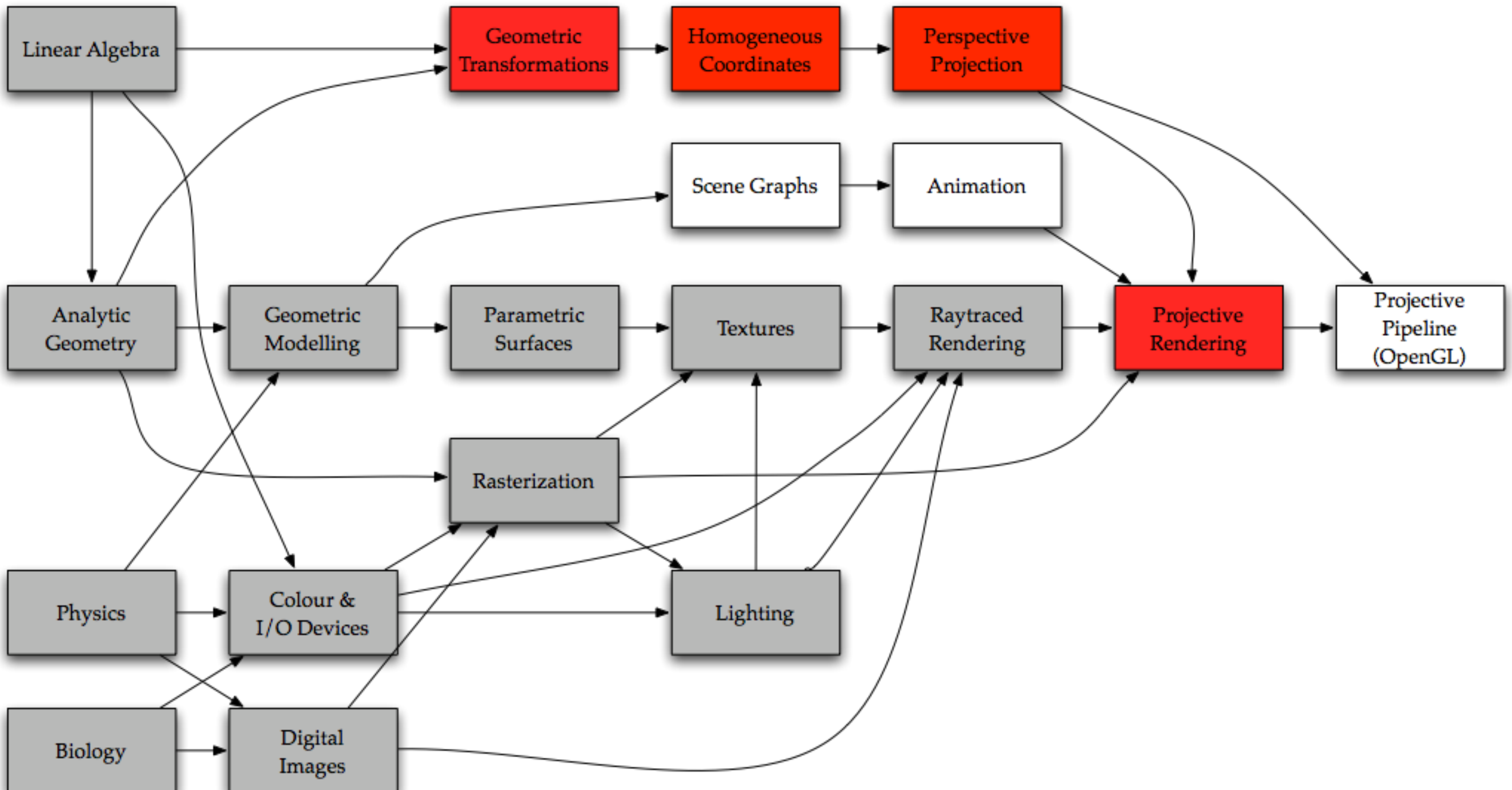
Homogeneous Coordinates & Perspective Projection

Where we Are



Linear Algebra
Geometric Transformations
Homogeneous Coordinates
Perspective Projection
Scene Graphs
Animation
Analytic Geometry
Geometric Modelling
Parametric Surfaces
Textures
Raytraced Rendering
Projective Rendering
Projective Pipeline (OpenGL)
Rasterization
Physics
Colour & I/O Devices
Lighting
Biology
Digital Images

