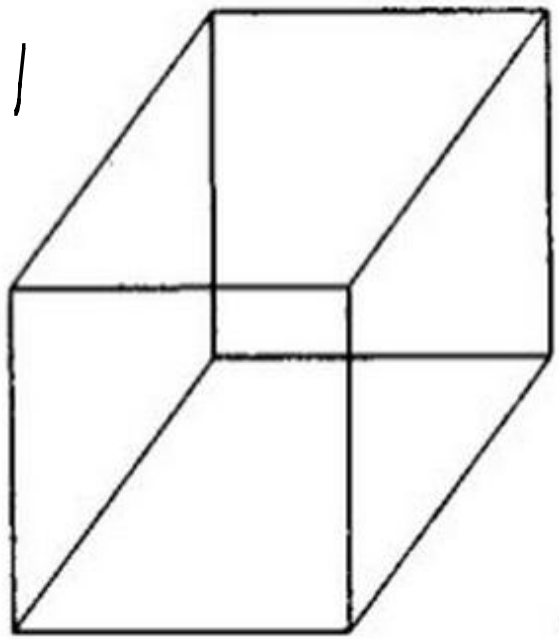
x, y axes of cube project to the same length in the image,

but z axis projects to half that length.

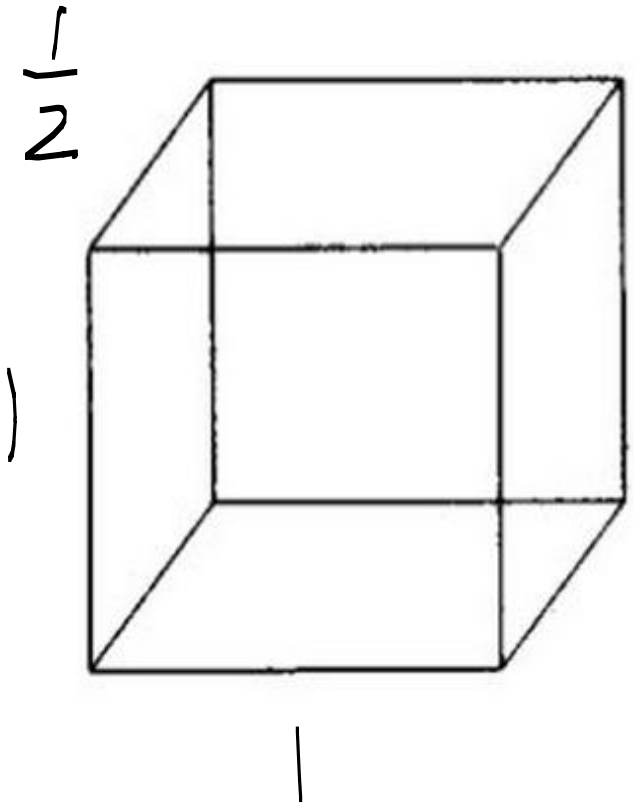


It doesn't have to be 45 deg. 30 deg is also common.

