Computer Vision
cmput 428/615

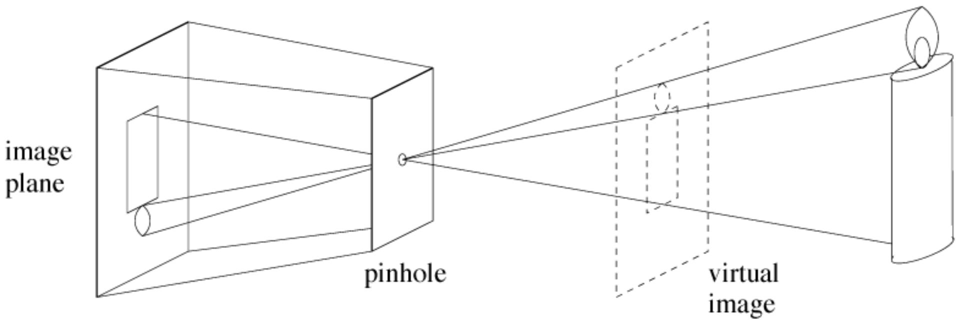
Basic 2D and 3D geometry and Camera models

Martin Jagersand

The equation of projection

How do we develop a consistent mathematical framework for projection calculations?

Intuitively:

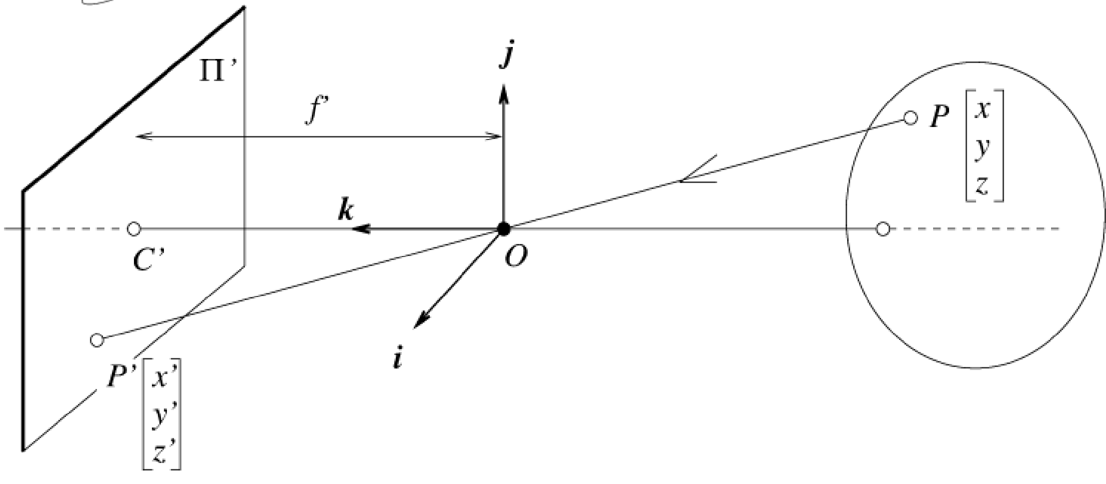


Mathematically:

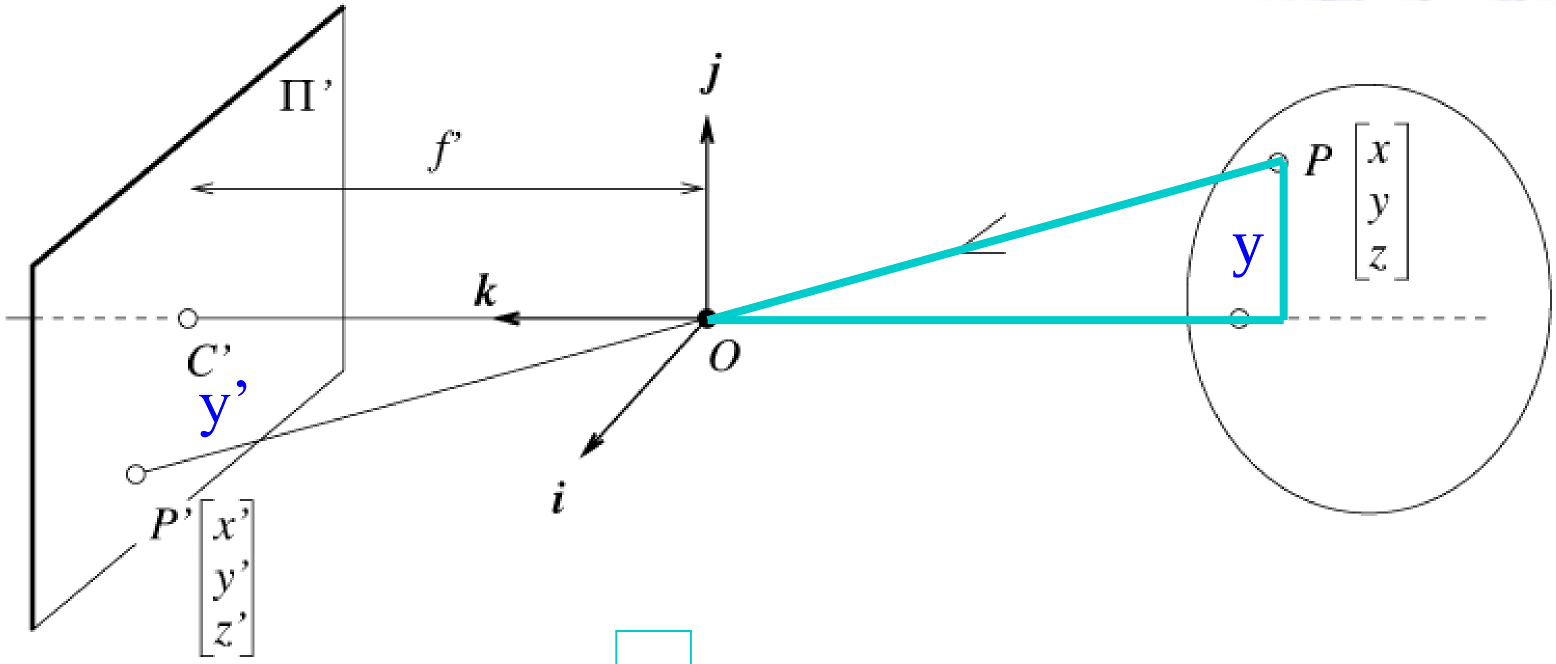
- Cartesian coordinates:

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

- Projectively: x = PX

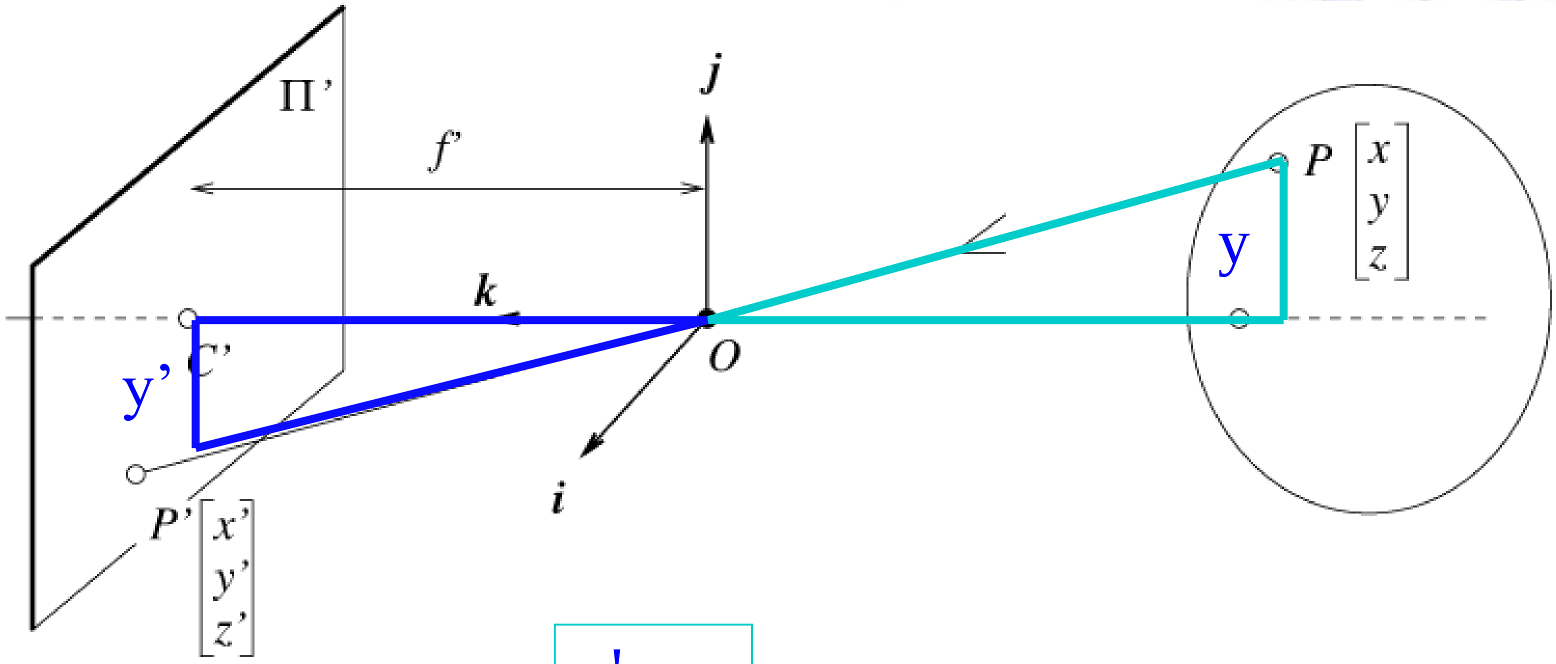


The equation of projection



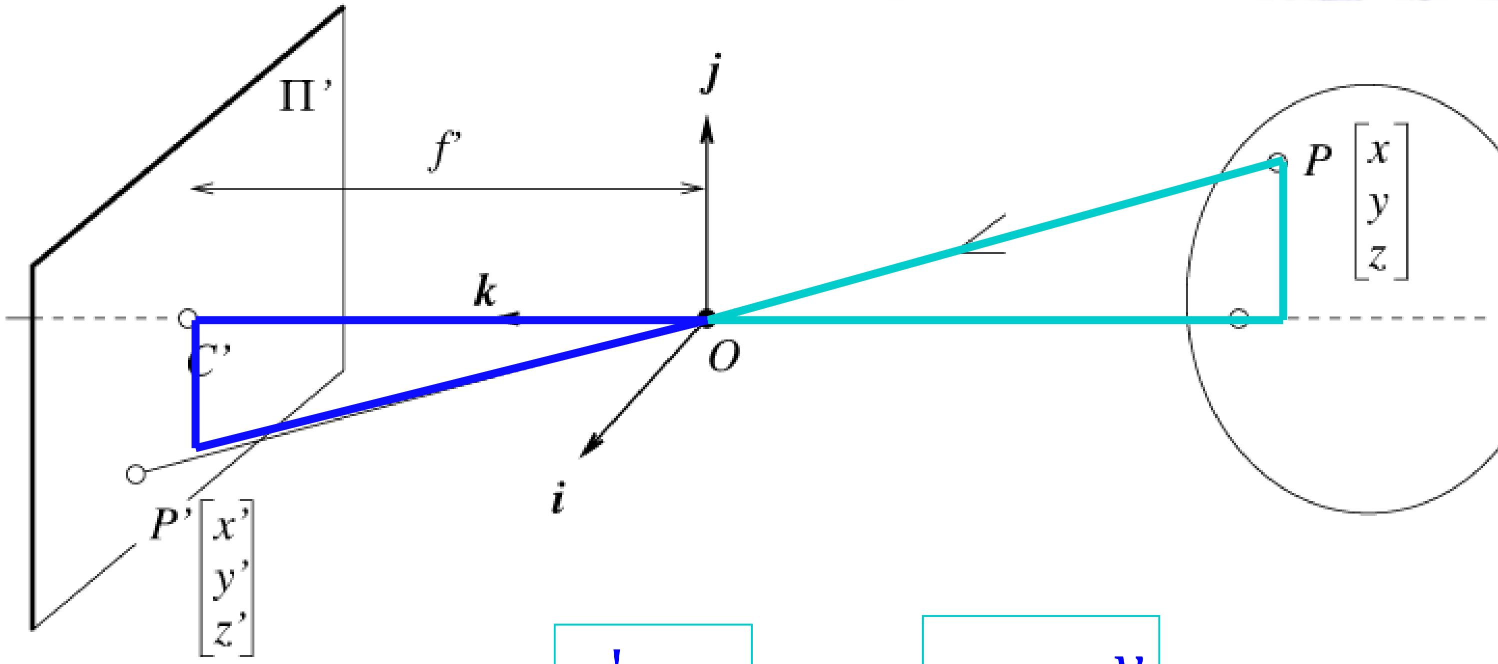
- Similar triangles: $\frac{y}{z}$

The equation of projection



- Similar triangles: $$\frac{y'}{f} = \frac{y}{z}$$

The equation of projection



- Similar triangles: $$\frac{y'}{f} = \frac{y}{z}$$ $$y' = f\frac{y}{z}$$
- Projection eq $$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

The camera matrix

$$(U, V, W) \rightarrow \left(\frac{U}{W}, \frac{V}{W}\right) = (u, v)$$

- Homogenous coordinates for 3D
  - four coordinates for 3D point, 3 for a 2D

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

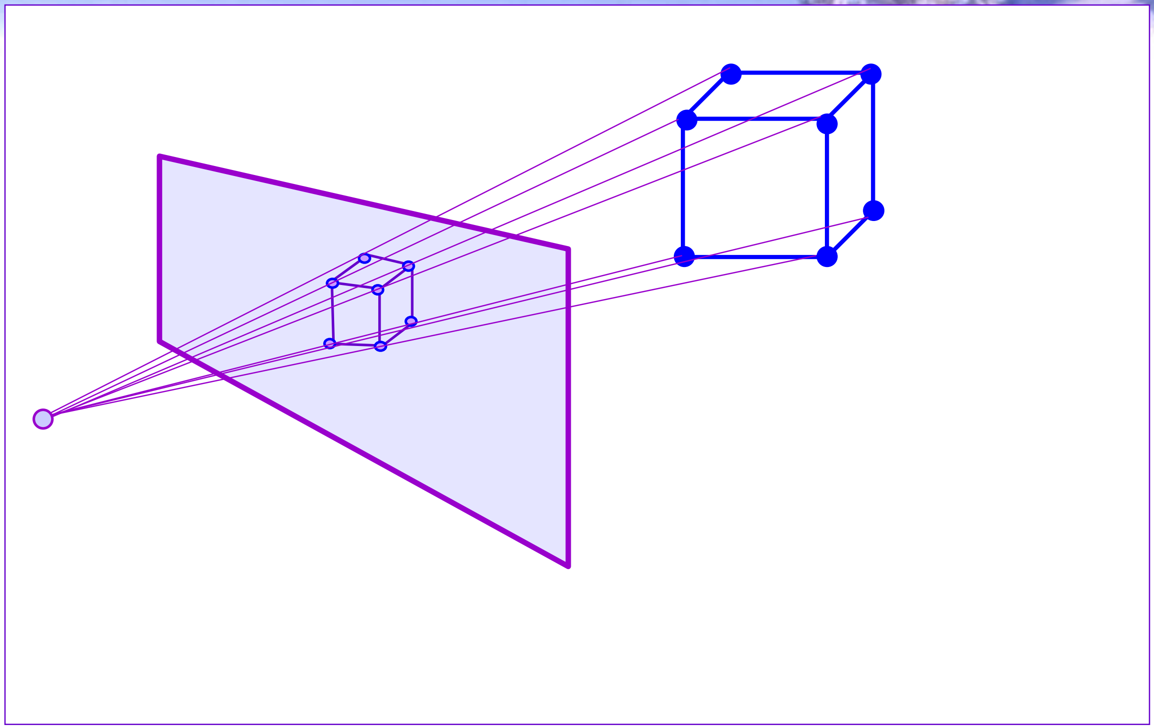
- When coordinate systems are not aligned
  - Projective: x image coordinates, X 3D coord, and P an arbitrary 3x4 matrix
  - x = PX
  - Euclidean
  - x = [R|T]X

Upcoming 2 weeks
Multi-view geometry - resection

- Projection equation

  $x_i = P_i X$

- Resection:
  - $x_i, X \longrightarrow P_i$



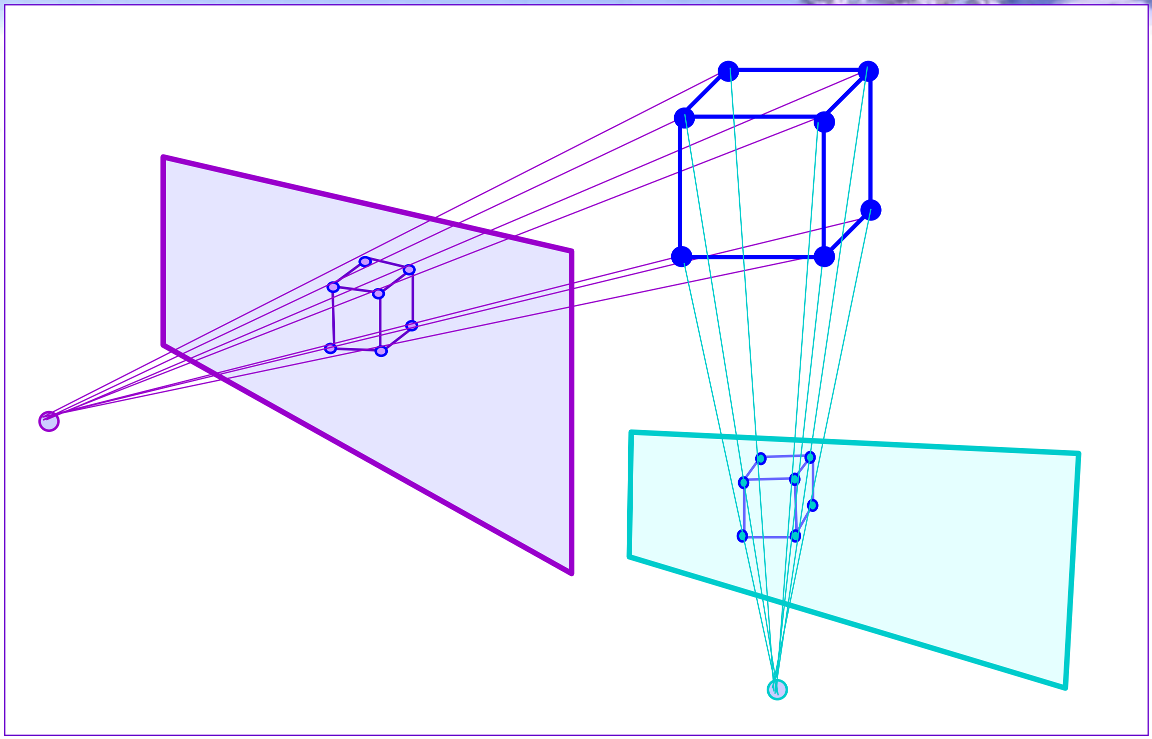
Given image points and 3D points calculate camera projection matrix.