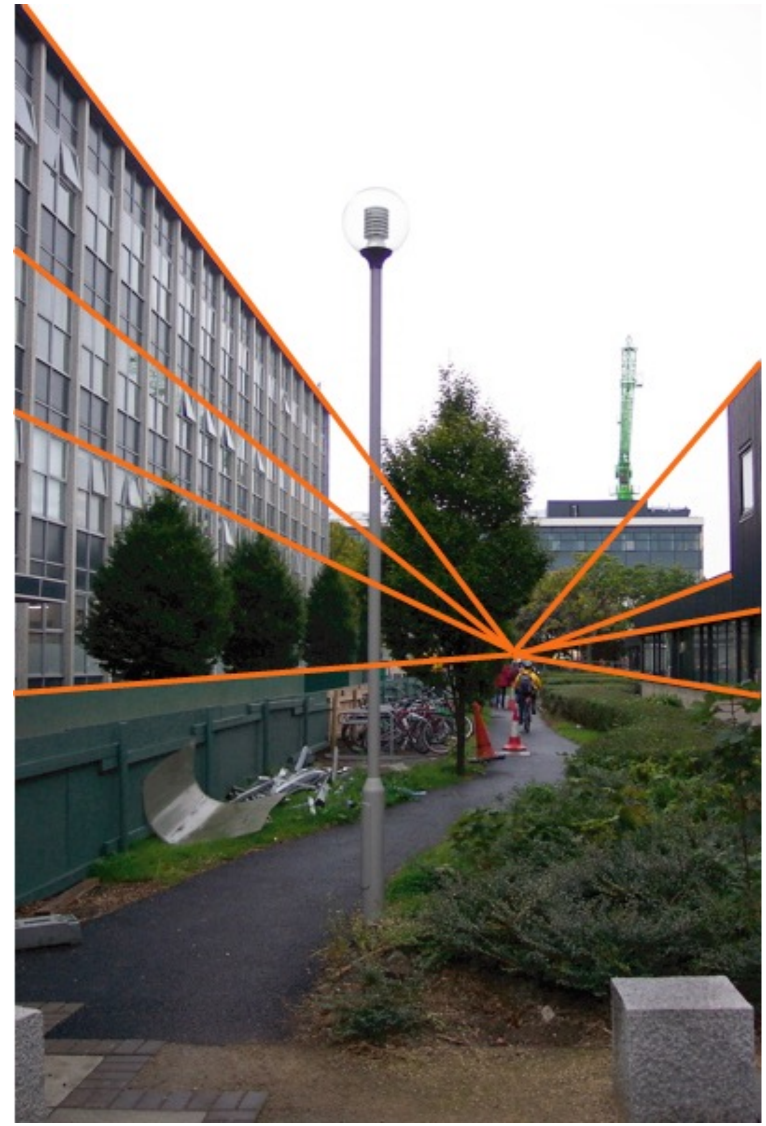
Perspective

Origin of Perspective

- Renaissance artists wanted to draw
  - buildings
  - streets
- These tend to have parallel lines
- But they don't *look* parallel

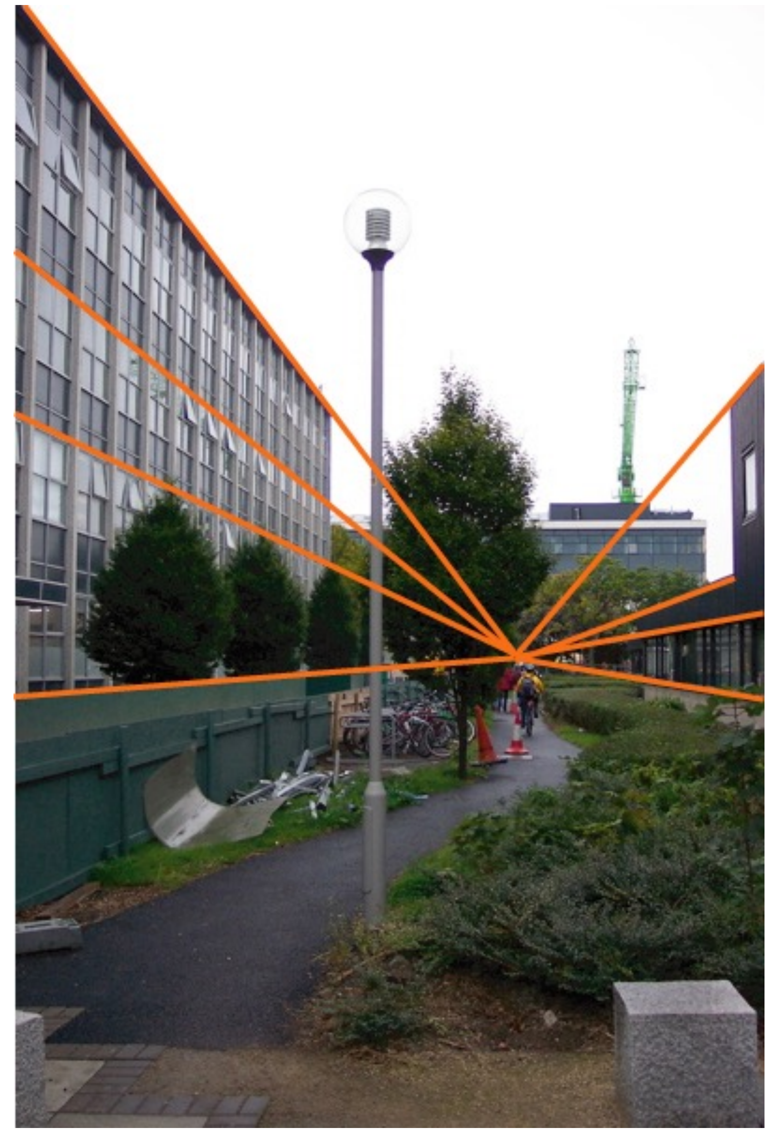
Receding Parallels

- Orange lines are:
  - parallel to view dir.
  - but not visually
- Other parallel lines
  - perp. to view dir.
  - remain parallel