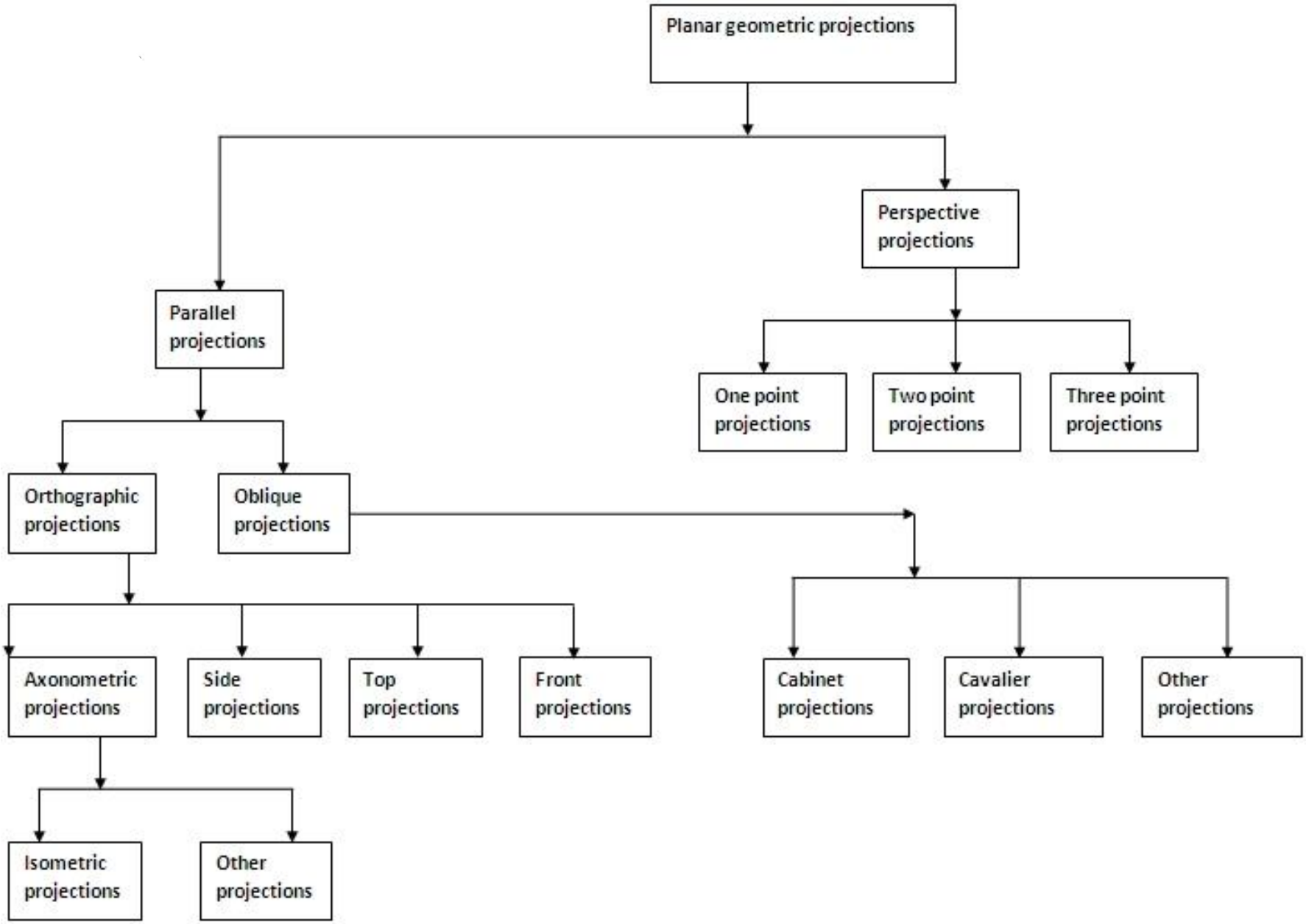
cavalier



cabinet



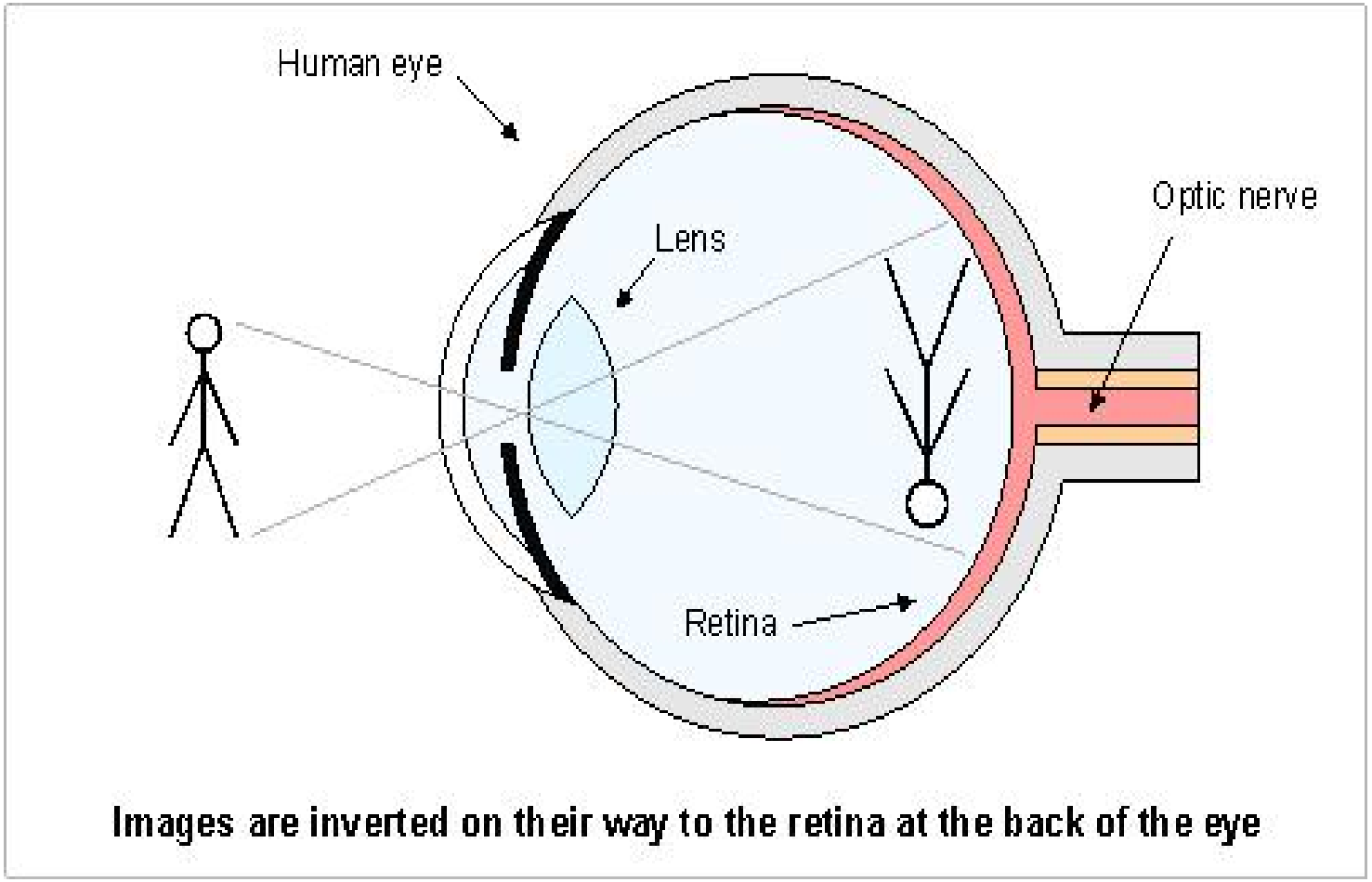
Which of these looks more like a cube?
(perceptual issue !)



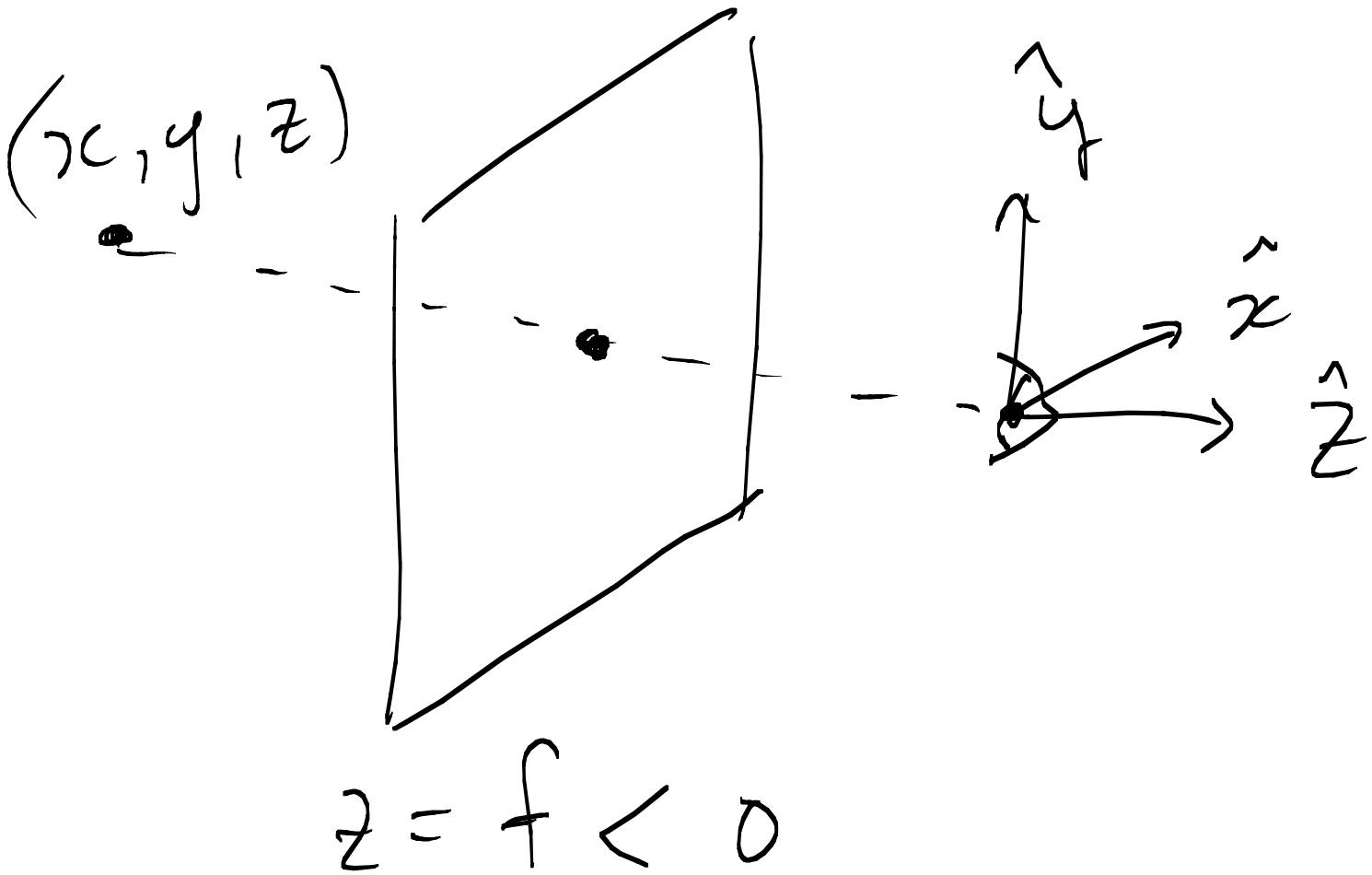
Architects and interior designers know these well.
(But you don't need to memorize the names.)

Perspective Projection

In real imaging systems (photography, human eye), real images are upside down and backwards.