Upcoming 3 weeks
Multi-view geometry - intersection

- Projection equation

  $x_i = P_i X$

- Intersection:
  - $x_i, P_i \longrightarrow X$



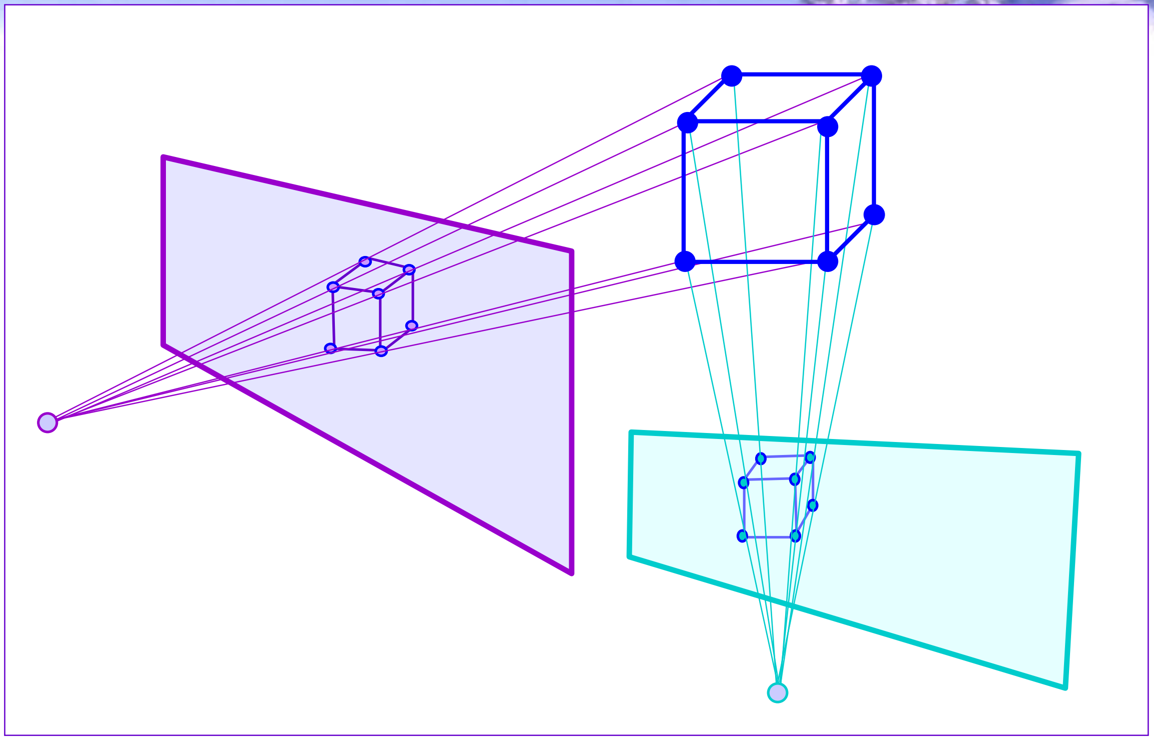
Given image points and camera projections in at least 2 views calculate the 3D points (structure)

Upcoming 4 weeks
Multi-view geometry - SFM

- Projection equation

  $x_i = P_i X$

- Structure from motion (SFM)
  - $x_i \longrightarrow P_i, X$



Given image points in at least 2 views calculate the 3D points (structure) and camera projection matrices (motion)

- Estimate projective structure
- Rectify the reconstruction to metric (autocalibration)

N-view geometry
Affine factorization
(HZ Ch 17, 18)

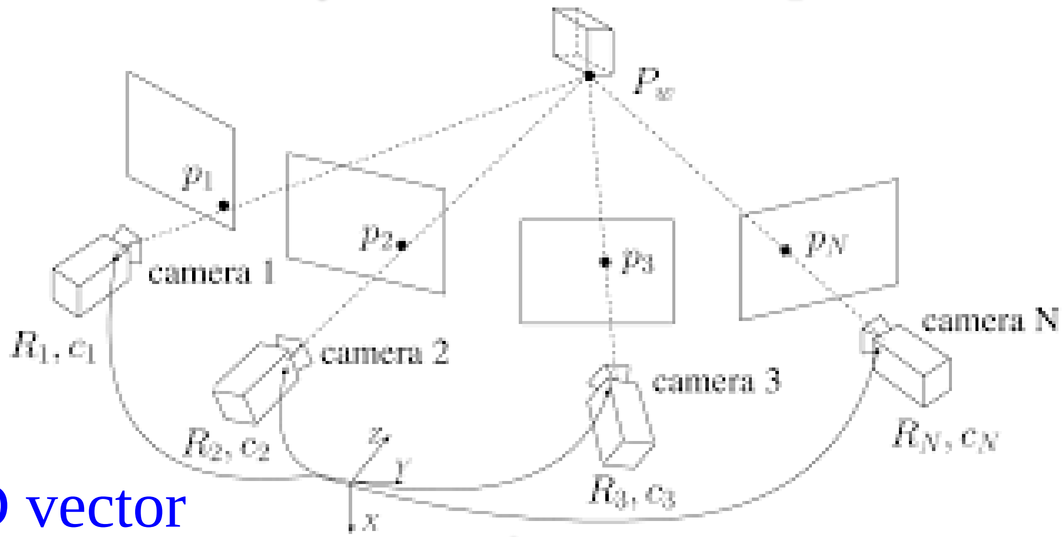
[Carlo Tomasi PhD thesis @CMU > Faculty offer at Stanford]

- Affine camera

$$P_\infty = [M \,|\, \mathbf{t}]$$

- Projection

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = M \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t}$$



M 2x3 matrix; t 2D vector

- *n* points, *m* views: measurement matrix

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{t}$$

$$W = \begin{bmatrix} \tilde{\mathbf{x}}_1^1 & \dots & \tilde{\mathbf{x}}_n^1 \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{x}}_1^m & \dots & \tilde{\mathbf{x}}_n^m \end{bmatrix} = \begin{bmatrix} M^1 \\ \vdots \\ M^m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_n \end{bmatrix}$$

W: Rank 3

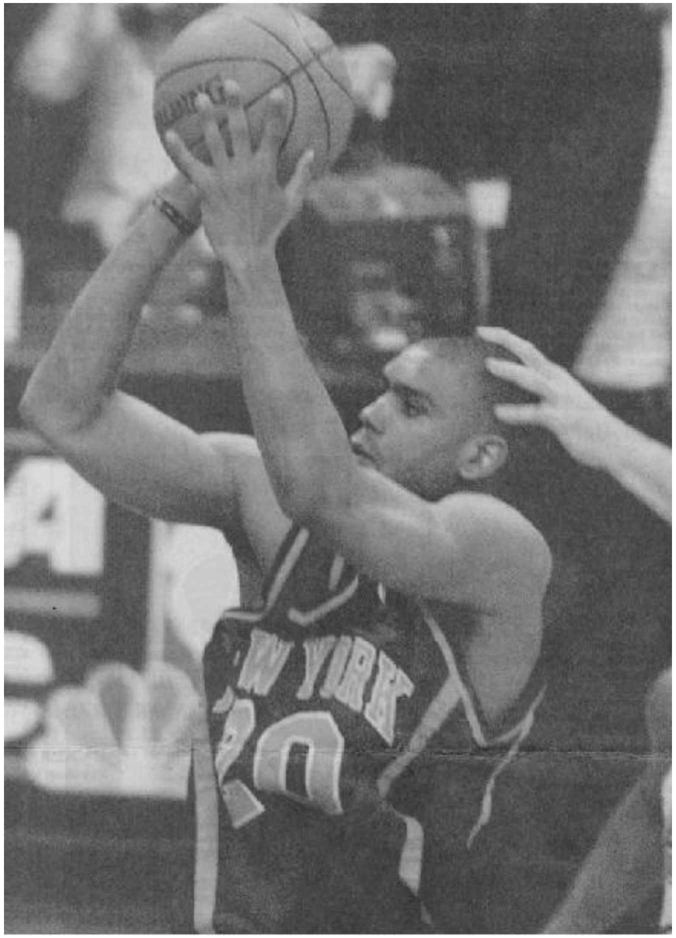
SVD!!

$$W = UDV^T$$

$$\hat{W} = U_{2m\times 3} D_{3\times 3} V^T_{n\times 3} = \hat{M}\hat{X}$$

Assuming isotropic zero-mean Gaussian noise, factorization achieves ML affine reconstruction.

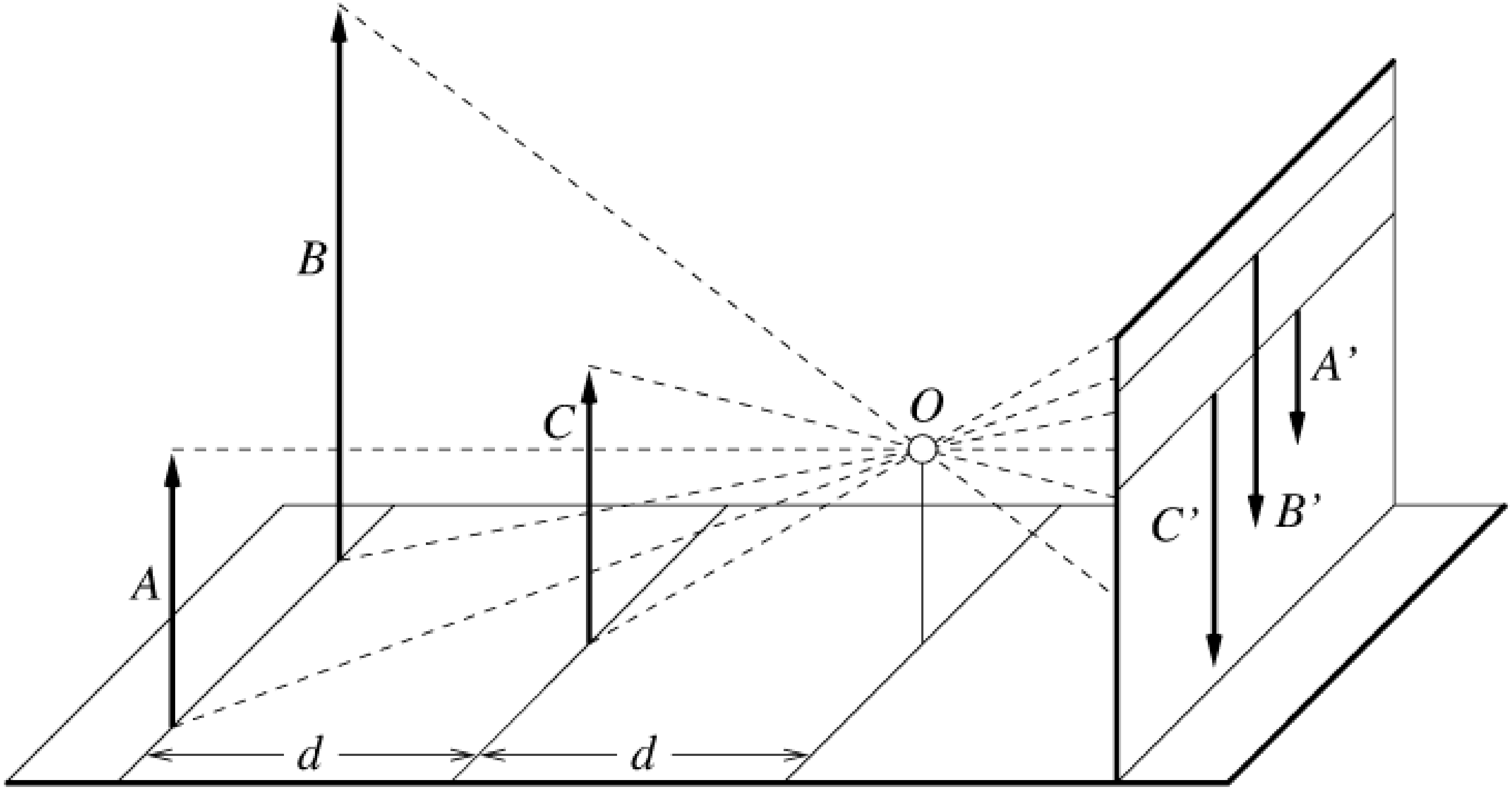
Challenges in Computer Vision: What images *don't* provide



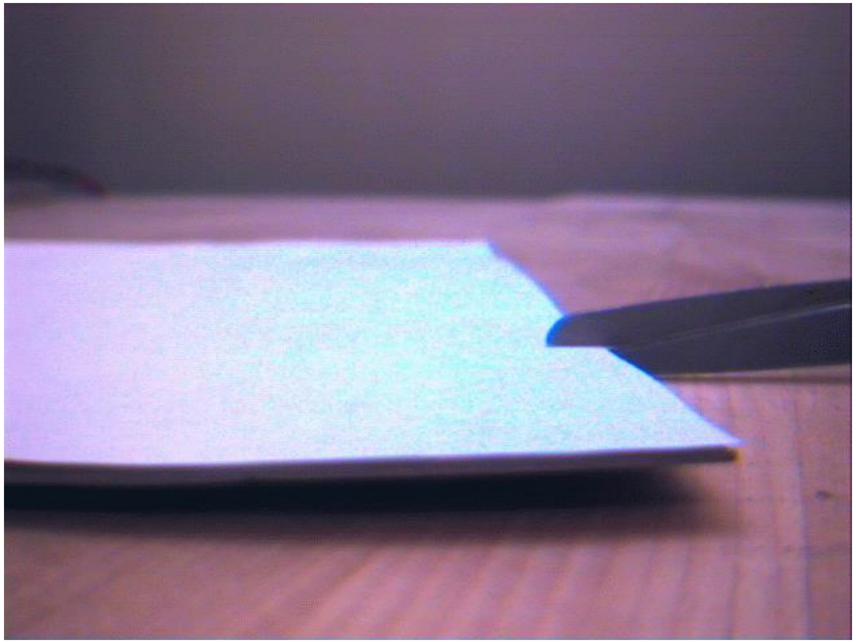
depth

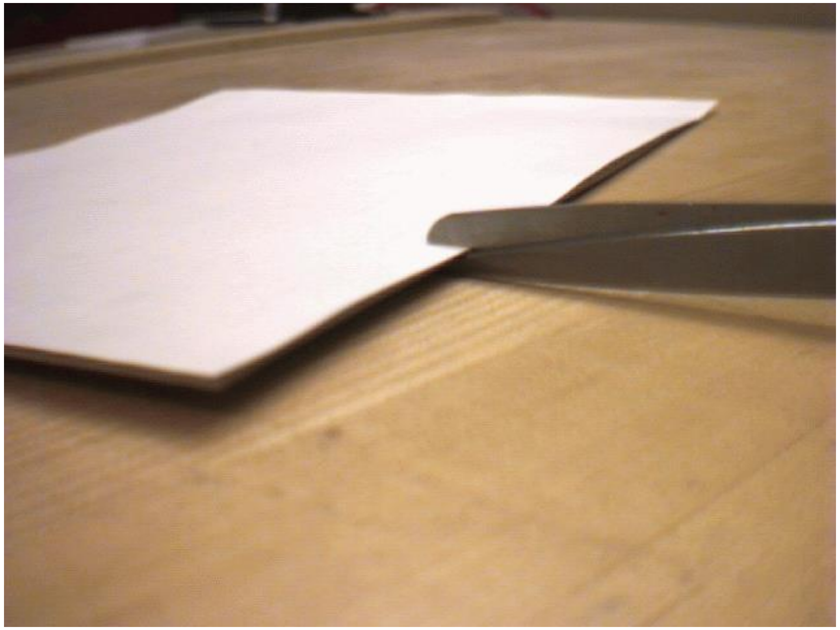
lengths

Distant objects are smaller



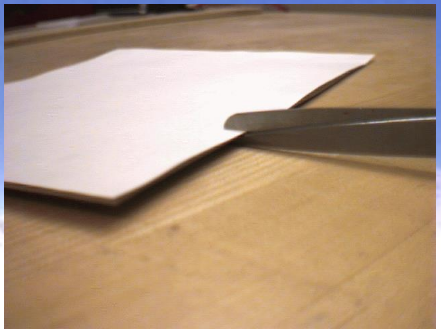
Visual ambiguity





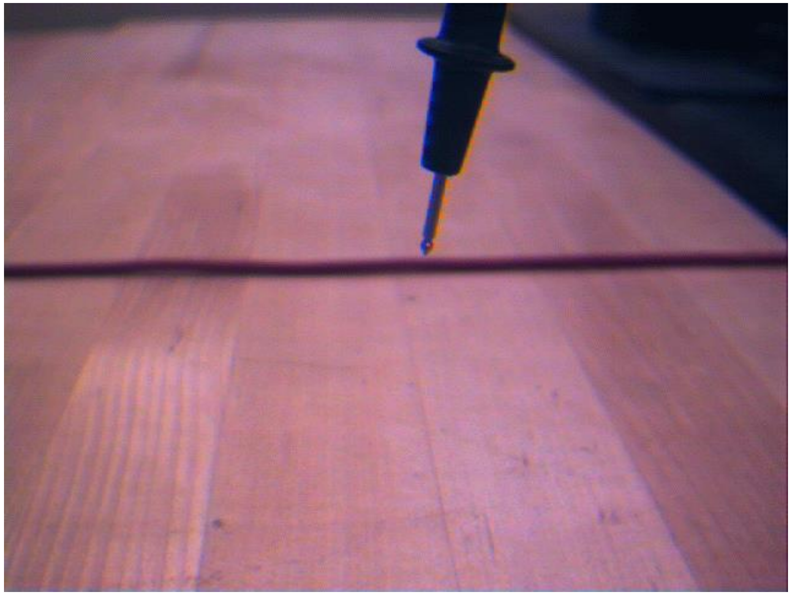
- Will the scissors cut the paper in the middle?