1-Point Perspective

- Orange parallel lines:
  - converge visually
  - to a *vanishing point*
- Artists exploit this
  - place vanishing point
  - sketch parallel lines
  - build rest of image



Result: Canaletto

1-point Perspective

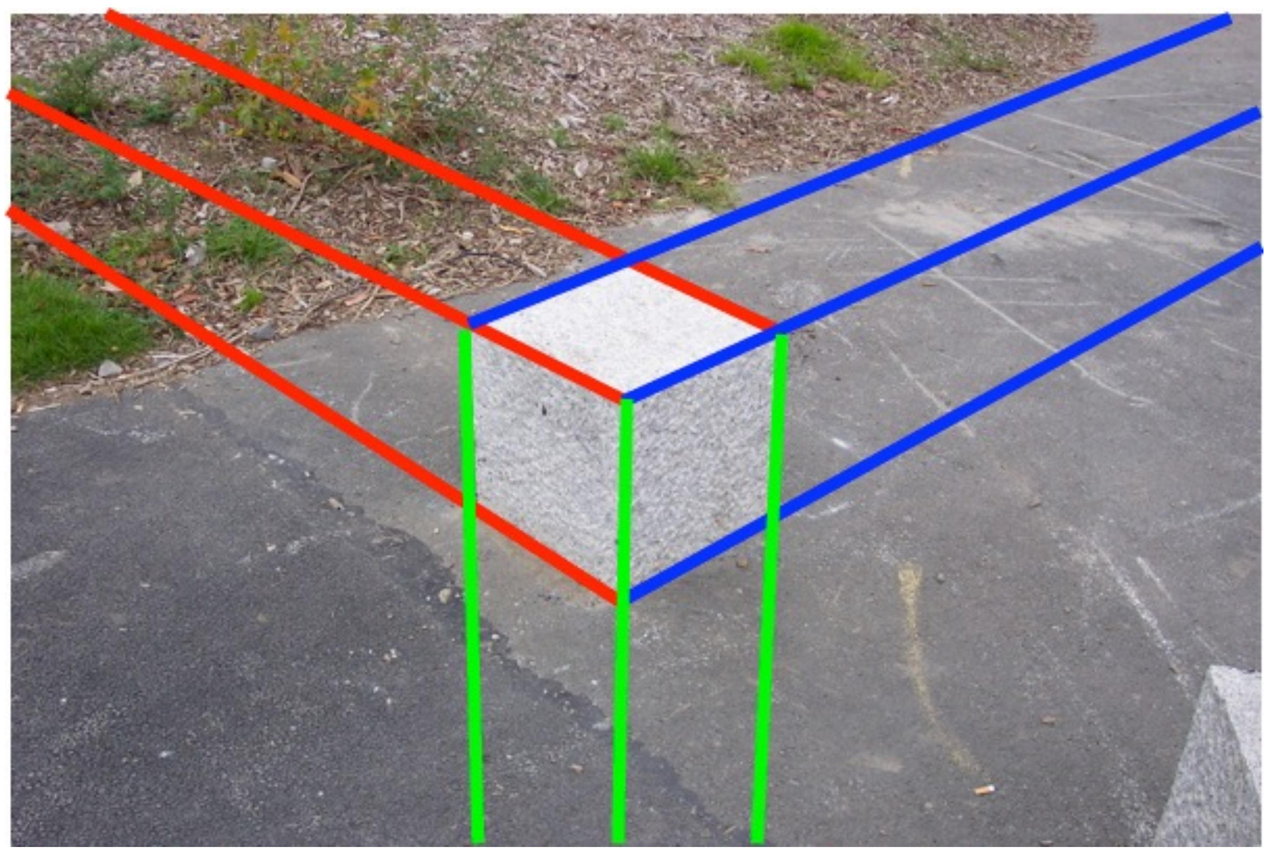
- These images look down a street
    - the view direction is straight down it
    - other surfaces are perpendicular
- This isn't always true
    - so we can get more vanishing points

2-point Perspective

- Parallel sets of lines *always* vanish
  - unless perpendicular to view direction