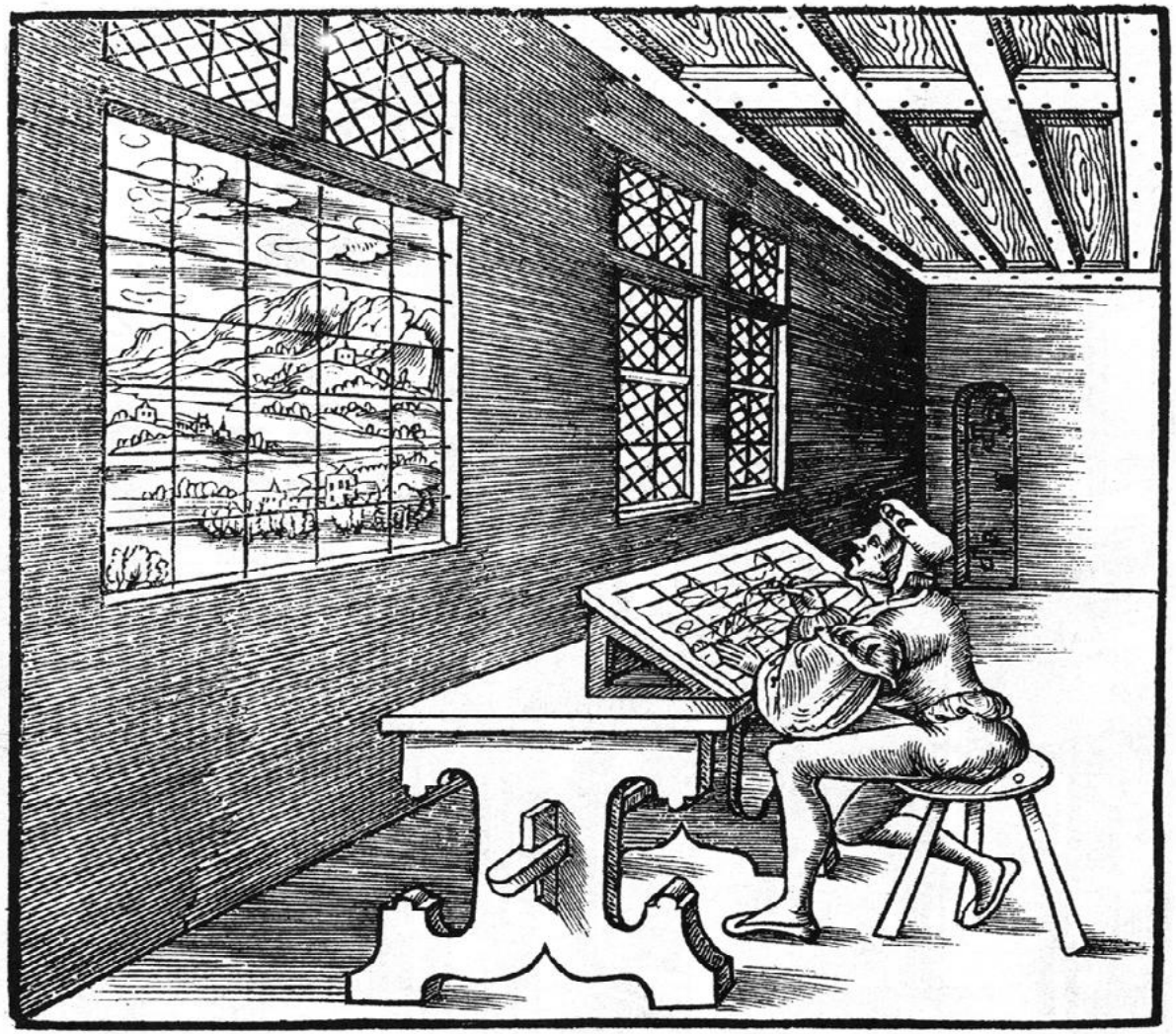
In computer graphics, the projection surface is in front of the viewer (negative z).



Think of the viewer as looking through a window.

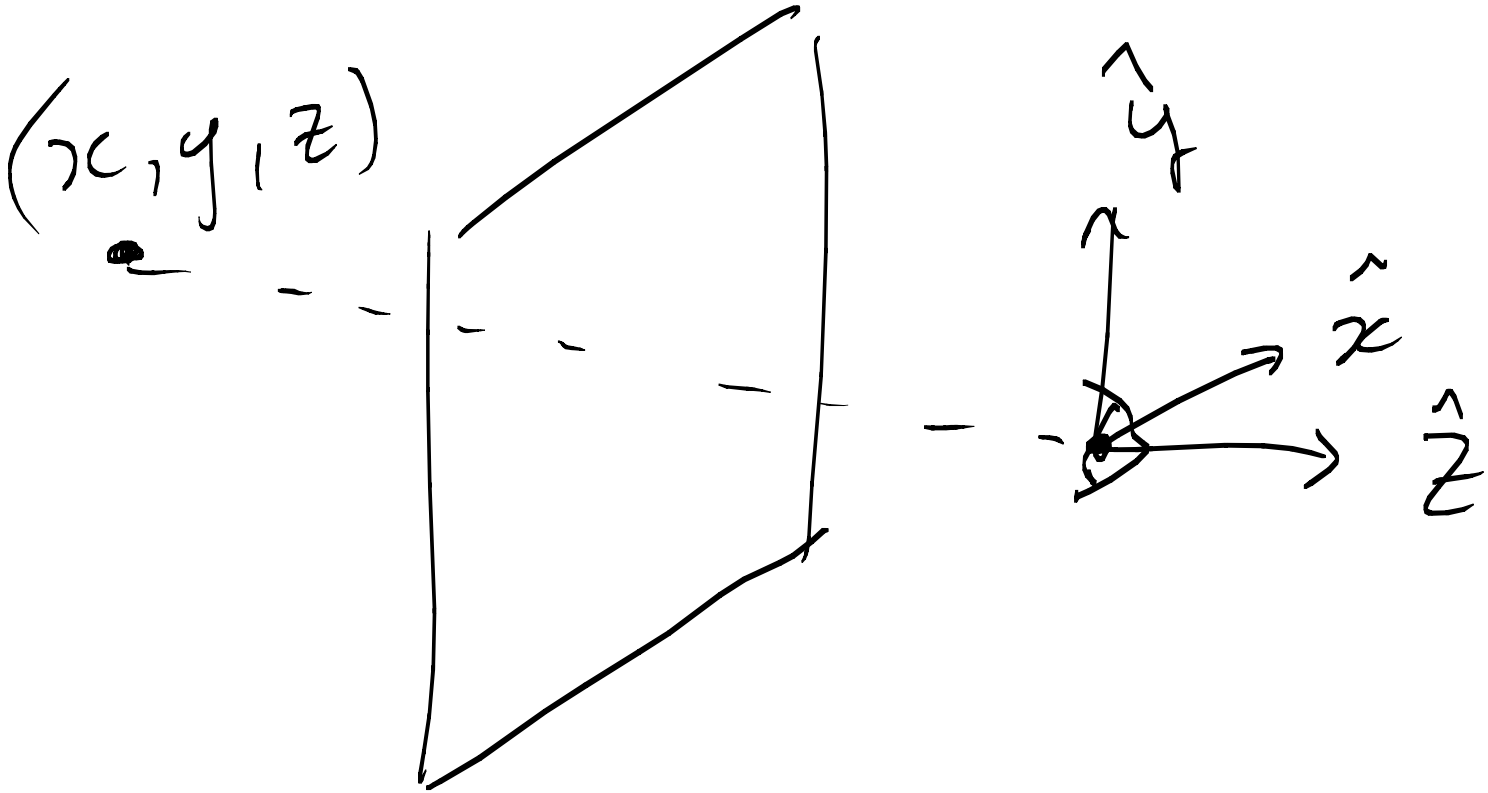
Alberti's window (1435).

Illustration below was drawn in 1531.