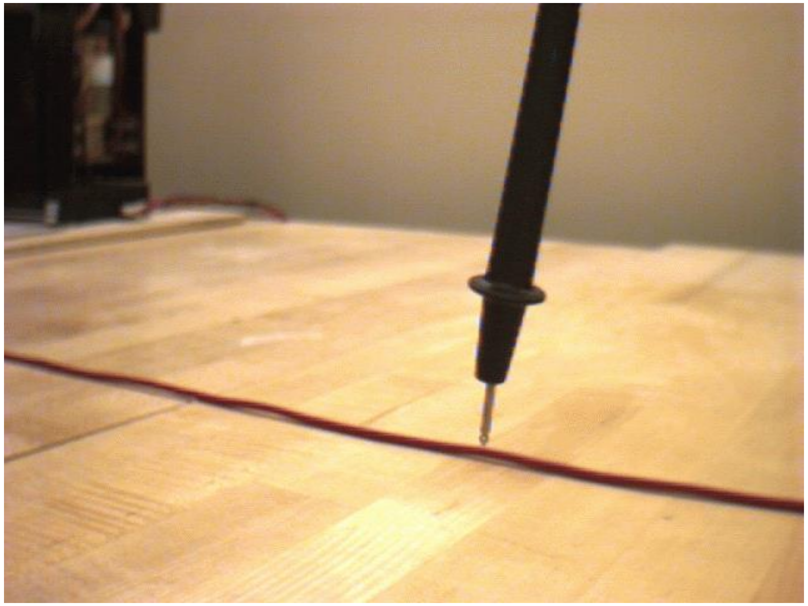
Ambiguity



- Will the scissors cut the paper in the middle? **NO!**

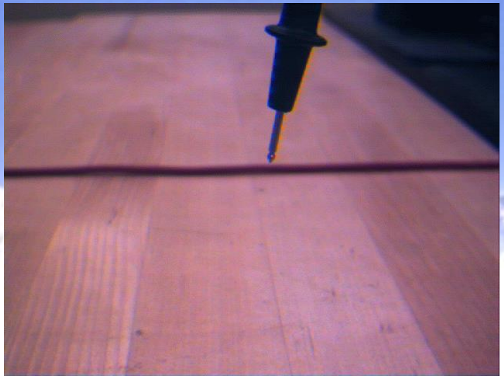
Visual ambiguity

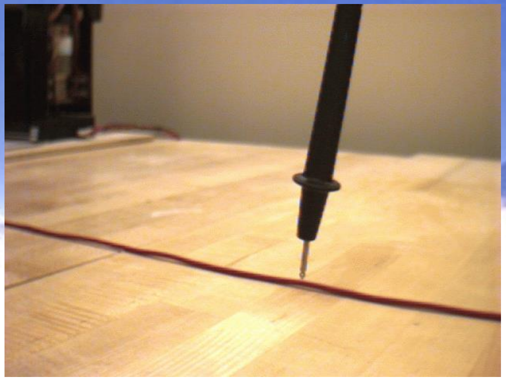


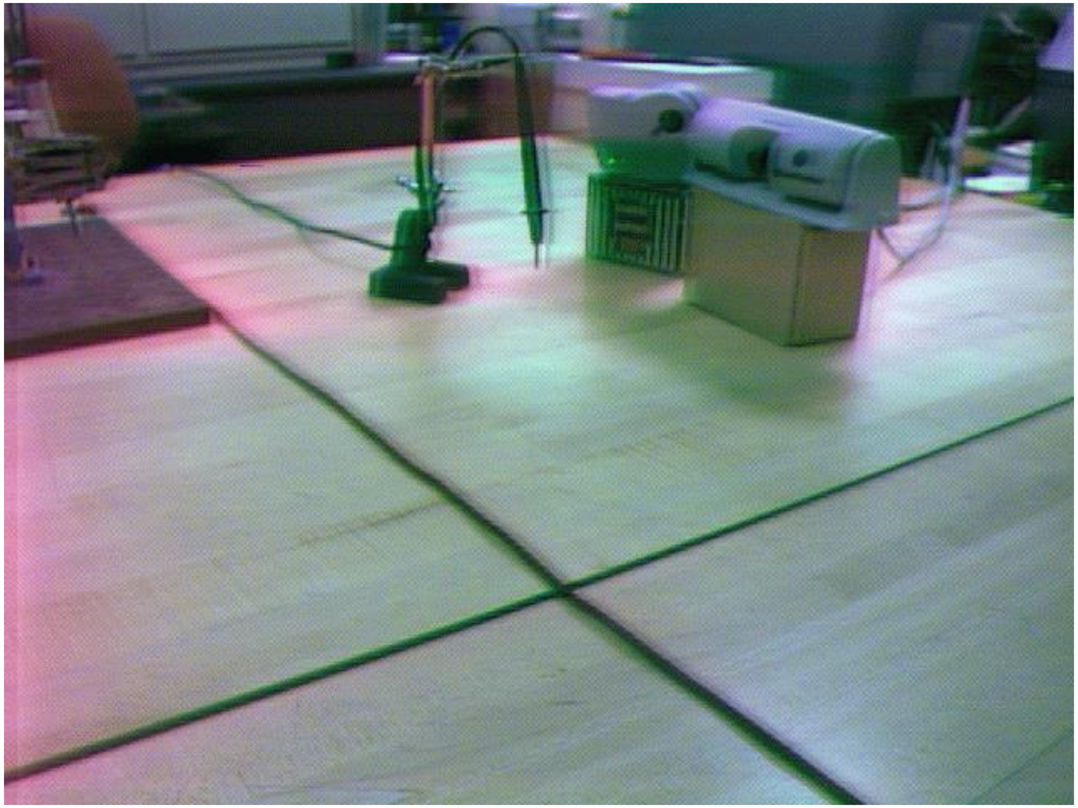


- Is the probe contacting the wire?

Ambiguity

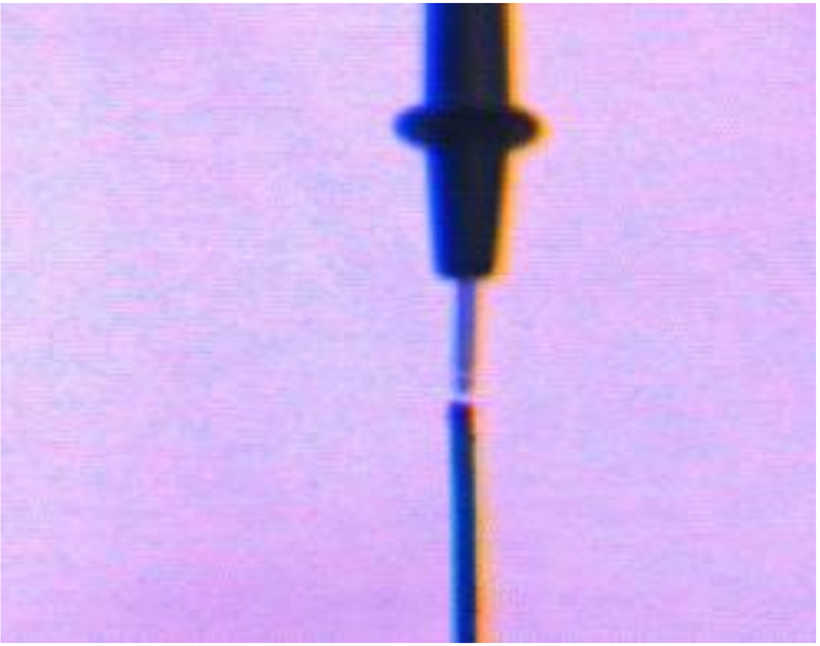


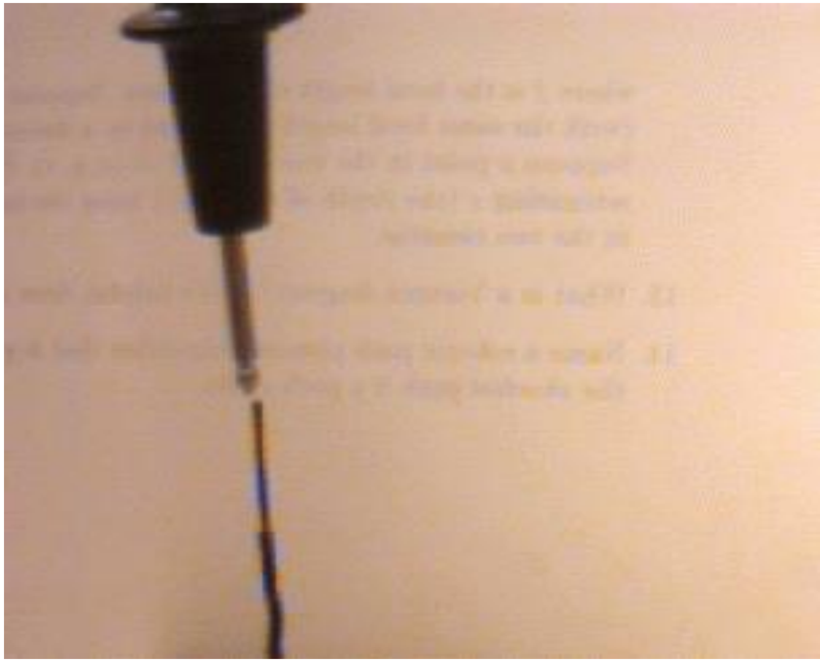




- Is the probe contacting the wire? **NO!**

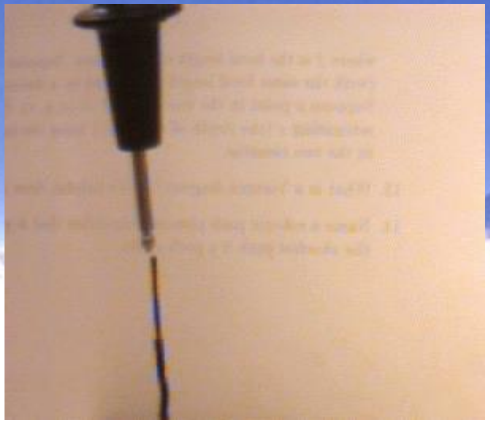
Visual ambiguity





- Is the probe contacting the wire?

Ambiguity

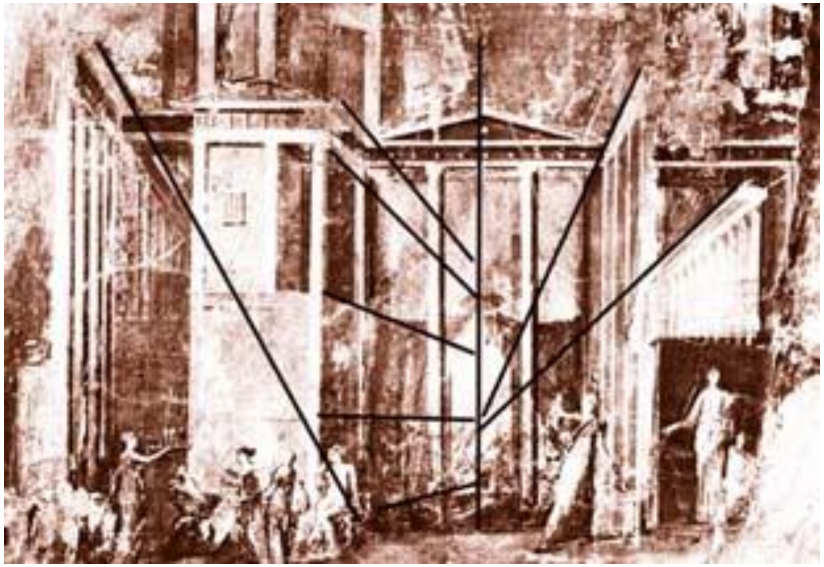


- Is the probe contacting the wire? **NO!**

History of Perspective

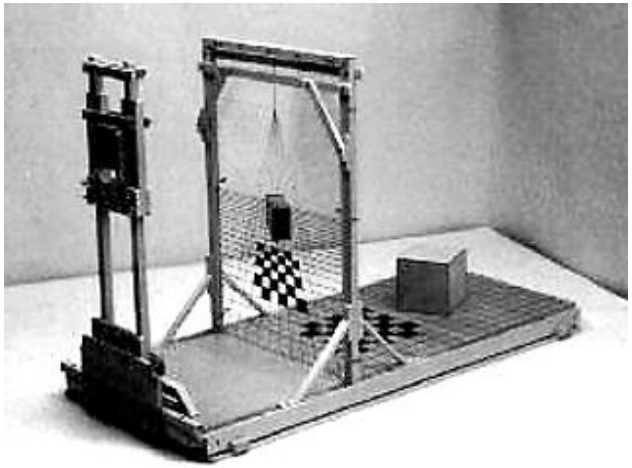
Prehistoric:

Roman



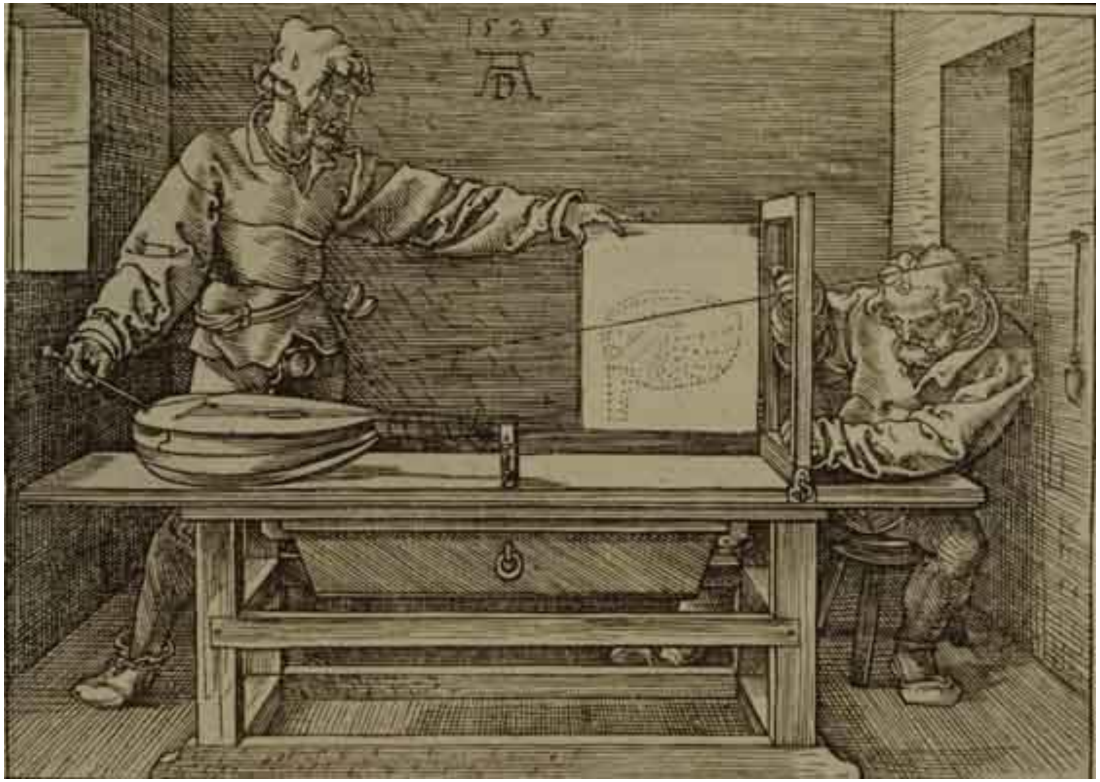
Perspective: Da Vinci

Visualizing perspective: Dürer



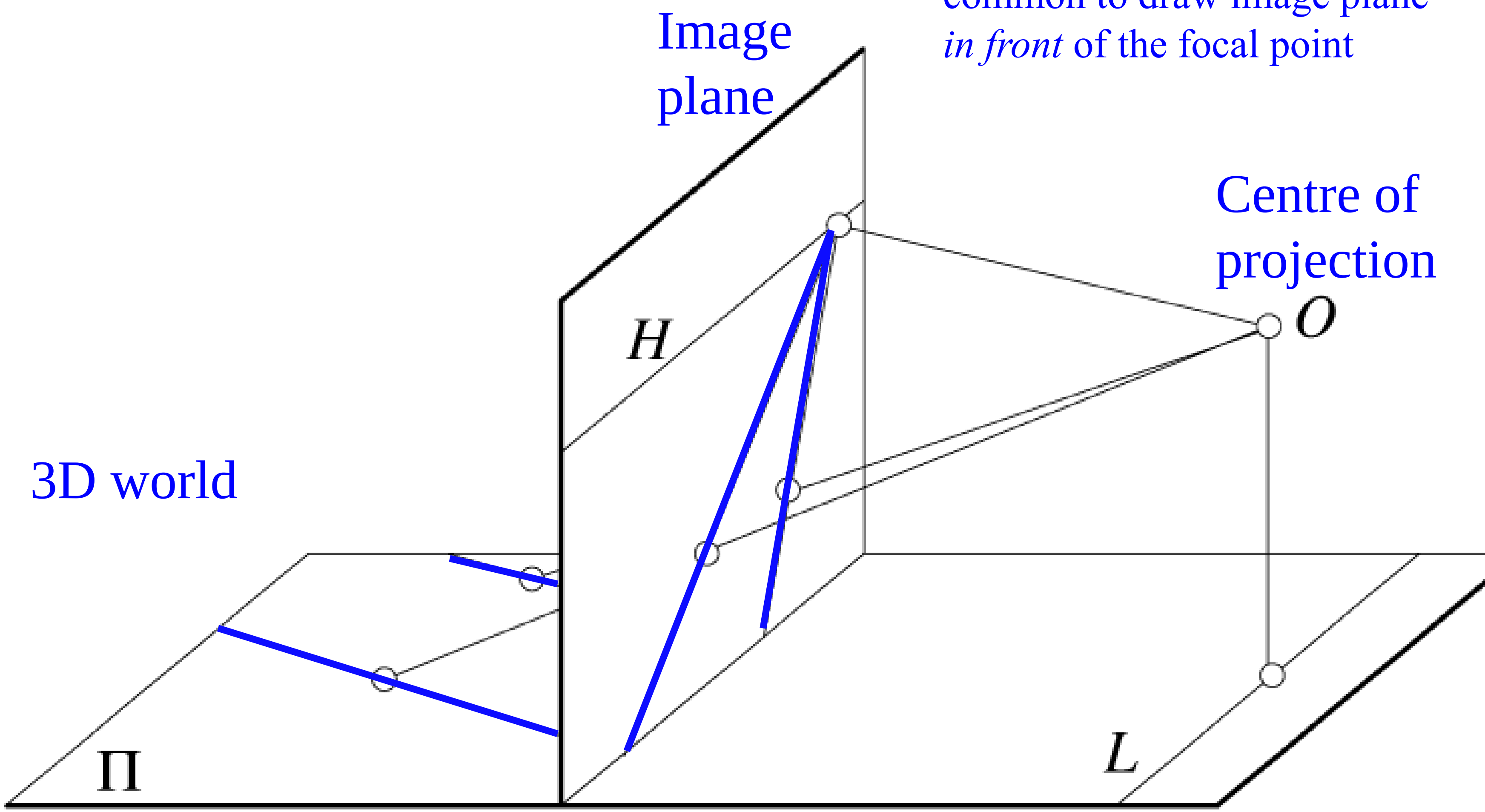
Perspectograph

1500's

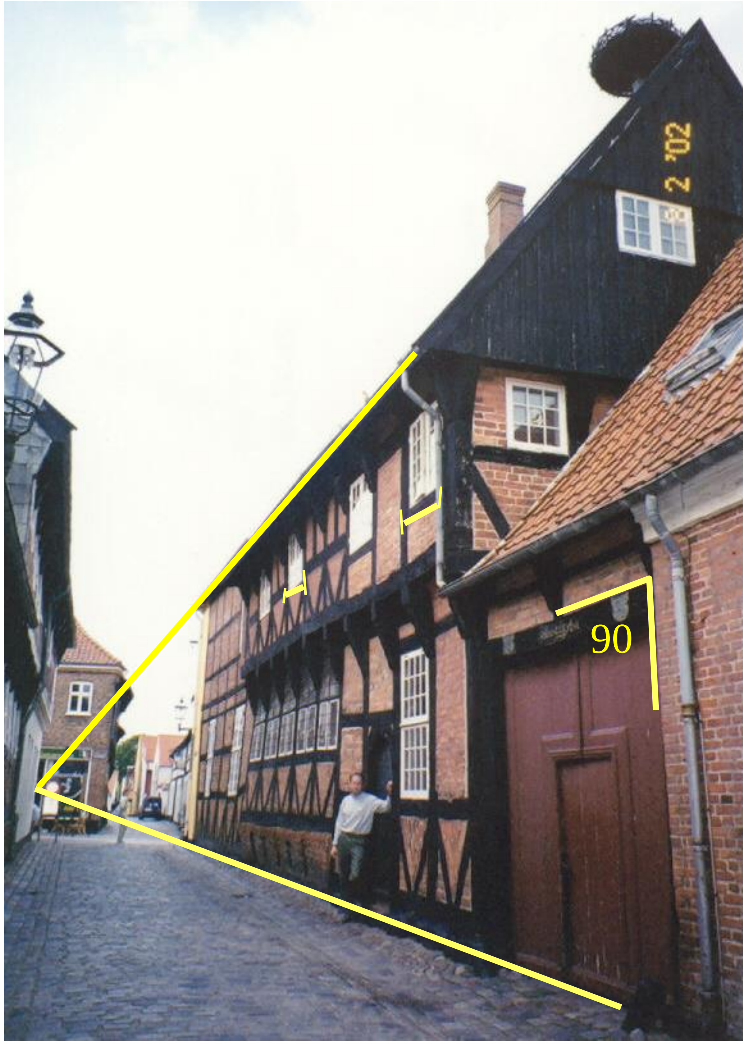


Parallel lines meet

common to draw image plane *in front* of the focal point



Perspective Imaging Properties



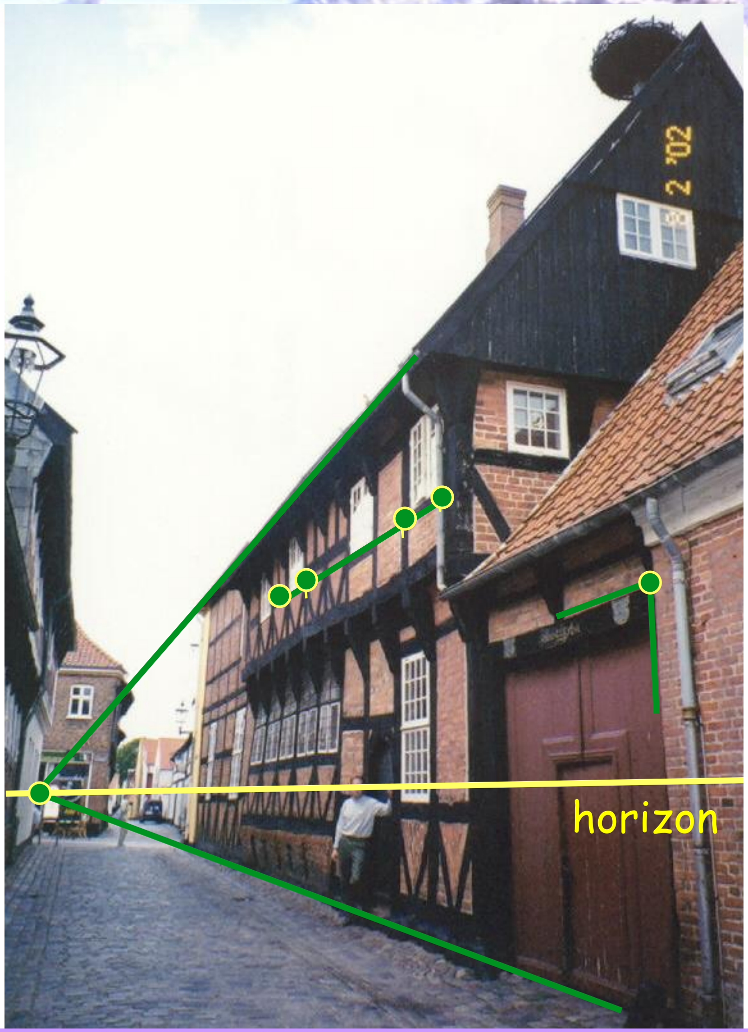
Challenges with measurements in multiple images:

- Distances/angles change
- Ratios of dist/angles change
- Parallel lines intersect

What is preserved?

Invariants:

- Points map to points
- Intersections are preserved
- Lines map to lines
- Collinearity preserved
- Ratios of ratios (cross ratio)
- Horizon

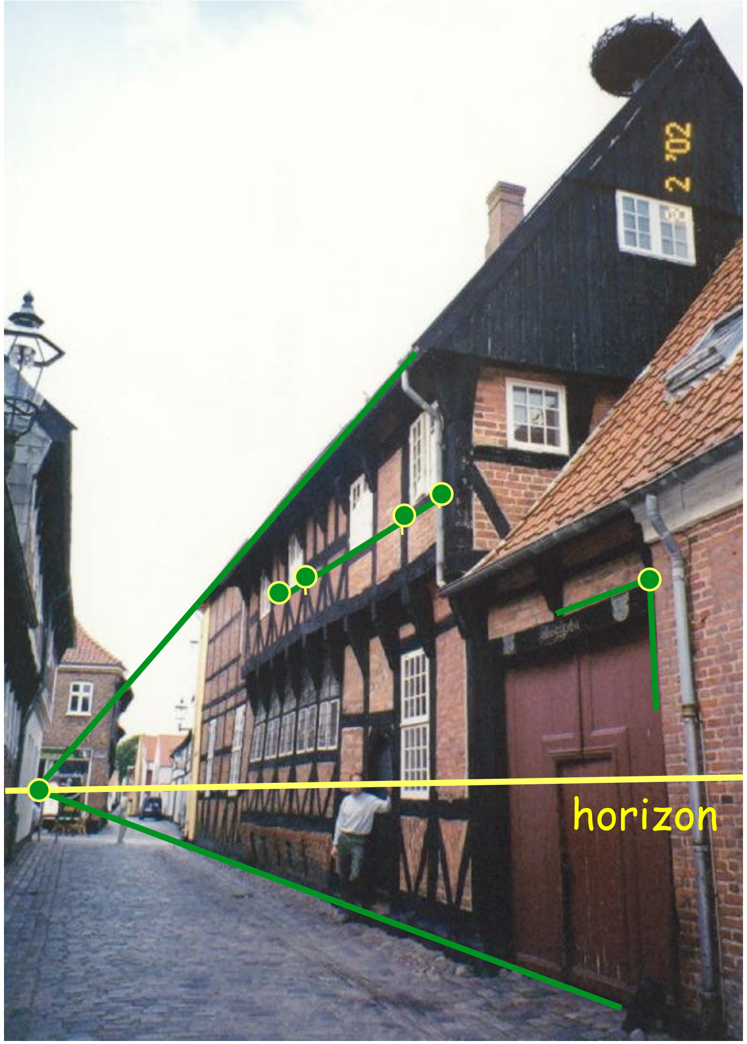


What is a good way to represent imaged geometry?

Vanishing points

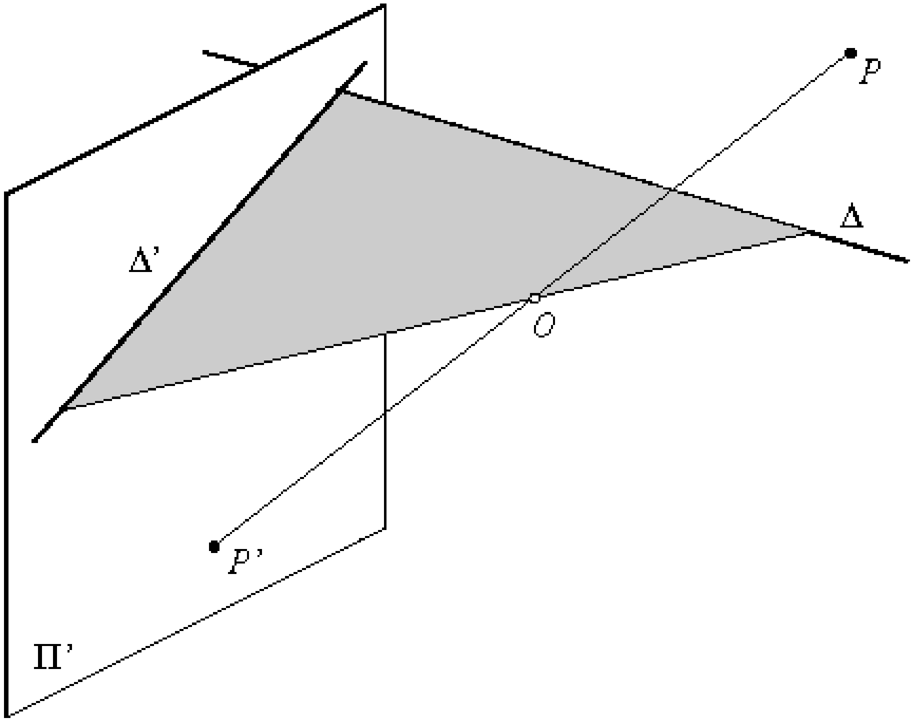
- each set of parallel lines (=direction) meets at a different point
  - The *vanishing point* for this direction
  - How would you show this?

- Sets of parallel lines on the same plane lead to *collinear* vanishing points.
  - The line is called the *horizon* for that plane



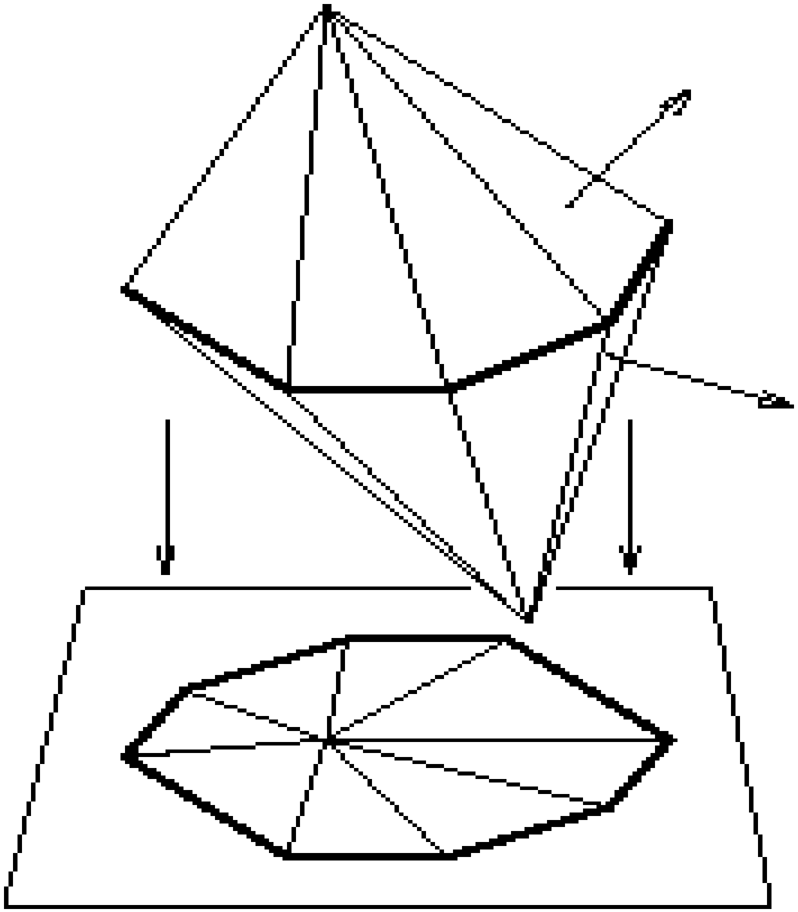
Geometric properties of projection

- Points go to points
- Lines go to lines
- Planes go to whole image
- Polygons go to polygons
- Degenerate cases
  - line through focal point to point
  - plane through focal point to line



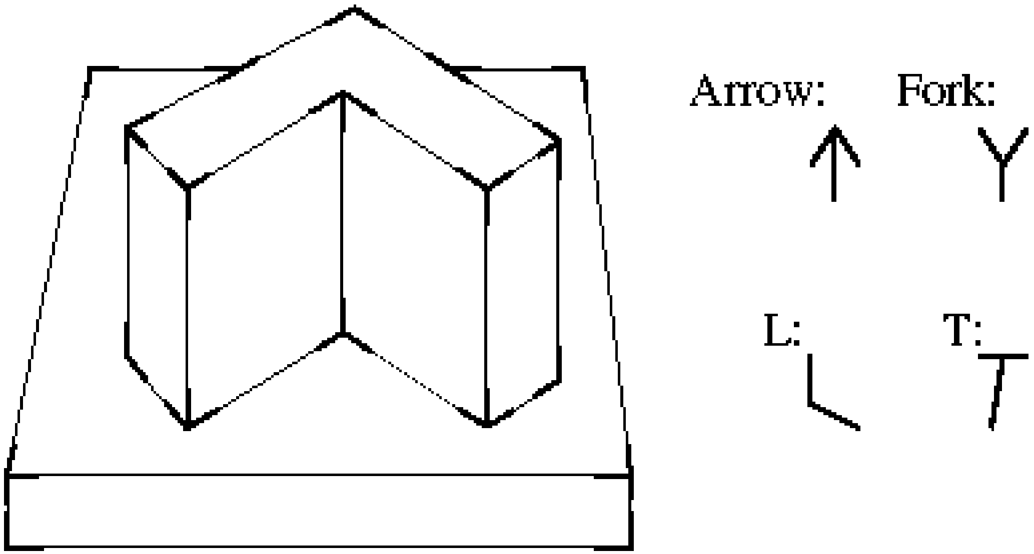
Polyhedra project to polygons

- (because lines project to lines)



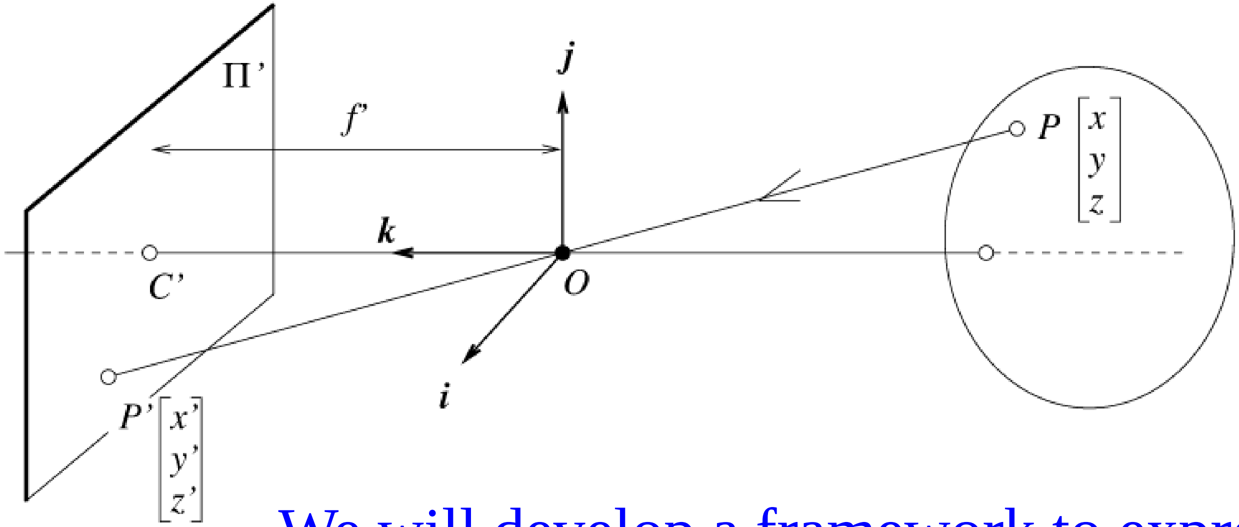
Junctions are constrained

- This leads to a process called "line labelling"
  - one looks for consistent sets of labels, bounding polyhedra
  - disadv - can't get the lines and junctions to label from real images



Back to projection

- Cartesian coordinates: $(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$



We will develop a framework to express projection as x=PX, where x is 2D image projection, P a projection matrix and X is 3D world point.