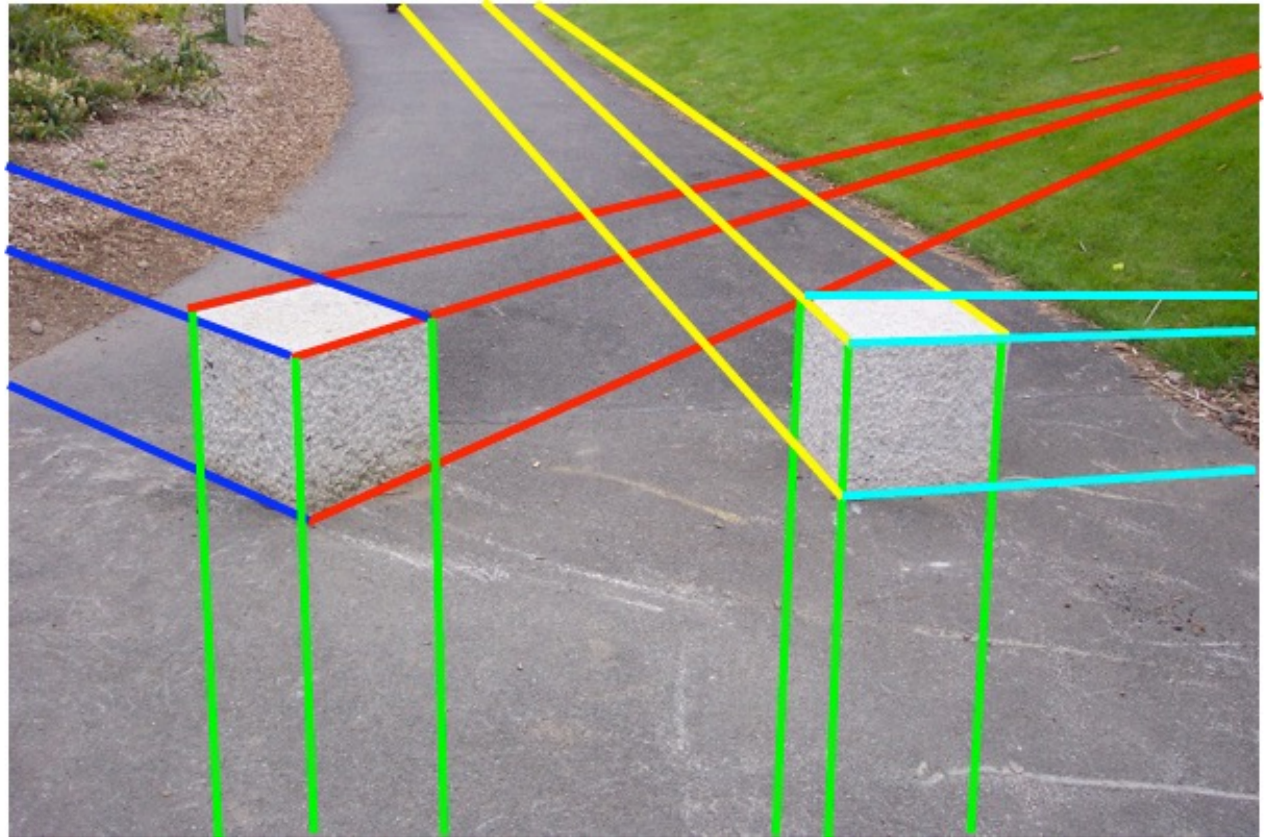
3-point Perspective

- We can even get 3 vanishing points:



Can we get more?

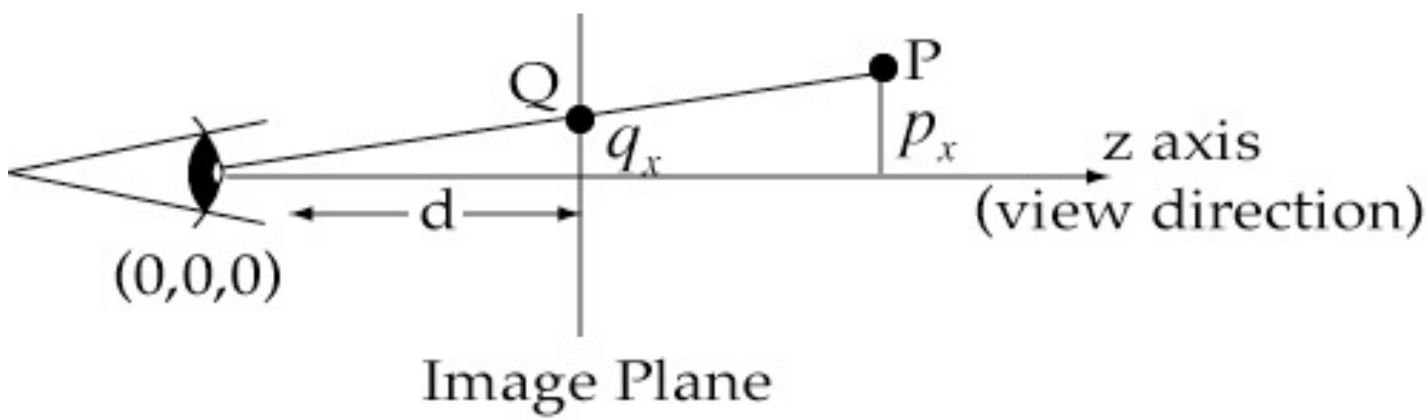
- Yes, if objects are misaligned:



Mathematical Perspective

- Use similar triangles to compute Q

$$\frac{q_x}{q_z} = \frac{p_x}{p_z} \text{ or } q_x = p_x \cdot \frac{d}{p_z}$$



All 3 Coordinates

$$\left(q_x, q_y, q_z\right) = \left(p_x \cdot \frac{d}{p_z}, p_y \cdot \frac{d}{p_z}, d\right)$$