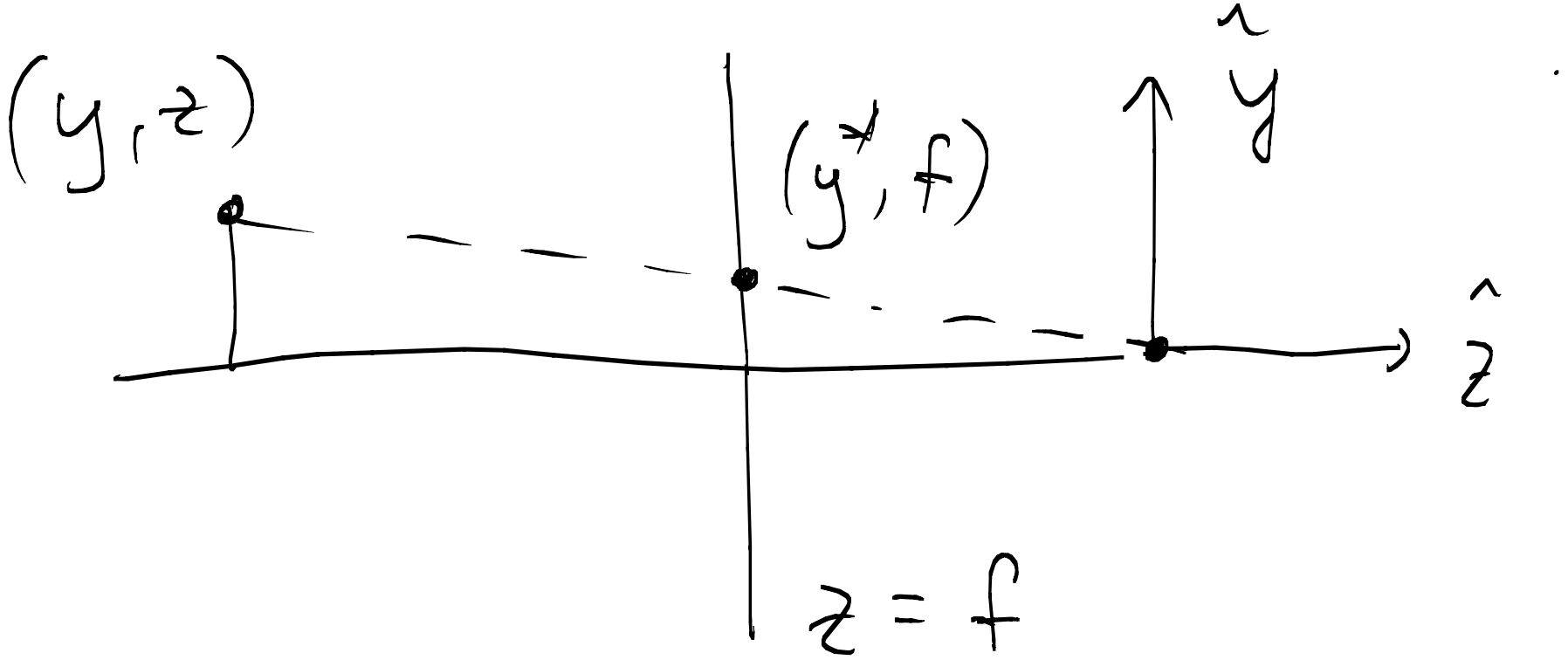


"Center of projection"

All scene points project towards the viewer (origin).



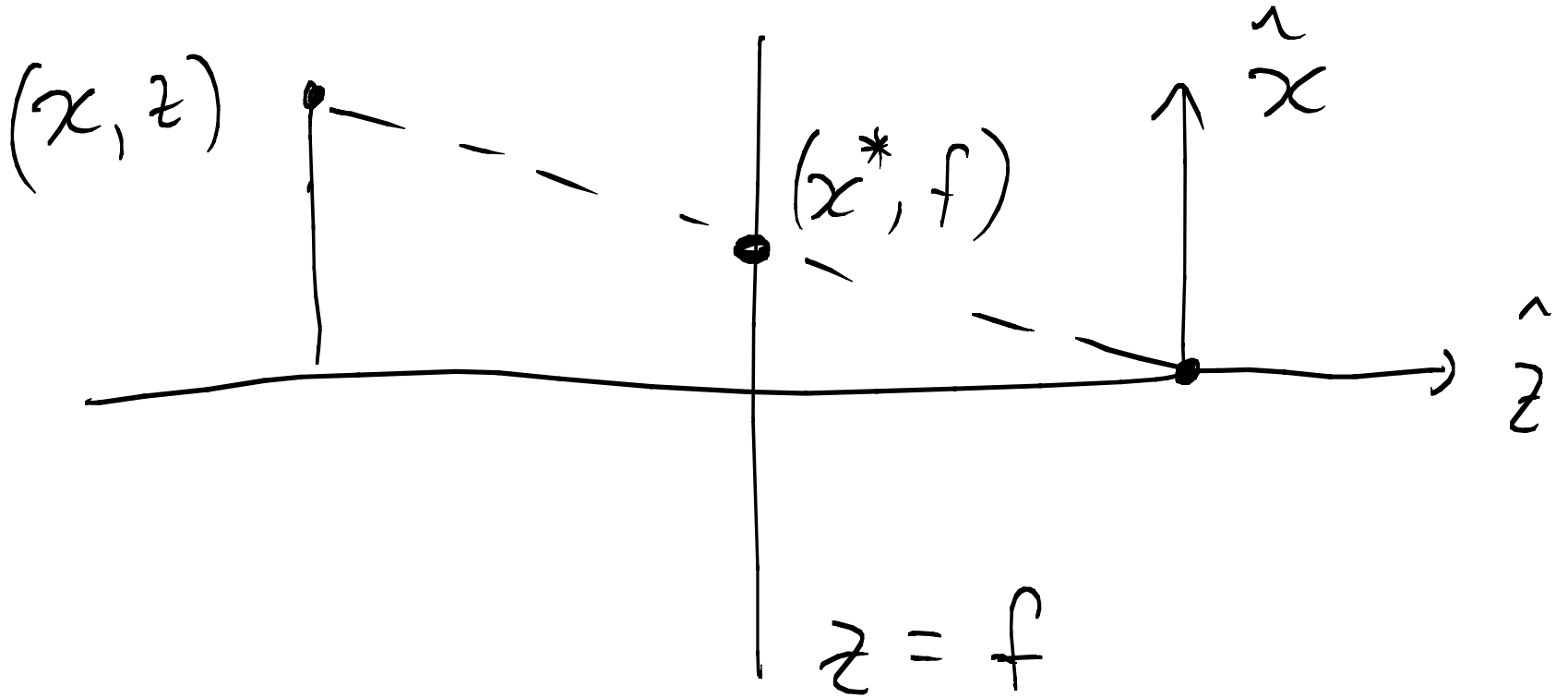
view from side



Similar triangles implies:

$$\frac{y}{z} = \frac{y^*}{f}$$

view from above



Similar triangles implies:

$$\frac{x}{z} = \frac{x^*}{f}$$

(x, y, z) projects to $\left(\frac{x}{z}f, \frac{y}{z}f, f\right)$

Re-write in homogeneous coordinates:

$(x, y, z, 1)$ projects to $\left(\frac{x}{z}f, \frac{y}{z}f, f, 1\right)$

$\equiv \left(xf, yf, zf, z\right)$

$$\begin{bmatrix} xf \\ yf \\ zf \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

We can consider these to be equivalent transformations.

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix}$$

$M(w\vec{p}) = wM\vec{p}$ where

$\vec{p} = (x, y, z, 1)$.

Vanishing Points

Under perspective projection, parallel lines in 3D meet at a single point in the image.

How to express this mathematically?

Parallel lines in 3D

$$\{(x_0, y_0, z_0) + t\,(v_x, v_y, v_z)\}$$

assume constant

Two different (x0,y0,z0) define two different lines.

Vanishing points ? Let t -> infinity and look at projection.

$$(x_0 + v_x t,\ y_0 + v_y t,\ z_0 + v_z t,\ 1)$$

$$\equiv \left(\frac{x_0}{t} + v_x,\ \frac{y_0}{t} + v_y,\ \frac{z_0}{t} + v_z,\ \frac{1}{t}\right)$$

Let $t \to \infty$

gives $(v_x, v_y, v_z, 0)$

i.e. all lines vanish at same point at infinity

The set of parallel lines all go to a point at infinity
(vx, vy, vz, 0).

This point projects to the image at a vanishing point.

$$\begin{bmatrix} f\,V_x/V_z \\ f\,V_y/V_z \\ f \\ 1 \end{bmatrix} \equiv \begin{bmatrix} f\,V_x \\ f\,V_y \\ f\,V_z \\ V_z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \\ 0 \end{bmatrix}$$

if $V_z \neq 0$

n-point Perspective (n = 1, 2, 3)

An image has n-point perspective if it has n ***finite*** vanishing points.

Many man-made scenes contain three sets of (perpendicular) parallel lines.

e.g. A building may be a scaled cube.