Basic geometric transformations: Translation

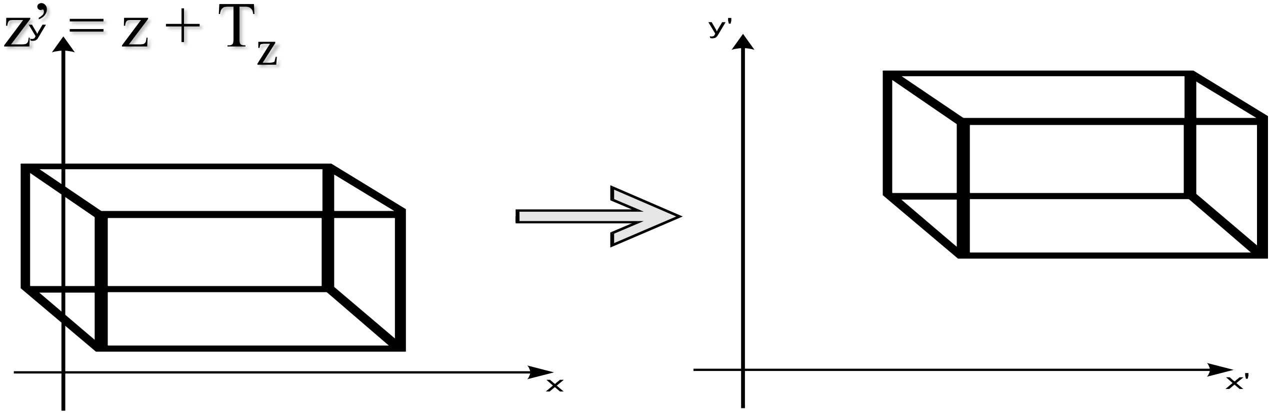
- A translation is a straight line movement of an object from one postion to another.

A point (x,y) is transformed to the point (x’,y’) by adding the translation distances T_x and T_y:

$$x' = x + T_x$$

$$y' = y + T_y$$

$$z' = z + T_z$$

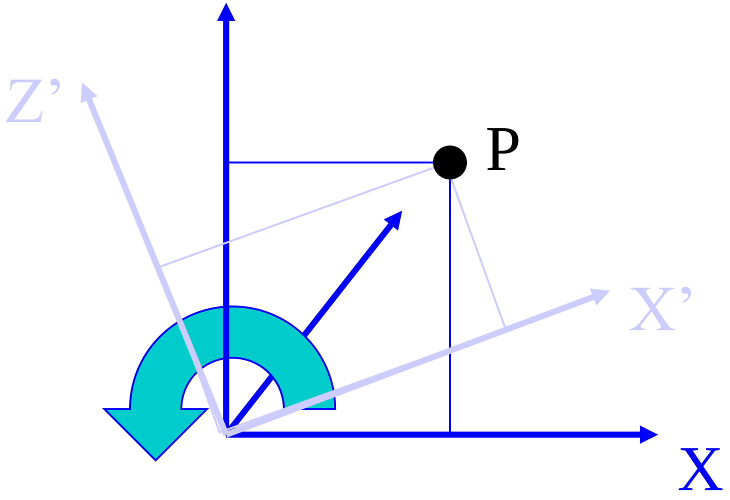


Translation

Coordinate rotation

- Example: Around y-axis

$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\nu & 0 & \sin\nu \\ 0 & 1 & 0 \\ -\sin\nu & 0 & \cos\nu \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}_y \mathbf{p}$$

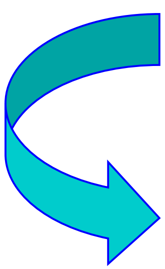


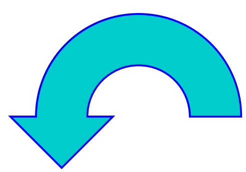
Euler angles

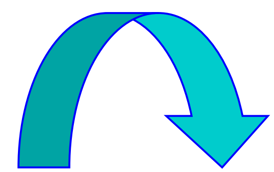
- Note: Successive rotations. Order matters.

$$\mathbf{R} = \mathbf{R_z R_y R_x}$$

$$= \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\nu & 0 & \sin\nu \\ 0 & 1 & 0 \\ -\sin\nu & 0 & \cos\nu \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$

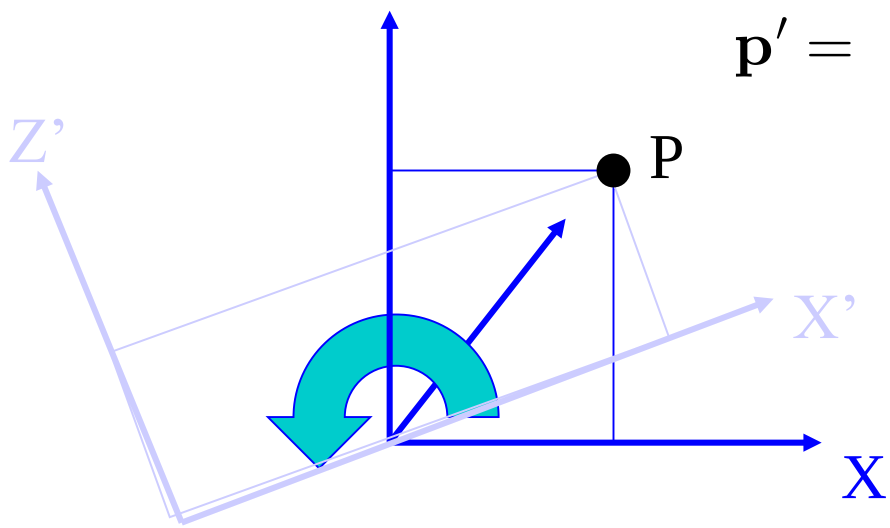






Rotation and translation

- Translation t' in new o' coordinates



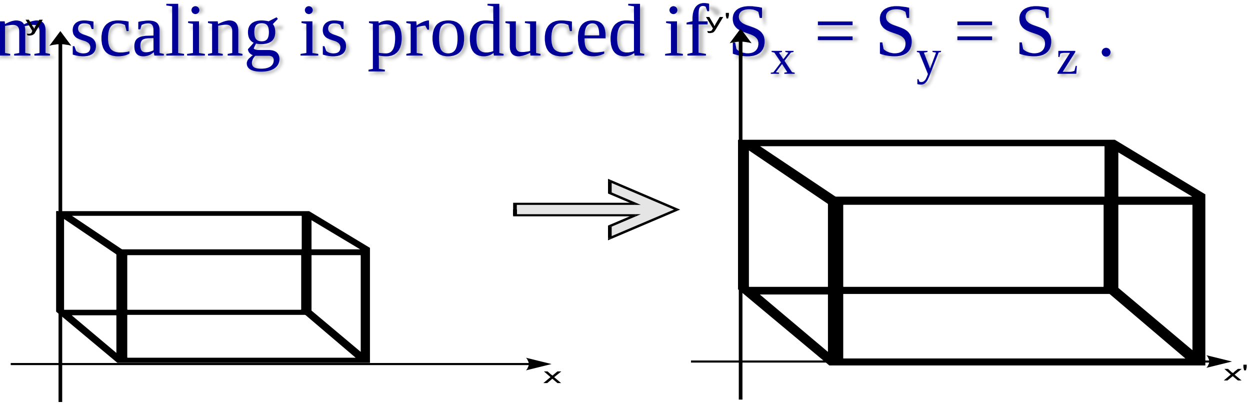
$$\mathbf{p}' = \begin{bmatrix} \cos\nu & 0 & \sin\nu \\ 0 & 1 & 0 \\ -\sin\nu & 0 & \cos\nu \end{bmatrix} \mathbf{p} + \mathbf{t}'$$

Basic transformations
Scaling

- A scaling transformation alters the scale of an object. Suppose a point (x,y) is transformed to the point (x',y') by a scaling with scaling factors S_x and S_y, then:

$$x' = x\,S_x$$
$$y' = y\,S_y$$
$$z' = z\,S_z$$

- A uniform scaling is produced if $S_x = S_y = S_z$.



Scaling about the Origin

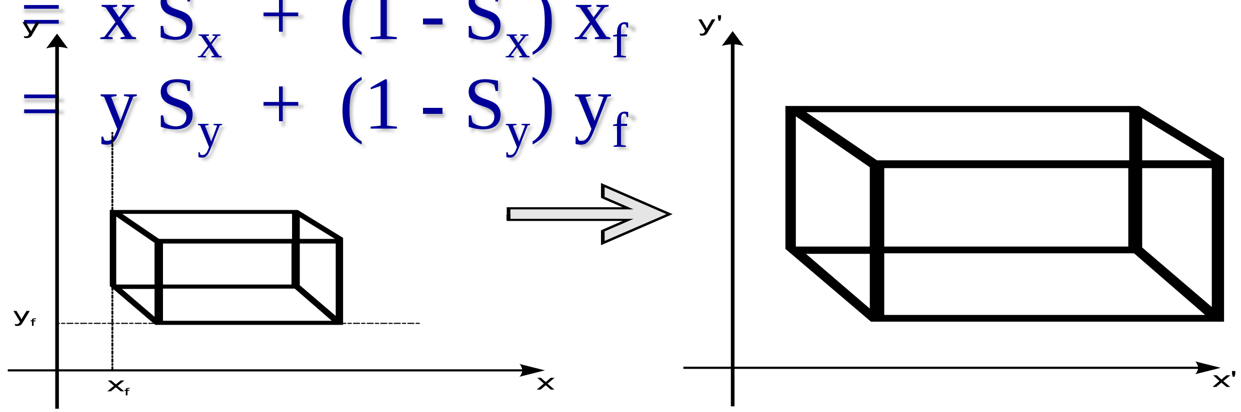
Basic transformations
Scaling

The previous scaling transformation leaves the origin unaltered. If the point (x_f,y_f) is to be the fixed point, the transformation is:

$$x' = x_f + (x - x_f)\,S_x$$
$$y' = y_f + (y - y_f)\,S_y$$

This can be rearranged to give:

$$x' = x\,S_x + (1 - S_x)\,x_f$$
$$y' = y\,S_y + (1 - S_y)\,y_f$$



Scaling about (x_f,y_f)

Affine Geometric Transforms

In general, a point in n-D space transforms by

P' = rotate(point) + translate(point)

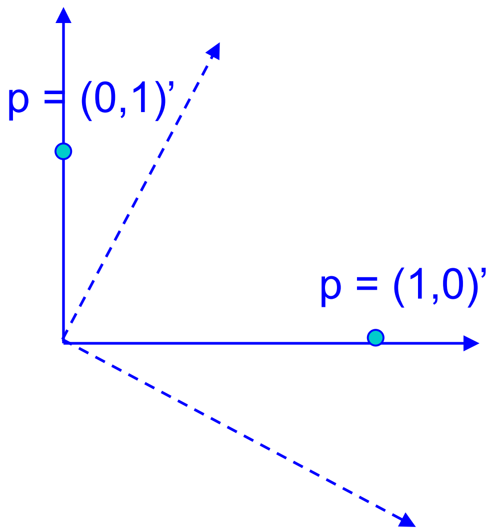
In 2-D space, this can be written as a matrix equation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} Cos(\theta) & -Sin(\theta) \\ Sin(\theta) & Cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} tx \\ ty \end{pmatrix}$$

In 3-D space (or n-D), this can generalized as a matrix equation:

$p' = R\,p + T$ or $p = R^t\,(p' - T)$

A Simple 2-D Example



Suppose we rotate the coordinate system through 45 degrees (note that this is measured relative to the *rotated* system!

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} Cos(\pi/4) & -Sin(\pi/4) \\ Sin(\pi/4) & Cos(\pi/4) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} Cos(\pi/4) \\ Sin(\pi/4) \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} Cos(\pi/4) & -Sin(\pi/4) \\ Sin(\pi/4) & Cos(\pi/4) \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} -Sin(\pi/4) \\ Cos(\pi/4) \end{pmatrix}$$

Matrix representation and Homogeneous coordinates

- Often need to combine several transformations to build the total transformation.
- So far using affine transforms need both add and multiply
- Good if all transformations could be represented as matrix multiplications then the combination of transformations simply involves the multiplication of the respective matrices
- As translations do not have a 2 x 2 matrix representation, we introduce homogeneous coordinates to allow a 3 x 3 matrix representation.

How to translate a 2D point:

- Old way: $$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$
- New way:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Relationship between 3D homogeneous and inhomogeneous

- The Homogeneous coordinate corresponding to the point (x,y,z) is the triple (x_h, y_h, z_h, w) where:

$$x_h = wx$$
$$y_h = wy$$
$$z_h = wz$$

  We can (initially) set $w = 1$.

- Suppose a point P = (x,y,z,1) in the homogeneous coordinate system is mapped to a point P' = (x',y',z',1) by a transformations, then the transformation can be expressed in matrix form.

Matrix representation and Homogeneous coordinates

- For the basic transformations we have:
  - Translation

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

  - Scaling

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

Geometric Transforms

Using the idea of homogeneous transforms, we can write:

$$p' = \begin{pmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{pmatrix} p$$

R and T both require 3 parameters.

$$\mathrm{R} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\nu & 0 & \sin\nu \\ 0 & 1 & 0 \\ -\sin\nu & 0 & \cos\nu \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$

Geometric Transforms

If we compute the matrix inverse, we find that

$$p = \begin{pmatrix} R' & -R'T \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} p'$$

R and T both require 3 parameters. These correspond to the 6 extrinsic parameters needed for camera calibration

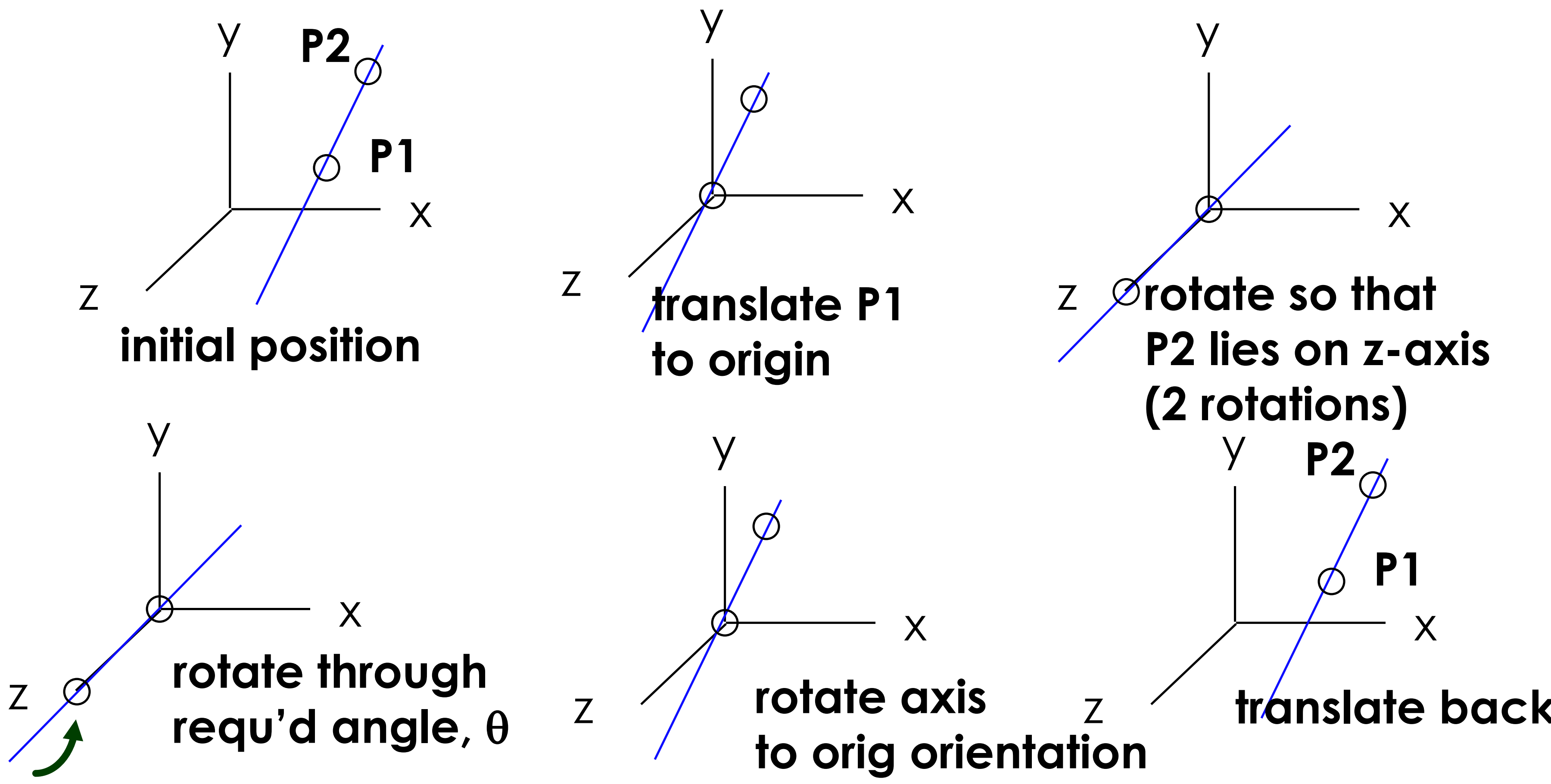
Recall inhomogenous inversion:

p' = R p + T or p = R^t (p' – T)

Rotation about a Specified Axis

- It is useful to be able to rotate about any axis in 3D space
- This is achieved by composing 7 elementary transformations (next slide)