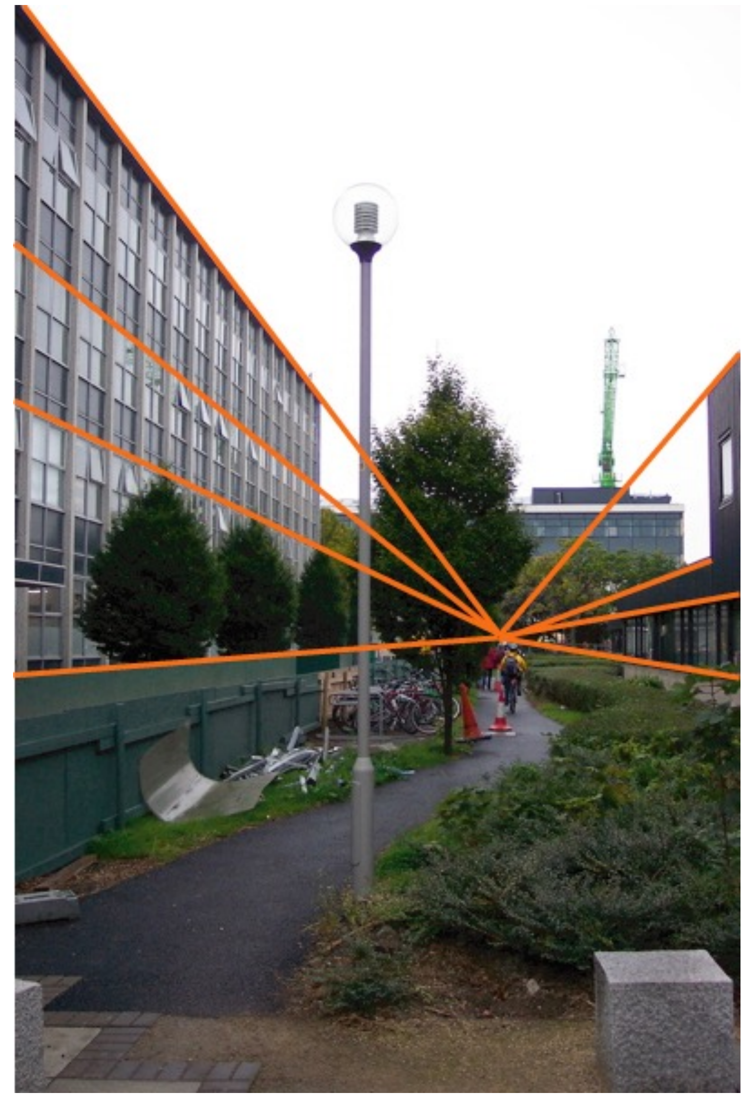
$$= \left(p_x \cdot \frac{d}{p_z}, p_y \cdot \frac{d}{p_z}, p_z \cdot \frac{d}{p_z}\right)$$

$$\cong \left(p_x, p_y, p_z, \frac{p_z}{d}\right) \text{(homog. coords)}$$

$$\cong \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \text{(homog. coords)}$$

Foreshortening

- Vertical spacing reduced further away
- Visible in window pillars on left
- One of the cues to depth of image



Foreshortening

- We assumed that z = 0, c = 1
- t is perpendicular distance to image plane
- What happens to evenly spaced points?
  - P + 1V, P + 2V, P + 3V
  - These map to $V + \frac{1}{1}P, V + \frac{1}{2}P, V + \frac{1}{3}P$
  - No longer evenly spaced

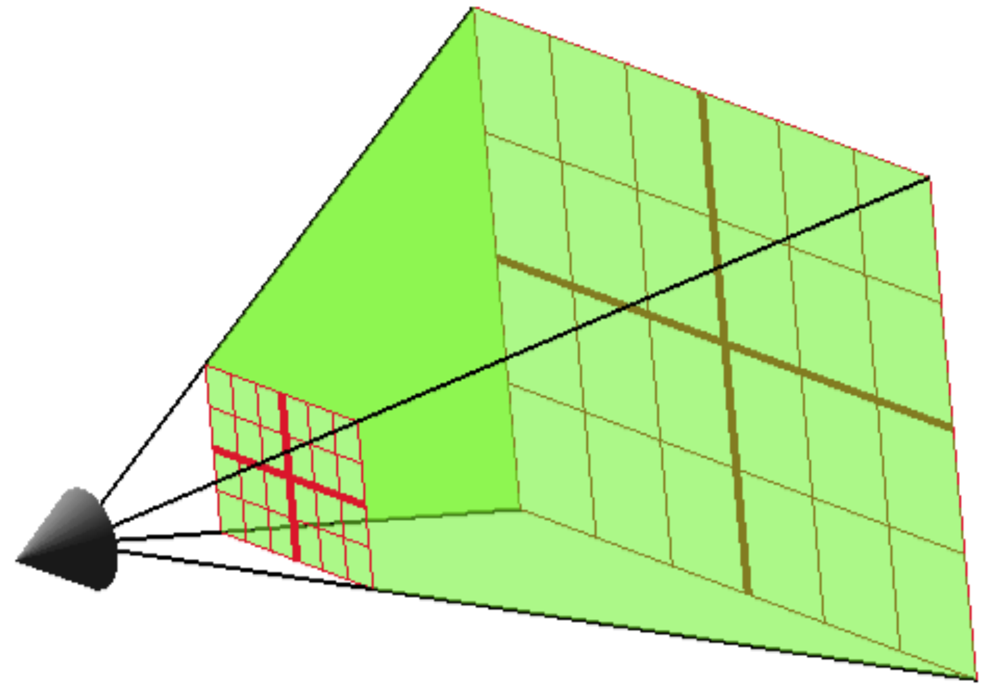
So . . .