A cube defines three points at infinity, and hence three vanishing points.

1-point perspective (not 3)

Lines that are parallel to camera x axis and y axis have vanishing points at infinity.

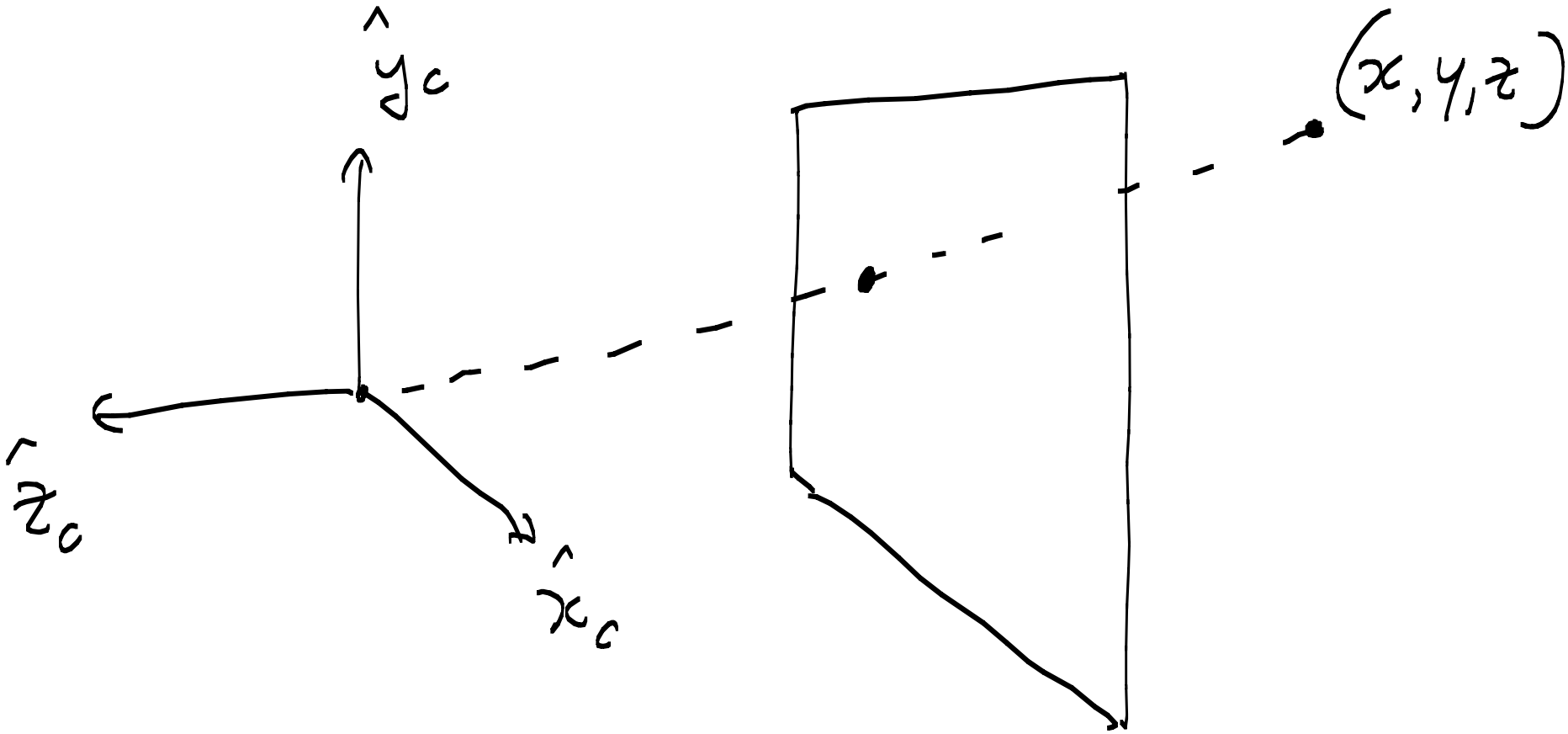
2-point perspective (not 3)

Lines that are parallel to camera y axis have a vanishing point at infinity.

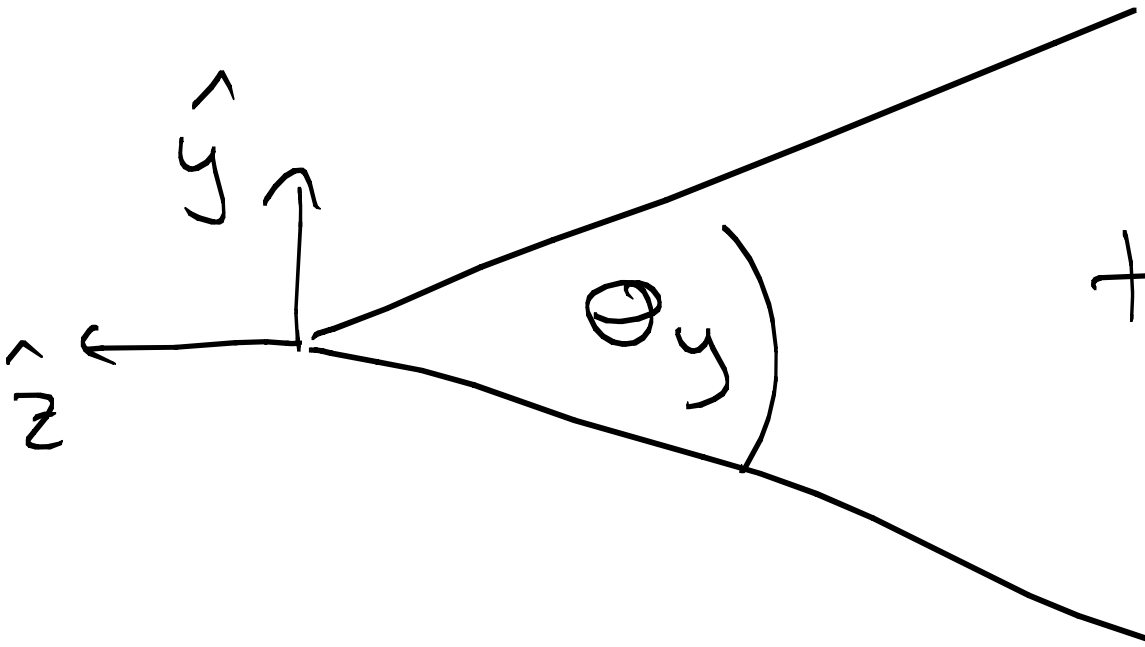
3-point perspective

All vanishing points are finite (but are outside window).

Recall the idea of a viewer looking through a **window**.