Rotation through θ about Specified Axis

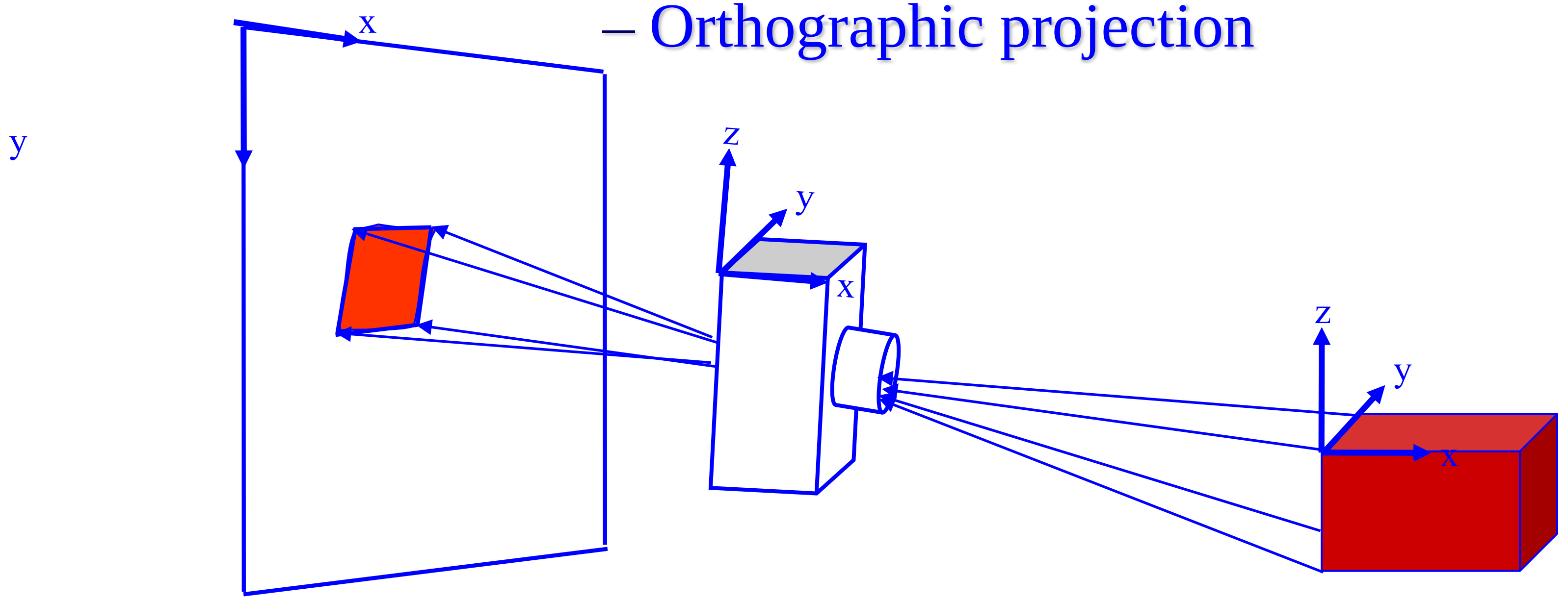


Comparison:

- Homogeneous coordinates
  - Rotations and translations are represented in a uniform way
  - Successive transforms are composed using matrix products: y = Pn*..*P2*P1*x
- Affine coordinates
  - Non-uniform representations: y = Ax + b
  - Difficult to keep track of separate elements

Camera models and projections Geometry part 2.

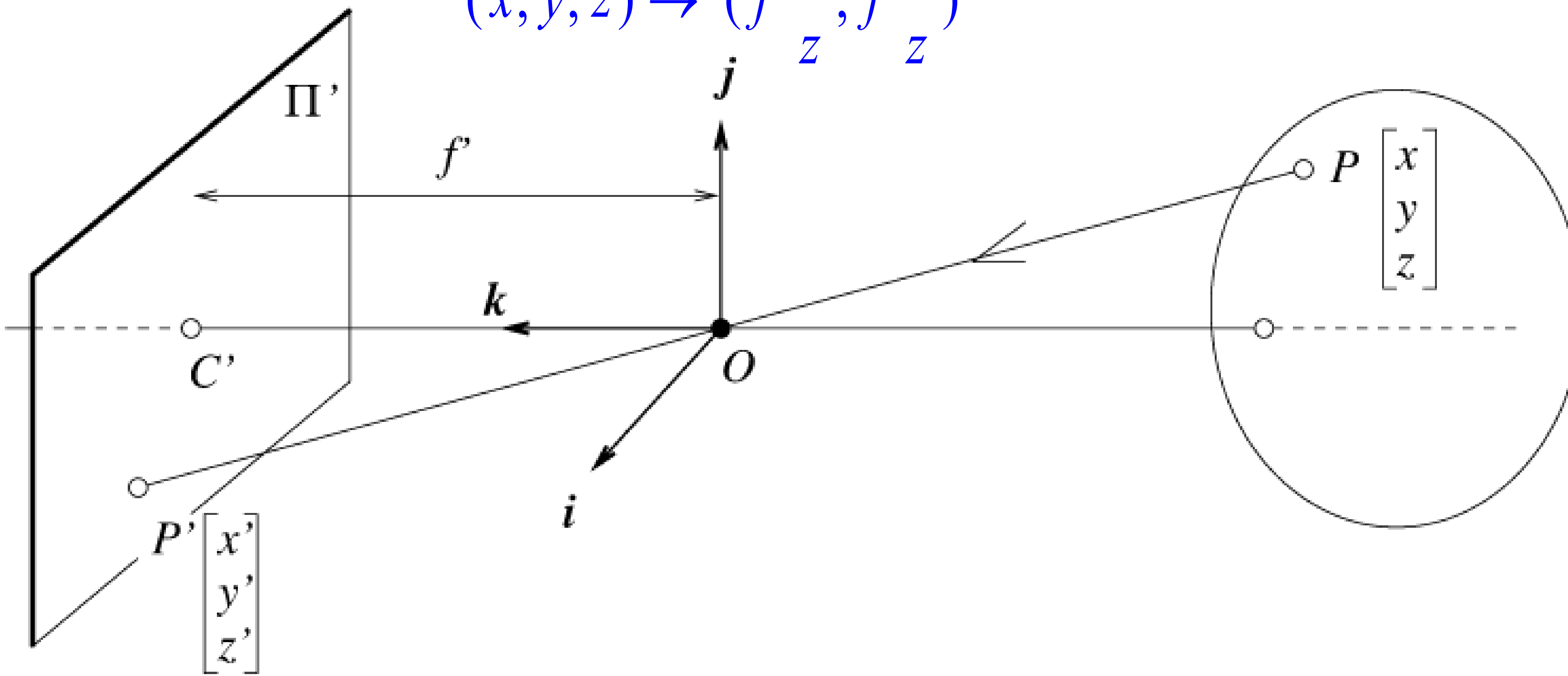
- Using geometry and homogeneous transforms to describe:
  - Perspective projection
  - Weak perspective projection
  - Orthographic projection



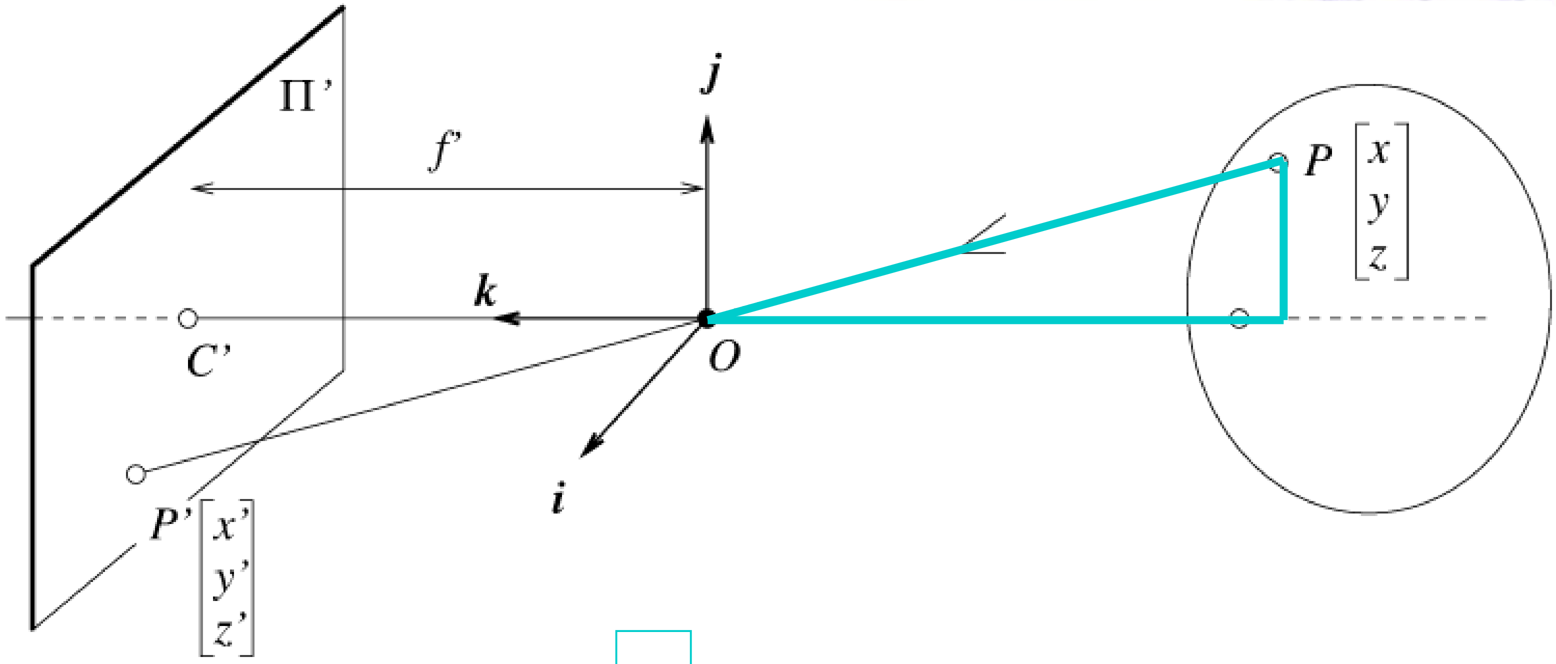
The equation of perspective projection

- Cartesian coordinates:
  - We have, by similar triangles, that (x, y, z) -> (f x/z, f y/z, -f)
  - Drop the third coordinate, and get

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$



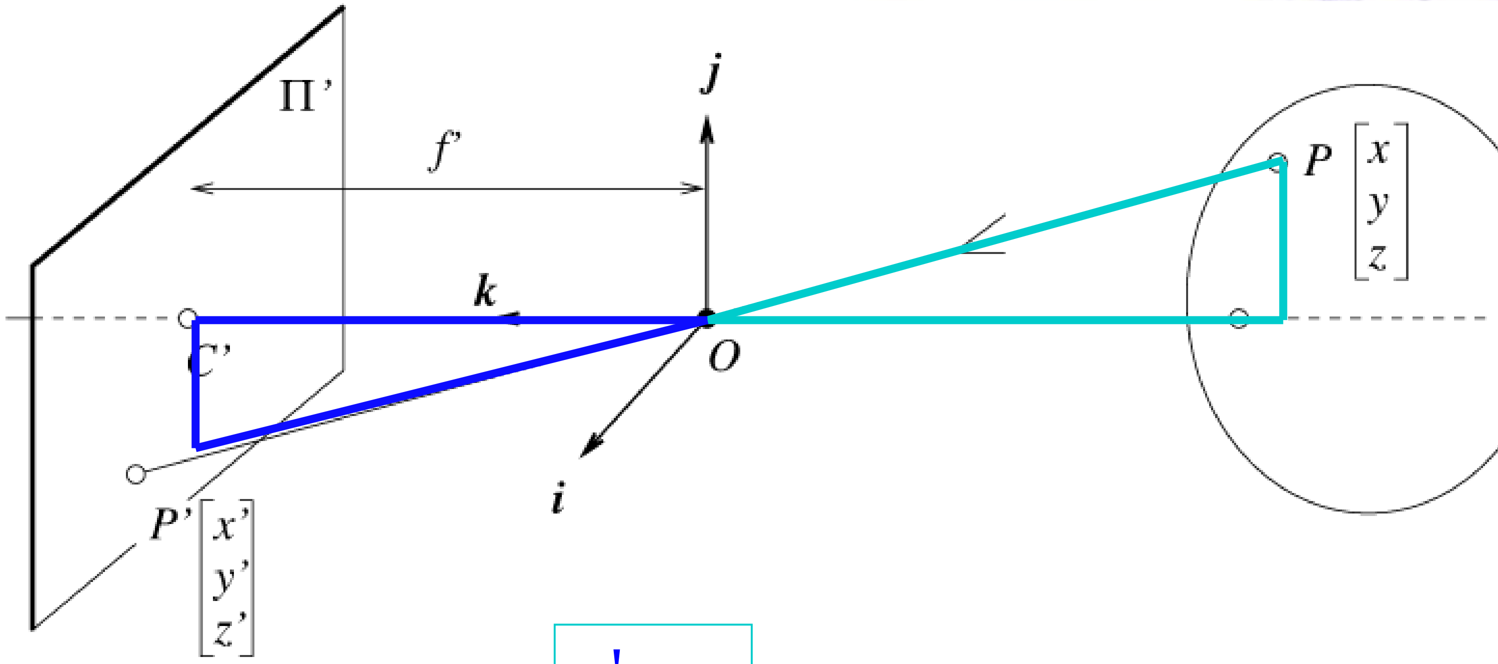
The equation of projection



- Similar triangles:

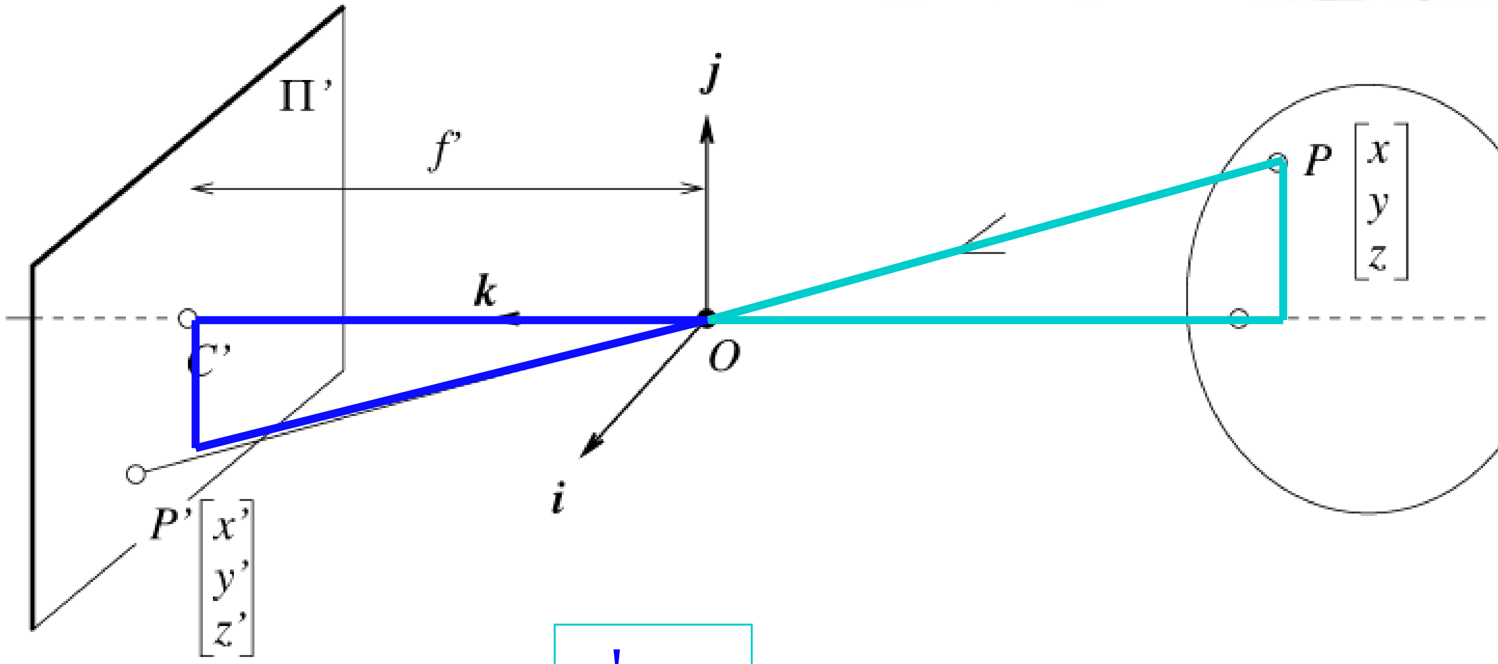
$$\frac{y}{z}$$

The equation of projection



- Similar triangles: $$\frac{y'}{f} = \frac{y}{z}$$

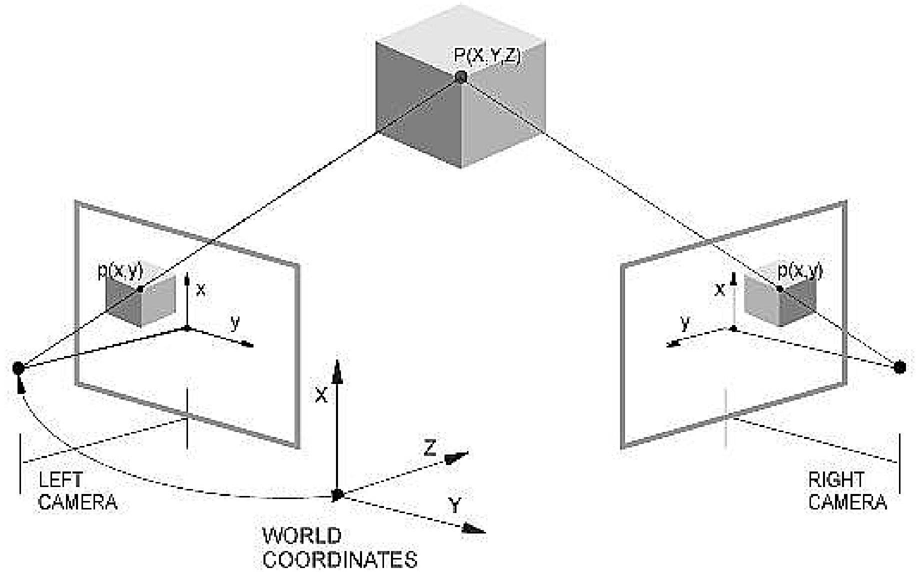
The equation of projection



- Similar triangles: $$\frac{y'}{f} = \frac{y}{z}$$
- Projection eq $$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