- Projection maps lines to lines
- Lines perpendicular to view stay parallel
- Others intersect at *vanishing points*
- And distant objects *foreshorten*

Field of View

- We cannot see everything
  - eyes have limited *field of view*
  - think of it as the size of the glass sheet
  - we ignore very near or far objects
  - this defines a *view volume* that we can see

View Frustum



- For perspective, view volume is
  - a view *frustum* (a truncated pyramid)
  - a box in clipping coordinates (*CCS*)

Three Problems

- Represent *translation* in matrix form
- Apply sequences of transformations *efficiently*
- Represent *perspective* in matrix form

Cartesian coordinates won't work

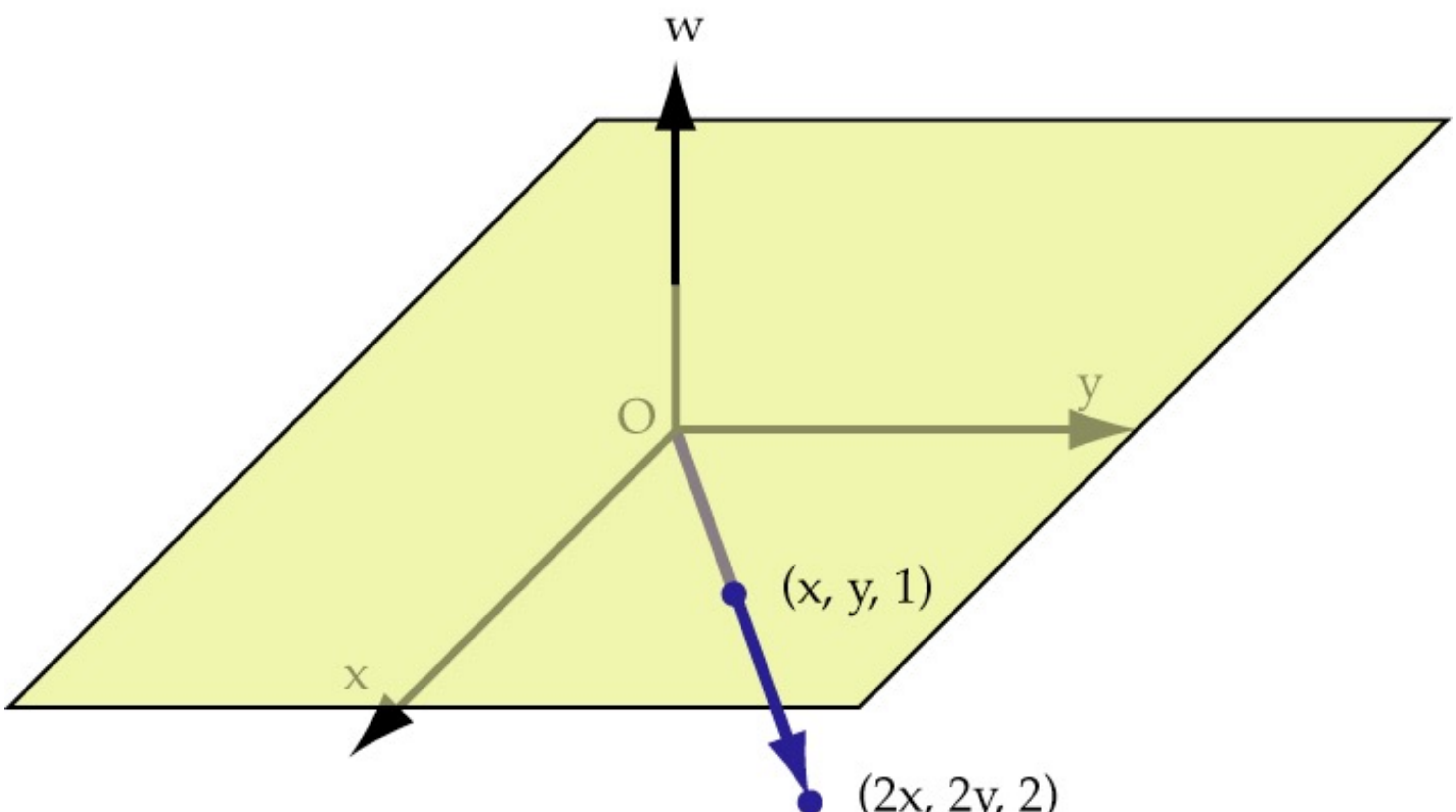
But *homogeneous* coordinates will

2D Homog. Coords

- Homogeneous coords exist in all dimensions
  - In 2D, (x, y) becomes (x, y, 1)
  - w is a *scale* factor: usually 1
  - (x, y, w) refers to the point $(\dfrac{x}{w}, \dfrac{y}{w})$
  - (1, 2, 1) is the same as (3, 6, 3)

Meaning of H.C.