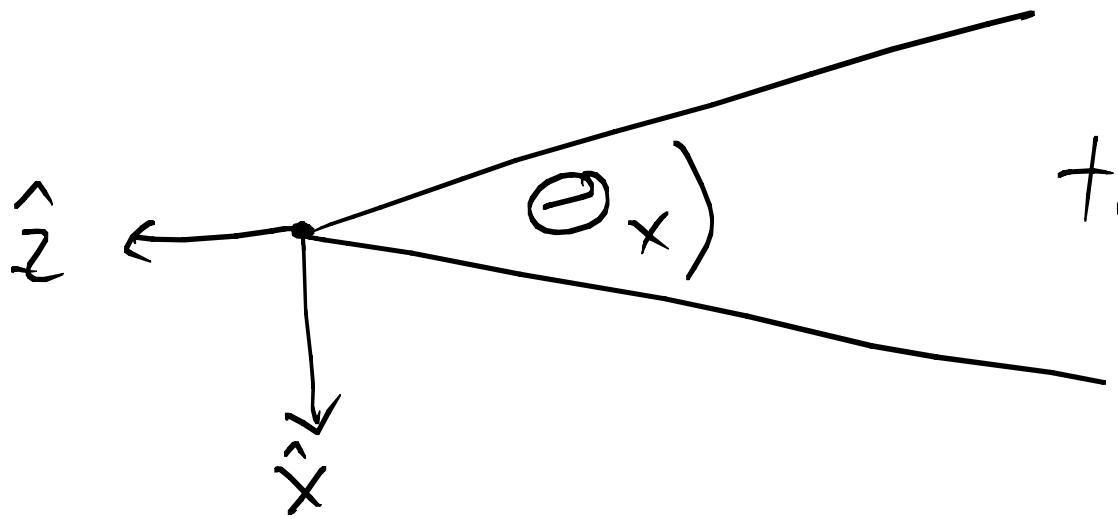


view from side



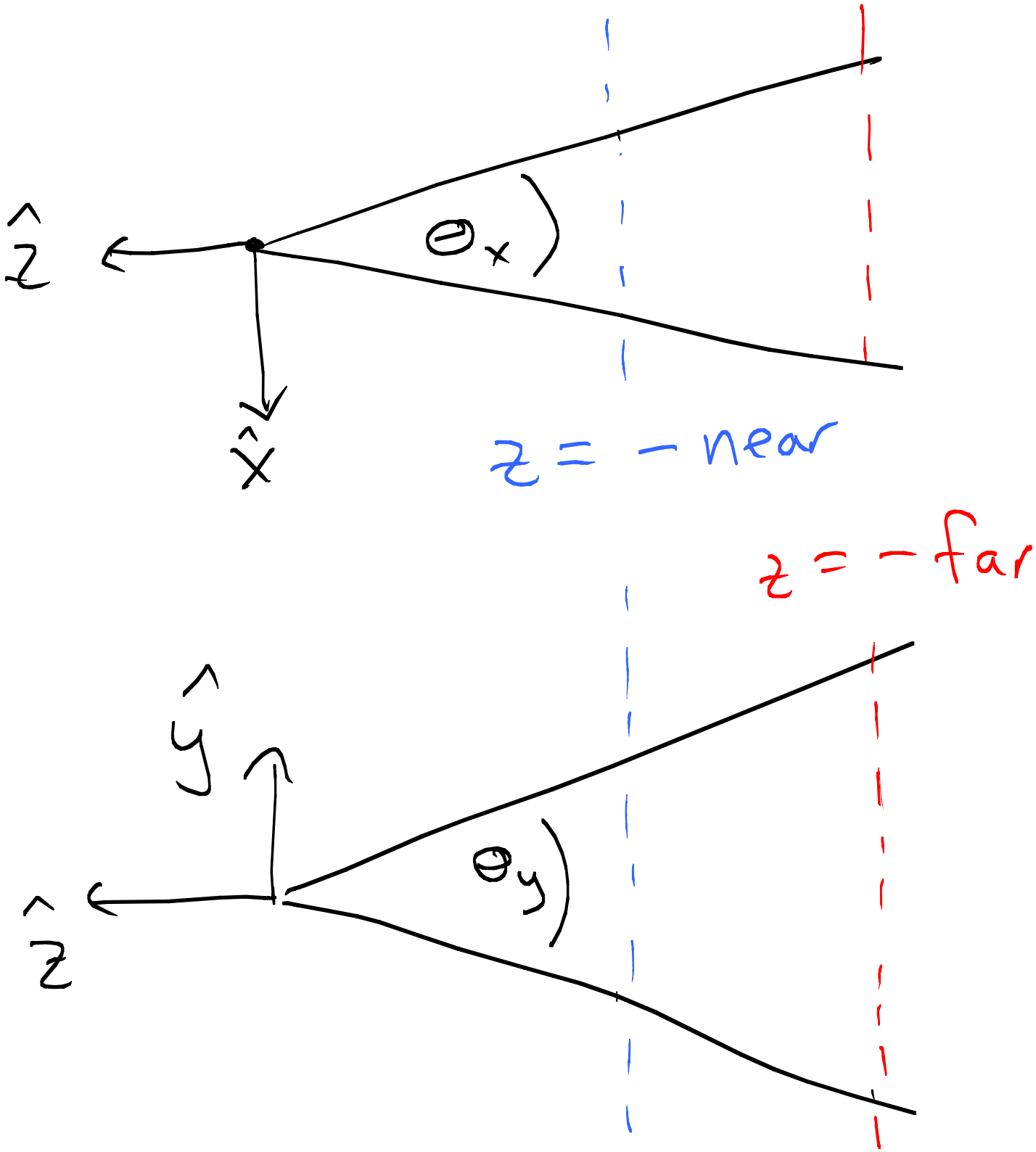
$$\tan\frac{\theta_y}{2} = \pm\frac{y}{z}$$

view from above

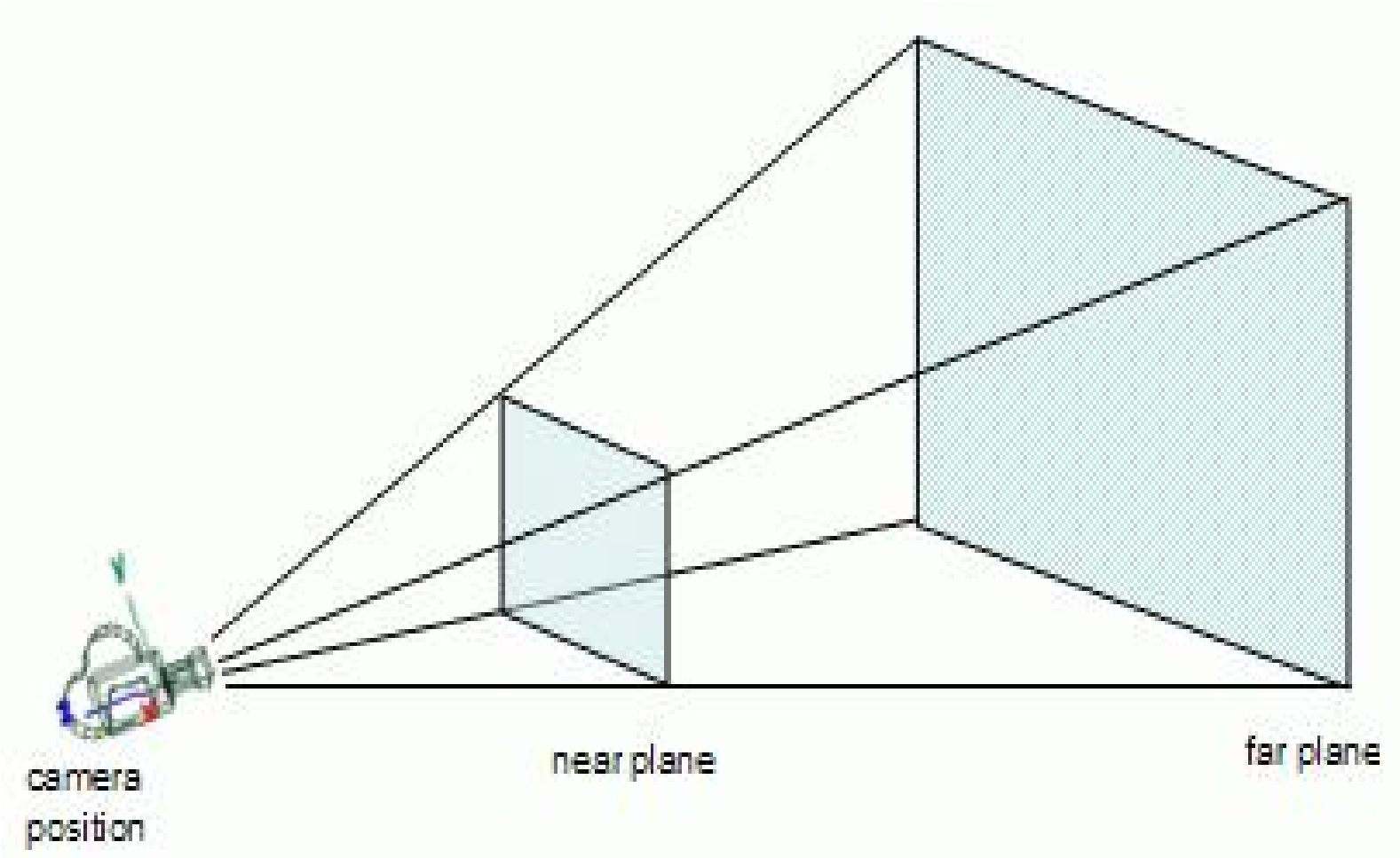


$$\tan\frac{\theta_x}{2} = \pm\frac{x}{z}$$

$$xy \text{ aspect ratio} = \tan\frac{\theta_x}{2} \Big/ \tan\frac{\theta_y}{2}$$