Stereo Vision

- **GOAL:** Passive 2-camera system for triangulating 3D position of points in space to generate a depth map of a world scene.
- Humans use stereo vision to obtain depth



Stereo depth calculation: Simple case, aligned cameras

DISPARITY= (XL - XR)

Similar triangles:

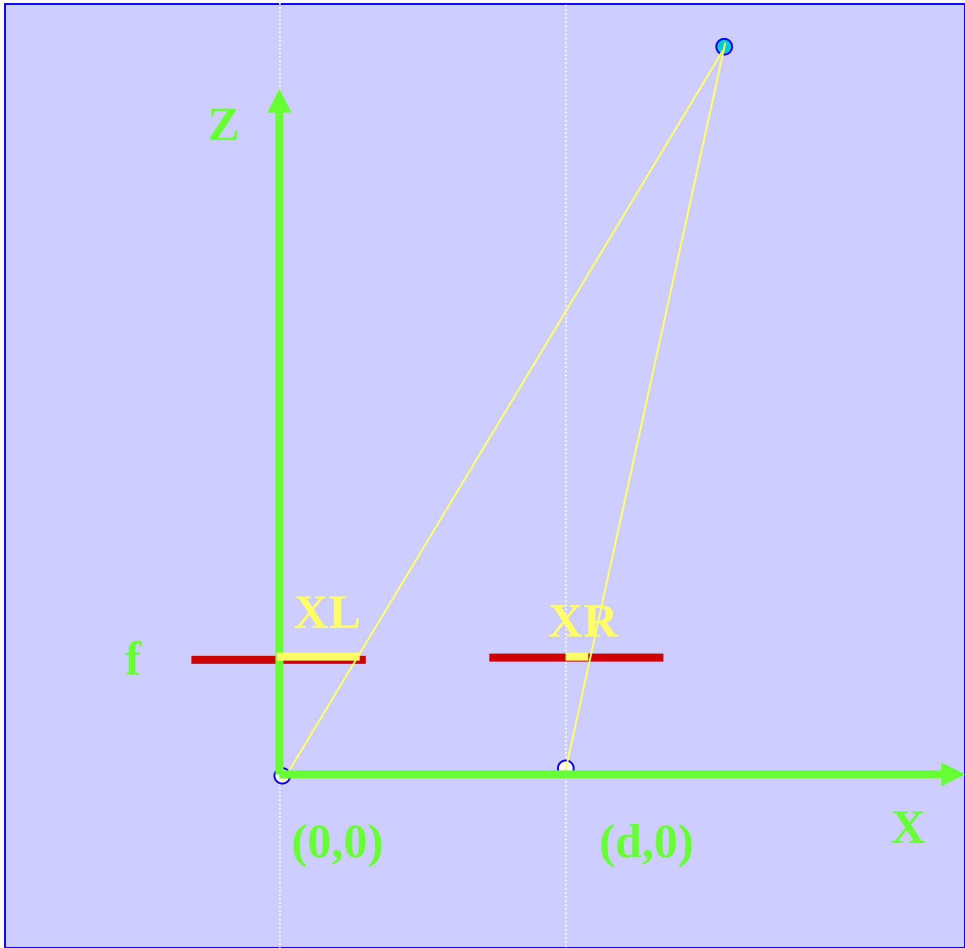
Z = (f/XL) X
Z= (f/XR) (X-d)

Solve for X:

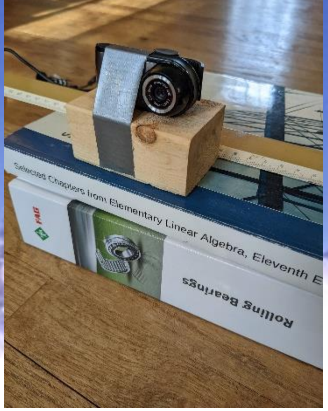
(f/XL) X = (f/XR) (X-d)
X = (XL d) / (XL - XR)

Solve for Z:

$$Z = \frac{d*f}{(XL - XR)}$$

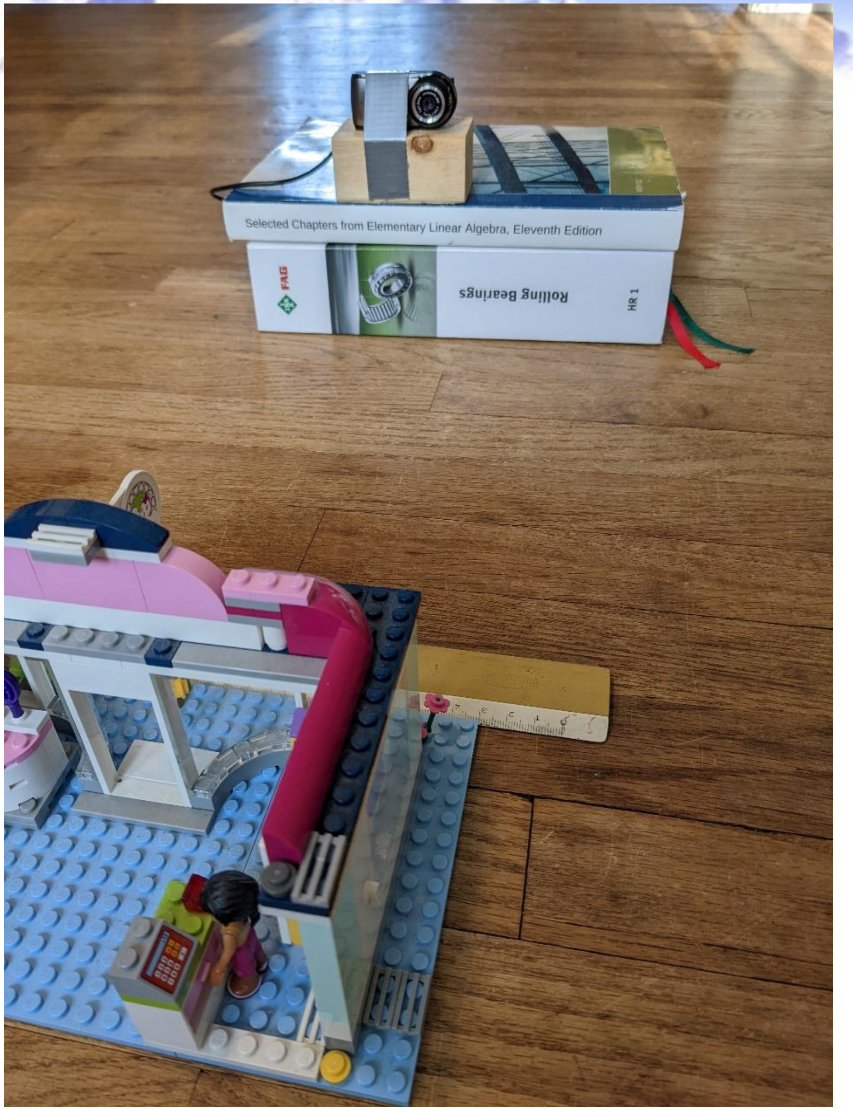


Lab: 3D “Stereo”



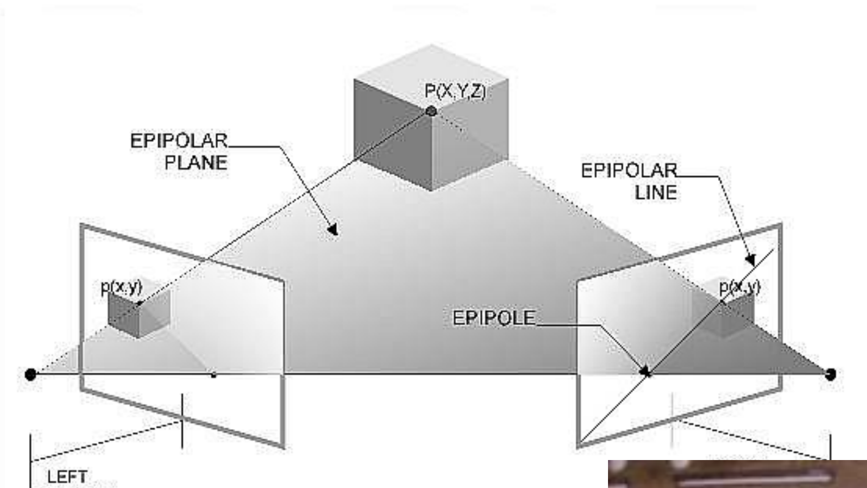
- To get several images slide camera along ruler
- For a non-square camera: tape it to a square object
- Track 10-100 salient points. (Can also “click” on them with ginput)
- Reconstruct 3D point cloud

Lab 3D "stereo"



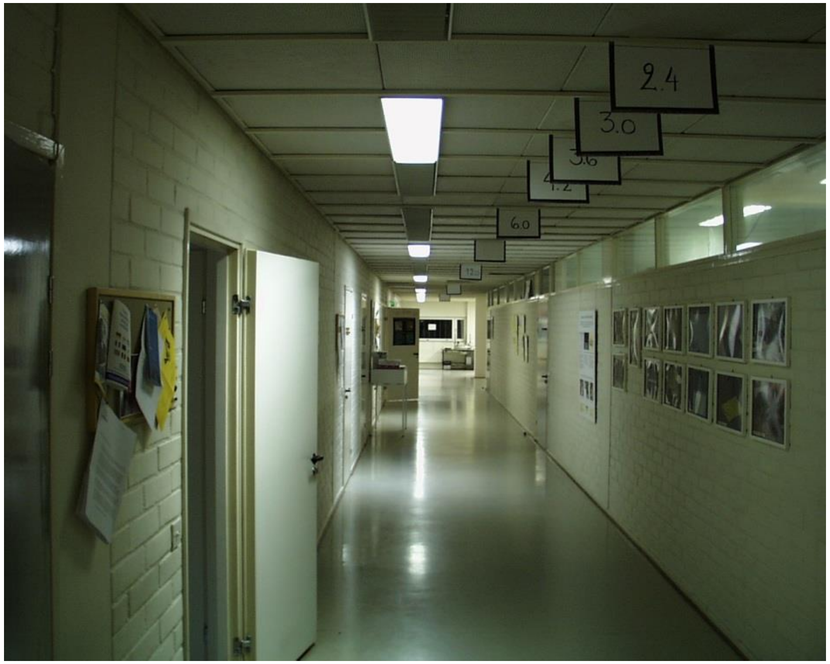
- Alternatively: Move object along ruler
- In both cases make sure motion is parallel to camera plane. (Only case these simplified 3D "Stereo" equations are valid for)

Epipolar constraint



Special case: **parallel cameras** – epipolar lines are parallel and aligned with rows

Stereo measurement example:



- Left image

Resolution = 1280 x 1024 pixels

f = 1360 pixels

- Right image

Baseline d = 1.2m

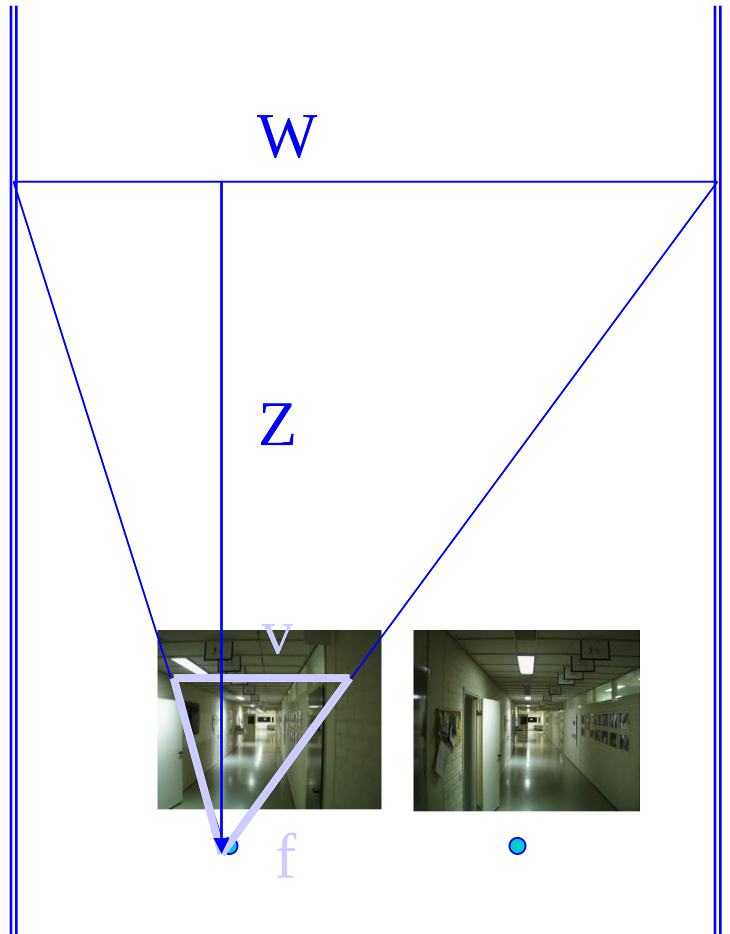
Q: How wide is the hallway

How wide is the hallway?
General strategy

- Similar triangles:

$$\frac{W}{Z} = \frac{v}{f}$$

- Need depth Z
- Then solve for W



How wide is the hallway? Steps in solution:

1. Compute focal length f in meters from pixels
2. Compute depth Z using stereo formula (aligned camera planes)

$$Z = \frac{d*f}{(XL - XR)}$$

3. Compute width:

$$W = Z \frac{v}{f}$$

Focal length:

Here screen projection is metric image plane.

0.224m is 1280 pixels

f = 1360 pixels

$$f = \frac{1360}{1280} * 0.224 = 0.238\text{m}$$

How wide…
Depth calculation

Disparity: XL – XR = 0.07m

(Note in the disparity calculation the choice of reference (here the edge) doesn't matter. But in the case of say X-coordinate calculation it should be w.r.t. the center of the image as in the stereo formula derivation

- Depth

$$Z = \frac{1.2 * 0.238}{0.07} = 4.1m$$

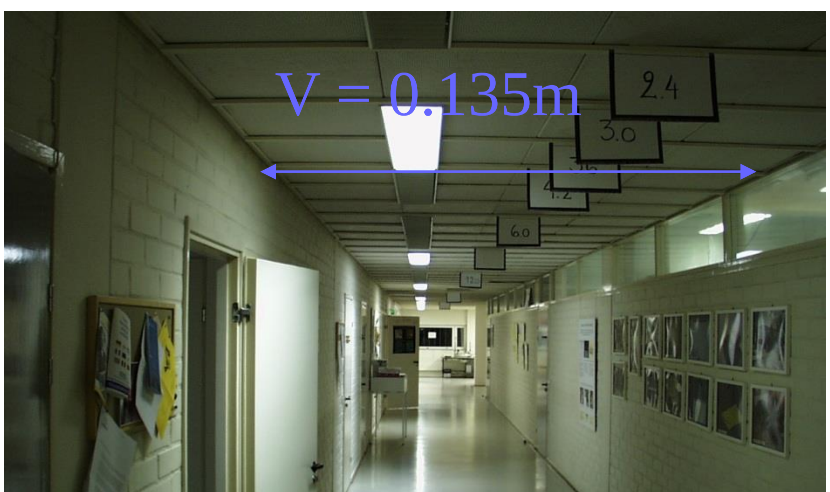
How wide…? Answer:

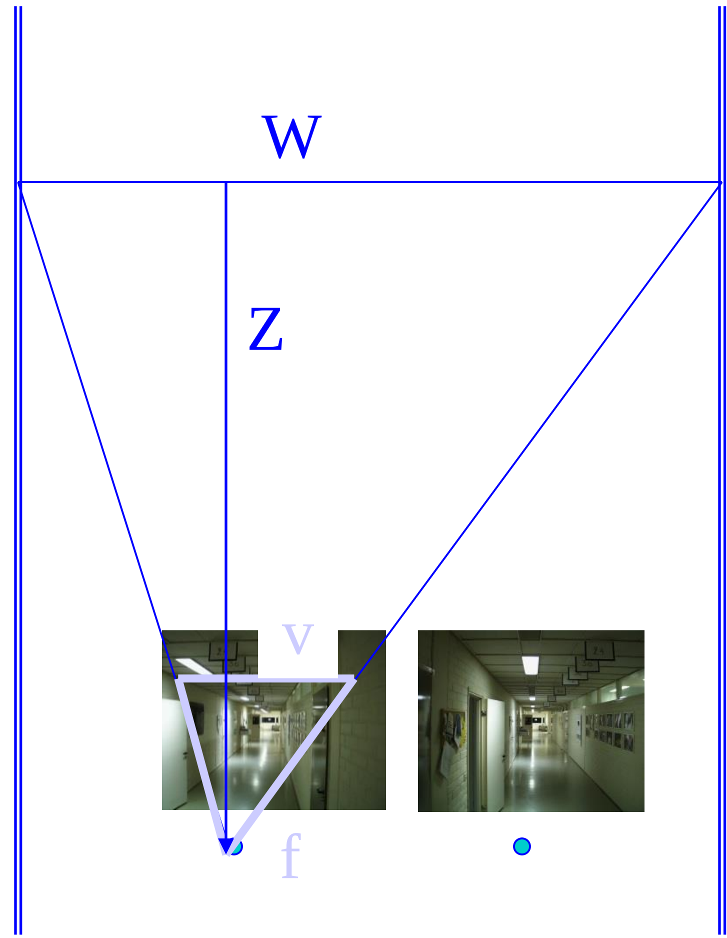
- Similar triangles:

$$W = Z\frac{v}{f}$$

- The width of the hallway is:

$$W = 4.1 * \frac{0.135}{0.238} = 2.3m$$





The camera matrix

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

- Homogenous coordinates for 3D
  - four coordinates for 3D point, 3 for a 2D

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

$$(U, V, W) \rightarrow (\frac{U}{W}, \frac{V}{W}) = (u, v)$$

The camera matrix

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

- Homogenous coordinates for 3D
  - Verify homogenous matrix form is the same:

$$\begin{pmatrix} X \\ Y \\ Z/f \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

$$(U, V, W) \rightarrow (\frac{U}{W}, \frac{V}{W}) = (u, v) \quad \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

The camera matrix

- Homogenous coordinates for 3D
  - equivalence relation (X,Y,Z,T) is the same as (k X, k Y, k Z,k T)

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \cong \begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Canonical form:
Left 3x3
identity matrix

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

The camera matrix

$$(U, V, W) \rightarrow \left(\frac{U}{W}, \frac{V}{W}\right) = (u, v)$$

- Homogenous coordinates for 3D
  - four coordinates for 3D point, 3 for a 2D

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

- When coordinate systems are not aligned
  - Projective: x image coordinates, X 3D coord, and P an arbitrary 3x4 matrix
  - x = PX
  - Euclidean
  - x = [R|T]X

The camera matrix

$$(U,V,W) \rightarrow (\frac{U}{W},\frac{V}{W}) = (u,v)$$

- Homogenous coordinates for 3D
  - four coordinates for 3D point
  - equivalence relation (X,Y,Z,T) is the same as (k X, k Y, k Z,k T)
- Turn previous expression into HC's
  - HC's for 3D point are (X,Y,Z,T)
  - HC's for point in image are (U,V,W)

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Camera parameters

- Issue
  - camera may not be at the origin, looking down the z-axis
    - extrinsic parameters
  - one unit in camera coordinates may not be the same as one unit in world coordinates
    - intrinsic parameters - focal length, principal point, aspect ratio, angle between axes, etc.