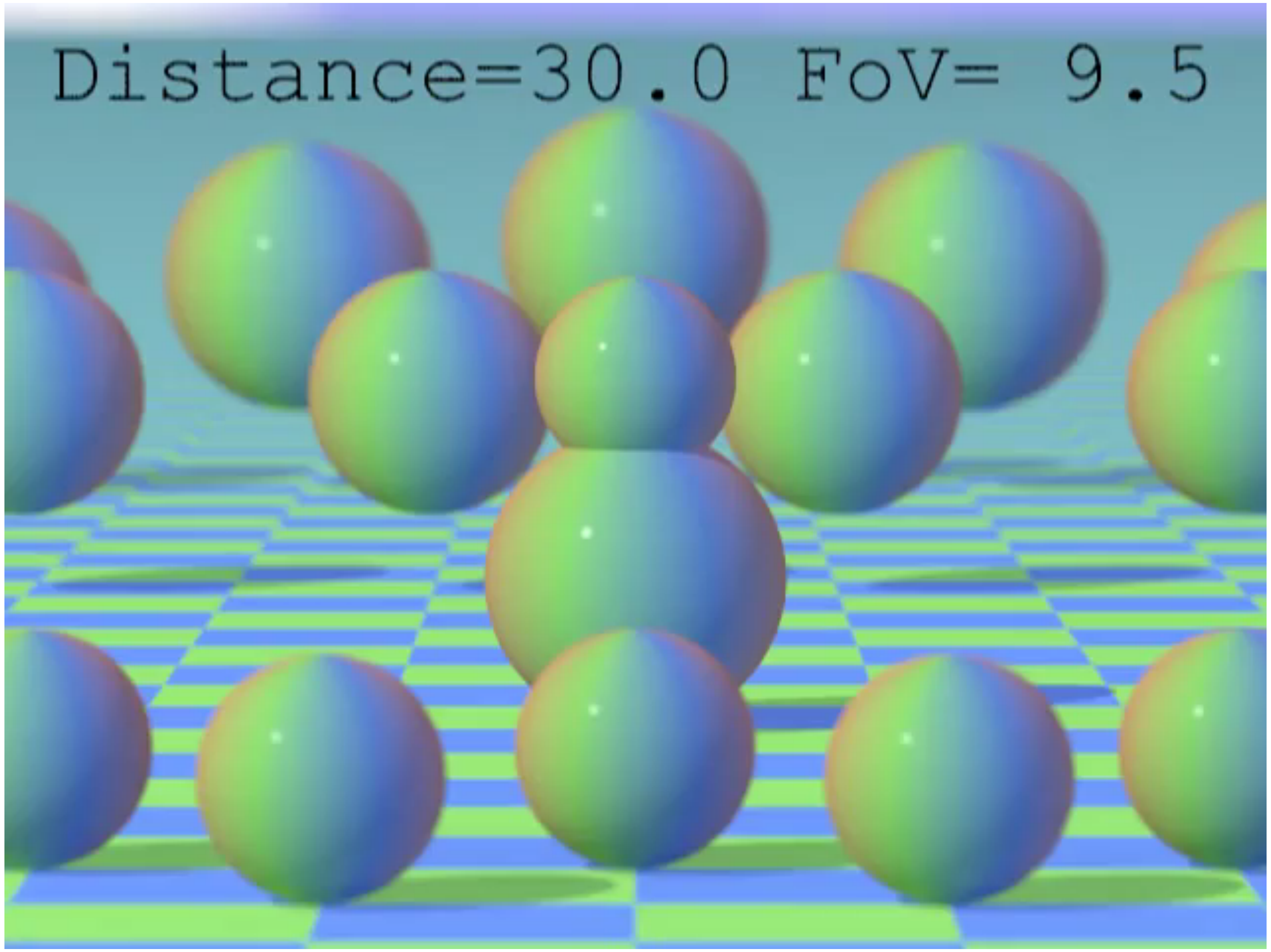
- Each point becomes a line in space
  - h.c. can represent projection as well



3D Homog. Coords.

- In 3D, homogeneous coordinates are (x, y, z, w)
    - x, y, z are the same as usual (almost)
    - w is the same as in 2D
- (x, y, z, w) refers to the point $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})$
- (1, 2, 3, 1) is the same as (3, 6, 9, 3)

Camera examples- Dolly zoom



Camera examples

Camera examples

Camera examples

Camera examples

Camera examples

Camera examples

Put simply

Homogeneous Vectors

- Vectors can be written as: (x, y, z, 0)
- Why?
  - Consider $\lim_{w \to 0} \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right)$

    $W \longrightarrow 0$
  - As the point travels outwards
  - So the vector (x, y, z) is (x, y, z, 0)
- Alternately, (x, y, z, 0) is infinitely far out

Homogeneous Normal Form

- Homogeneous normal form of a plane:

$$\begin{bmatrix} n_x \\ n_y \\ n_z \\ -c \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = n_x p_x + n_y p_y + n_z p_z - c$$

$$= \vec{n} \cdot p - c$$

Rotations

- Transformation matrices add 1 row / col
- Result of the multiplication is the same

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x \\ y\cos\theta - z\sin\theta \\ y\sin\theta + z\cos\theta \\ w \end{bmatrix}$$

Scaling

- Again, pretty much the same

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ s_z z \\ w \end{bmatrix}$$

Shearing

$$\begin{bmatrix} s_x & s_{xy} & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} s_x x + s_{xy} y \\ s_y y \\ s_z z \\ w \end{bmatrix}$$