$\hat{z}$
θ_x
$\hat{x}$
$z = -near$
$z = -far$
$\hat{y}$
θ_y
$\hat{z}$

View volume (frustum)



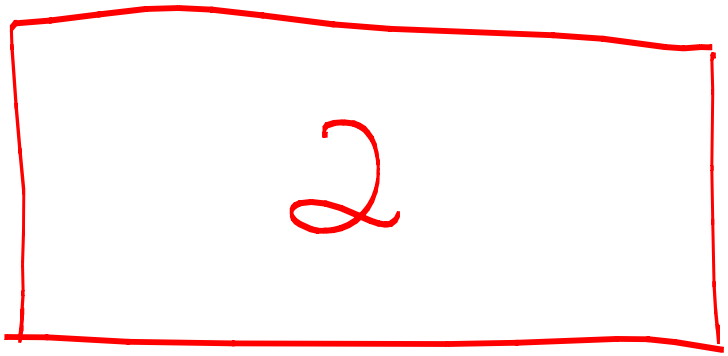
"truncated pyramid"

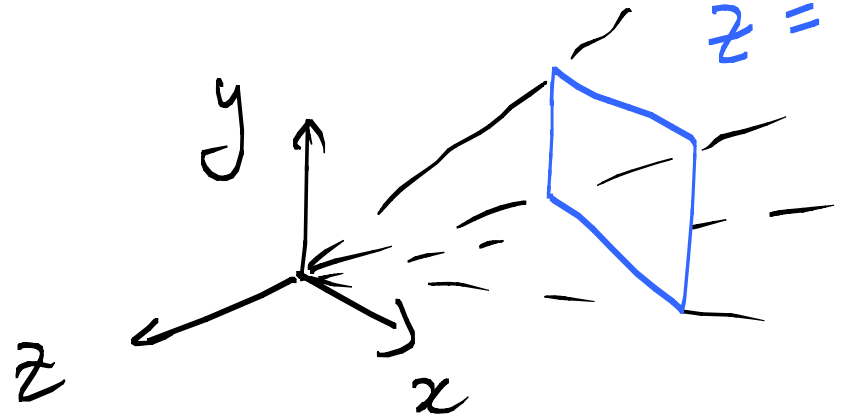
OpenGL

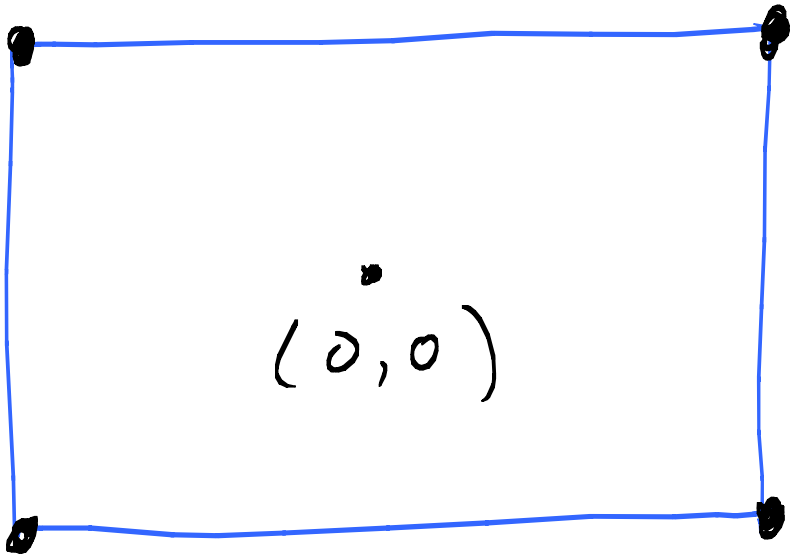
gluPerspective(θ_y , xy aspect ratio , near, far)

$$\text{xy aspect ratio} = \frac{\tan \frac{\theta_x}{2}}{\tan \frac{\theta_y}{2}}$$

eg







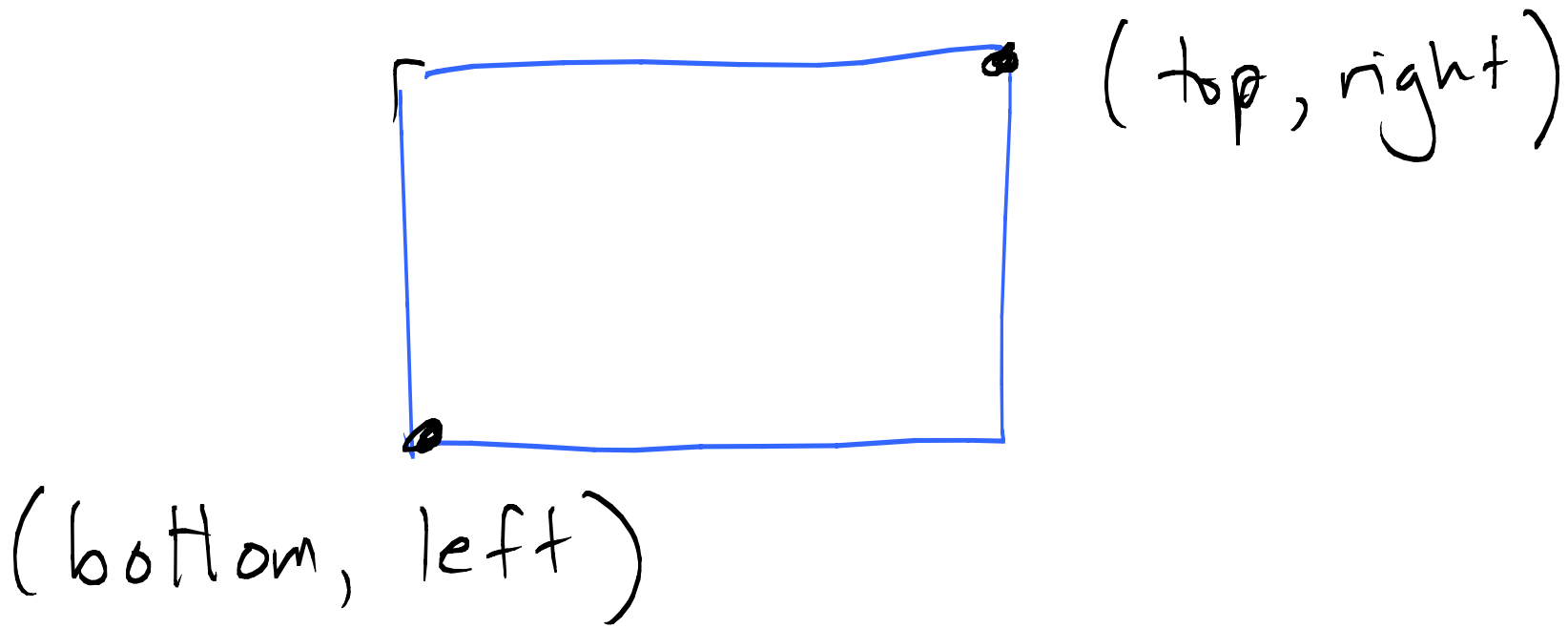
$$y = \pm \text{near} \tan\frac{\theta_y}{2}$$