Translation

- To translate (x, y, z, w) by (a, b, c, 1):

$$\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x + aw \\ y + bw \\ z + cw \\ w \end{bmatrix} \cong \begin{bmatrix} \frac{x + aw}{w} \\ \frac{y + bw}{w} \\ \frac{z + cw}{w} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} + a \\ \frac{y}{w} + b \\ \frac{z}{w} + c \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \\ \frac{z}{w} \end{bmatrix} + \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Homogeneous Matrix

- Divides into
  - *rotation (r)*
    - also scale, shear
  - *translation (t)*
  - *projection (p)*
  - *1*

$$\left[\begin{array}{ccc:c} r & r & r & t \\ r & r & r & t \\ r & r & r & t \\ \hdashline p & p & p & 1 \end{array}\right]$$

Advantages

1. H.C. represent all affine transformations
   - Rotation
   - Scaling
   - Shearing
   - Translation
2. Vectors have different rep. than points
3. H.C. also represent projective transforms
4. We can compose transformations

Multiple Transformations

- What if we want to do several things?
  - e.g. rotate (R), scale (S), then shear (H)
- We just multiply by each matrix
  - $p' = H(S(Rp)))$
  - but this is slow

Transform Cost

- Each vertex has 4 coordinates
- Matrix multiply takes 16 mult, 12 add
- For 10,000 vertices, it adds up
- If we apply 3 matrices (H,S,R), it costs:
  - 16 * 10,000 * 3 = 480,000 operations

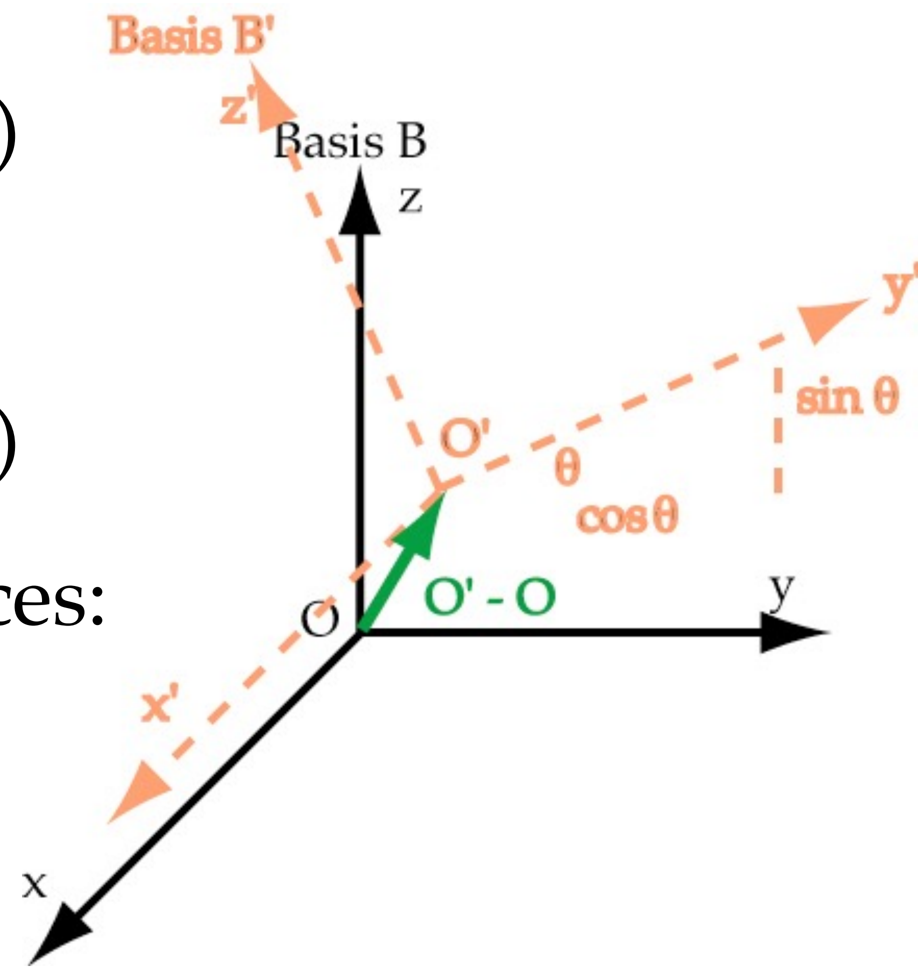
Optimizing

- We can compose the matrices instead:
  - $v' = (HSR)v$
  - Matrix multiplication is associative
- Now cost is: 128 multiplications (for the matrix)
  - 160,000 operations (apply to all the vertices)
- That's why we wanted matrices!

Arbitrary Rotation

- Translate by (O - O′)
- Rotate at O
- Translate by (O′ - O)
- Compose the matrices:

$$M = TRT^{-1}$$



Composition

$$\begin{bmatrix}1 & 0 & 0 & -a\\ 0 & 1 & 0 & -b\\ 0 & 0 & 1 & -c\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & -\sin\theta & 0\\ 0 & \sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & a\\ 0 & 1 & 0 & b\\ 0 & 0 & 1 & c\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\\ w\end{bmatrix}$$

$$=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & -\sin\theta & b\cos\theta - c\sin\theta - b\\ 0 & \sin\theta & \cos\theta & b\sin\theta + c\cos\theta - b\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\\ w\end{bmatrix}$$

- Ugly, but it's a single matrix!