$$x = \pm \text{near} \tan\frac{\theta_x}{2}$$

OpenGL

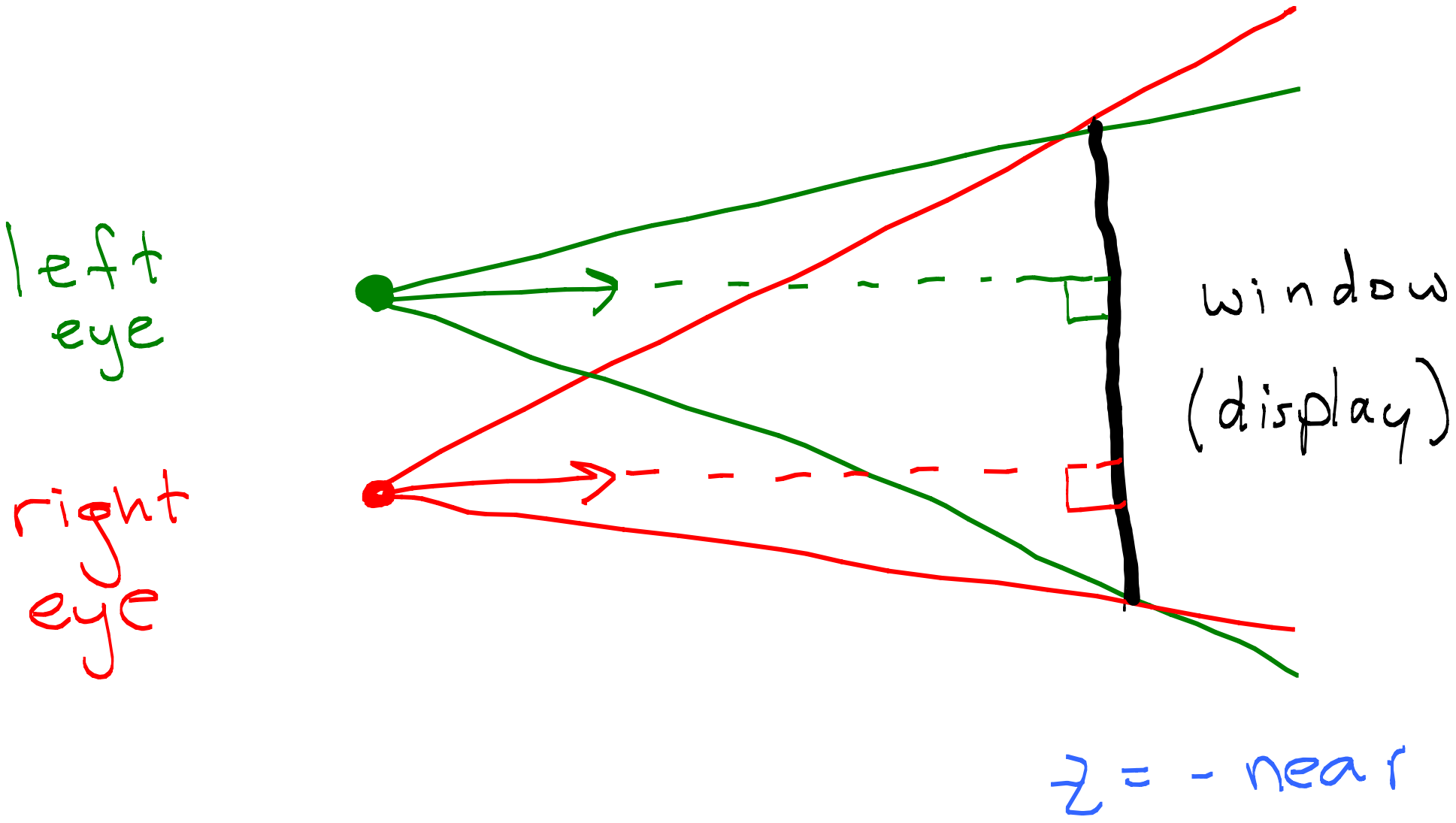
A more general definition of a view volume/frustrum.

In the z = - near plane, define:

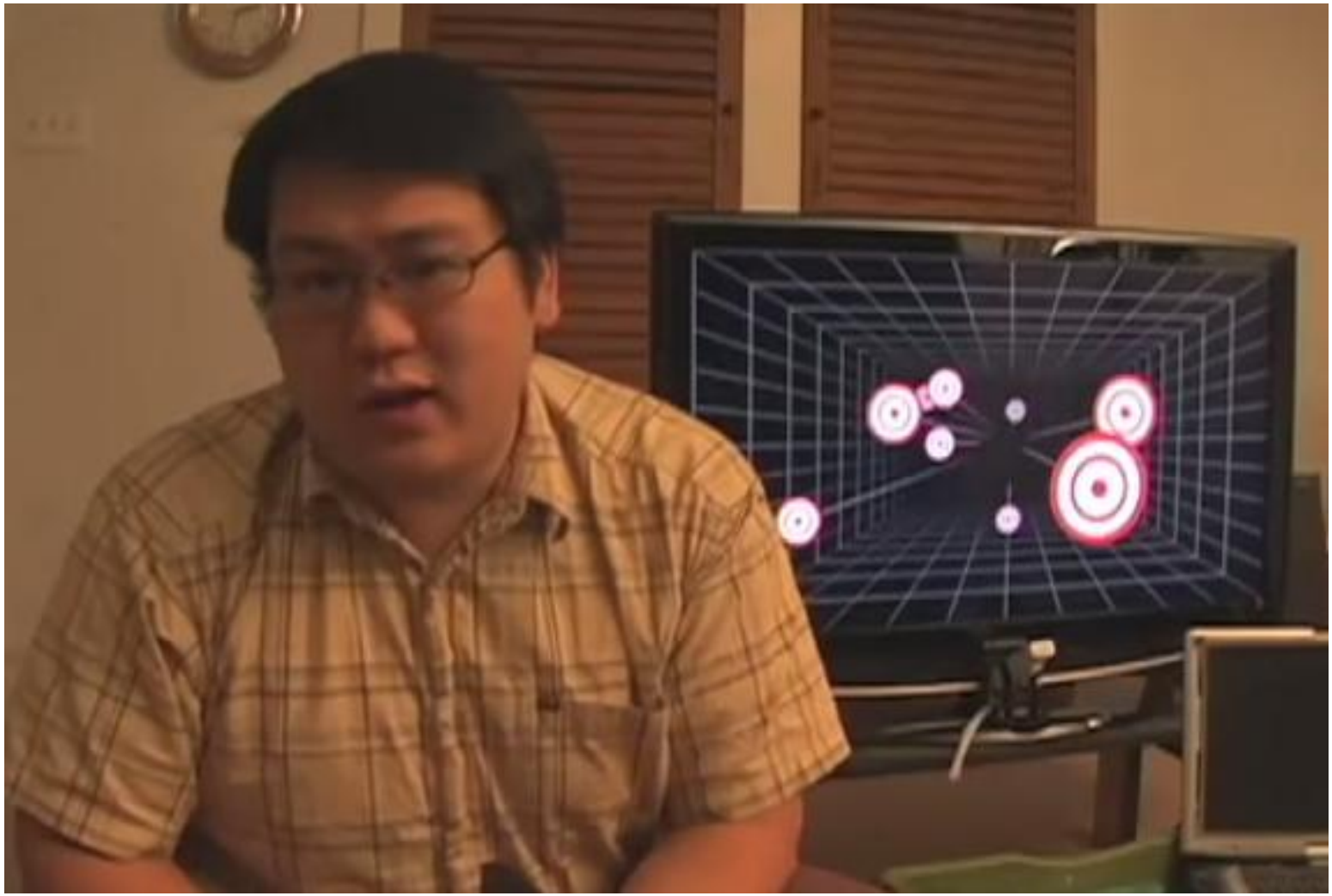


glFrustum(left, right, bottom, top, near, far)

Application 1: 3D stereo displays



Application 2: head-tracked displays



https://www.youtube.com/watch?v=Jd3-eiid-Uw

Assignment 1 to be posted end of next week.

Programming language will be Python.

(Python/OpenGL version and installation details possibly available before then.)