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{projection model} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Note: f moved from proj to intrinsics!

Intrinsic Parameters

Intrinsic Parameters describe the conversion from metric to pixel coordinates (and the reverse)

$$x_{mm} = - (x_{pix} - o_x)\, s_x$$
$$y_{mm} = - (y_{pix} - o_y)\, s_y$$

or

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix}_{pix} = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}_{mm} = M_{\text{int}}\, p$$

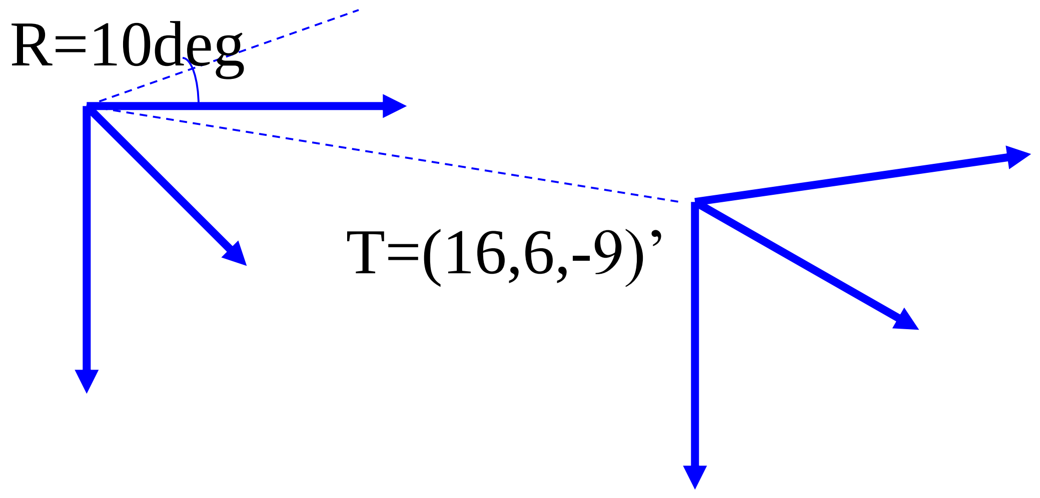
Note: Focal length is a property of the camera and can be incorporated as above

Example:
A real camera

- Laser range finder
- Camera

Relative location Camera-Laser

- Camera
- Laser



In homogeneous coordinates

- Rotation:

$$R = \begin{bmatrix} \cos -10 & 0 & \sin -10 \\ 0 & 1 & 0 \\ -\sin -10 & 0 & \cos -10 \end{bmatrix}$$

- Translation

$$\mathrm{T} = \begin{pmatrix} 1 & 0 & 0 & 16 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & -9 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Full projection model

- Camera internal parameters
- Camera projection

$$p_{\text{camera}} = \begin{pmatrix} 1278.6657 & 0 & 256 \\ 0 & 1659.5688 & 240 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0.985 & 0 & -0.174 & 0 \\ 0 & 1 & 0 & 0 \\ 0.174 & 0 & 0.985 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 16 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & -9 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.6612 \\ -10.55 \\ 108.0 \\ 1 \end{pmatrix} = \begin{pmatrix} 22262 \\ 16755 \\ 97.47 \end{pmatrix}$$

Extrinsic rot and translation

Full projection model

Coord from clicking in laser scan

$$\begin{pmatrix}22262\\16755\\97.47\end{pmatrix}=\begin{pmatrix}1279&0&256\\0&1660&240\\0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\end{pmatrix}\begin{pmatrix}0.985&0&-0.174&16\\0&1&0&6\\0.174&0&0.985&-9\\0&0&0&1\end{pmatrix}\begin{pmatrix}0.6612\\-10.55\\108.0\\1\end{pmatrix}$$

$$\begin{pmatrix}U\\V\\W\end{pmatrix}=\begin{pmatrix}\text{Transformation}\\\text{representing}\\\text{intrinsic parameters}\end{pmatrix}\begin{pmatrix}\text{Transformation}\\\text{representing}\\\text{projection model}\end{pmatrix}\begin{pmatrix}\text{Transformation}\\\text{representing}\\\text{extrinsic parameters}\end{pmatrix}\begin{pmatrix}X\\Y\\Z\\T\end{pmatrix}$$

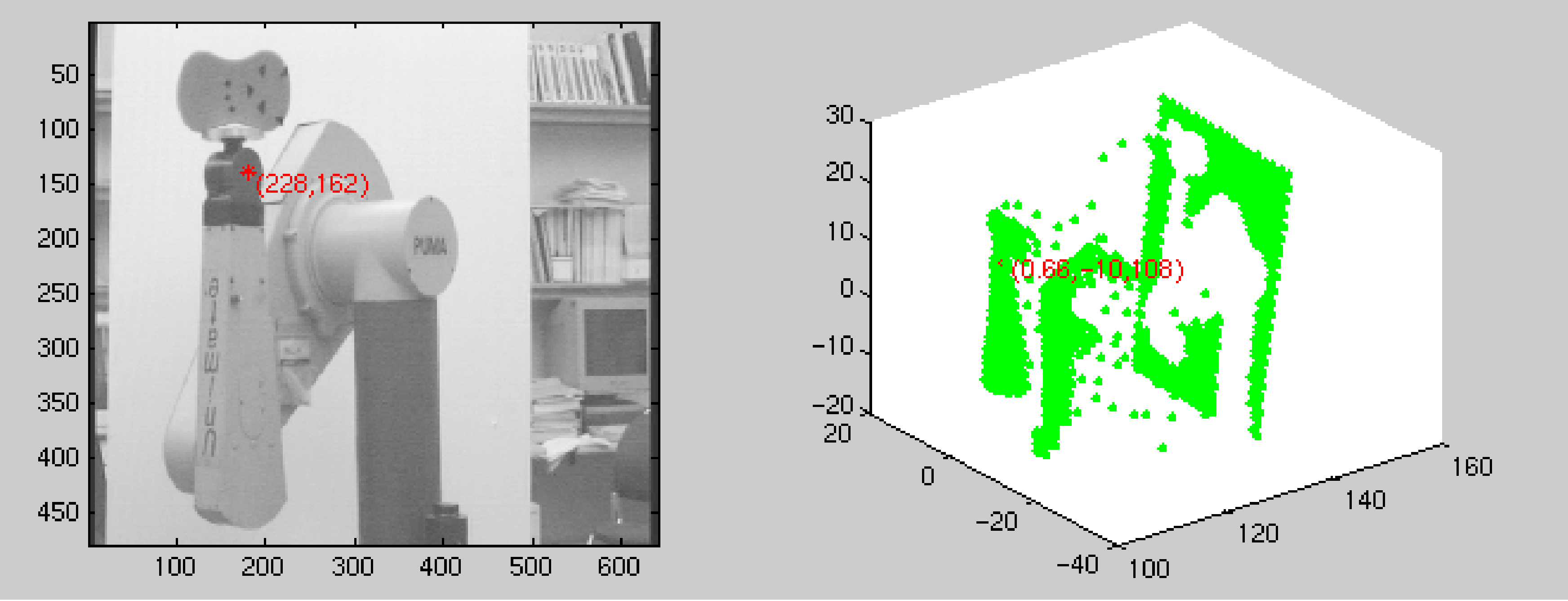
$$(U,V,W)\rightarrow\left(\frac{U}{W},\frac{V}{W}\right)=(u,v)$$

$$\begin{pmatrix}22262\\16755\\97.47\end{pmatrix}\longrightarrow\begin{pmatrix}\frac{22267}{97.47}\\\frac{16755}{97.47}\end{pmatrix}=\begin{pmatrix}228\\162\end{pmatrix}$$

Image pixel coordinates

Result

- Camera image
- Laser measured 3D structure



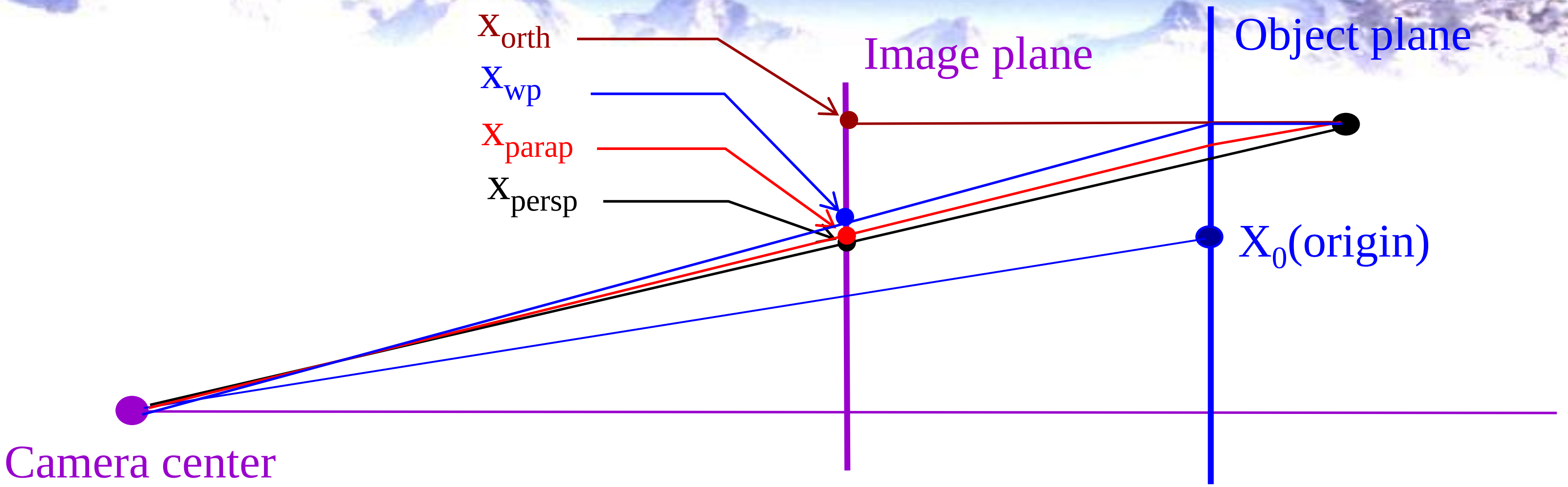
Camera parameters

- Issue
  - camera may not be at the origin, looking down the z-axis
    - extrinsic parameters
  - one unit in camera coordinates may not be the same as one unit in world coordinates
    - intrinsic parameters - focal length, principal point, aspect ratio, angle between axes, etc.

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{projection model} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Note: f moved from proj to intrinsics!

Hierarchy of different camera models



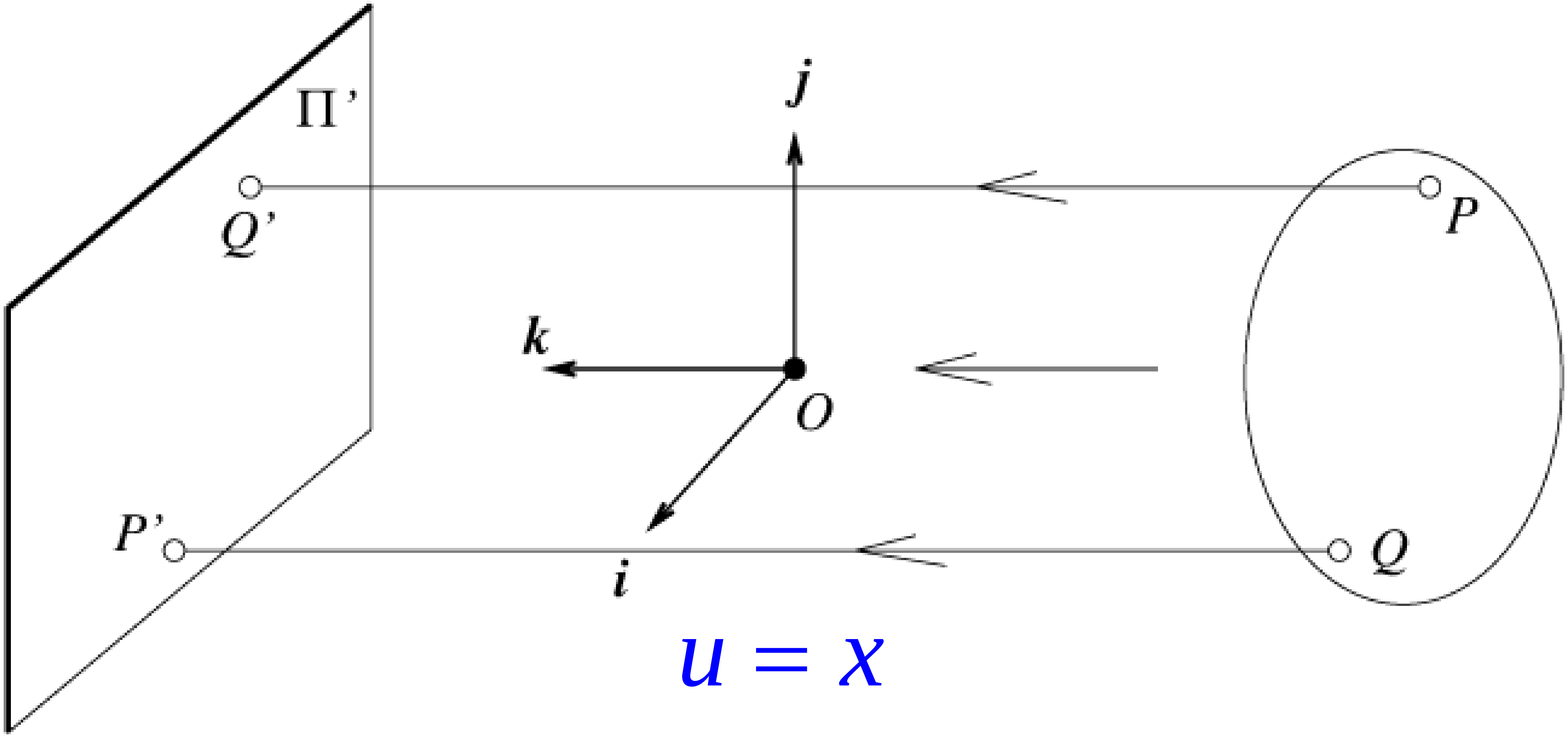
Perspective: non-linear

Weak perspective: linear approx

Orthographic: lin, no scaling

Para-perspective: lin

Orthographic projection



$$u = x$$

$$v = y$$

The fundamental model for orthographic projection

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Perspective and Orthographic Projection

perspective

Orthographic (parallel)

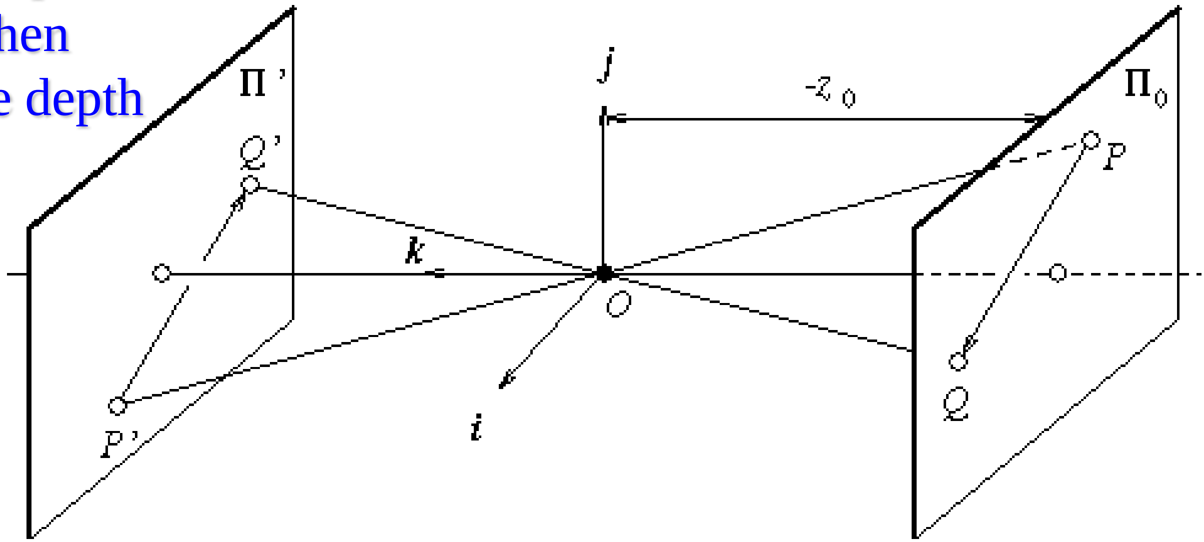
Weak perspective

- Issue
  - perspective effects, but not over the scale of individual objects
  - collect points into a group at about the same depth, then divide each point by the depth of its group
  - Adv: easy
  - Disadv: wrong

$$u = Tx$$

$$v = Ty$$

$$T = f / Z$$



The fundamental model for weak perspective projection

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & f/Z^* \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

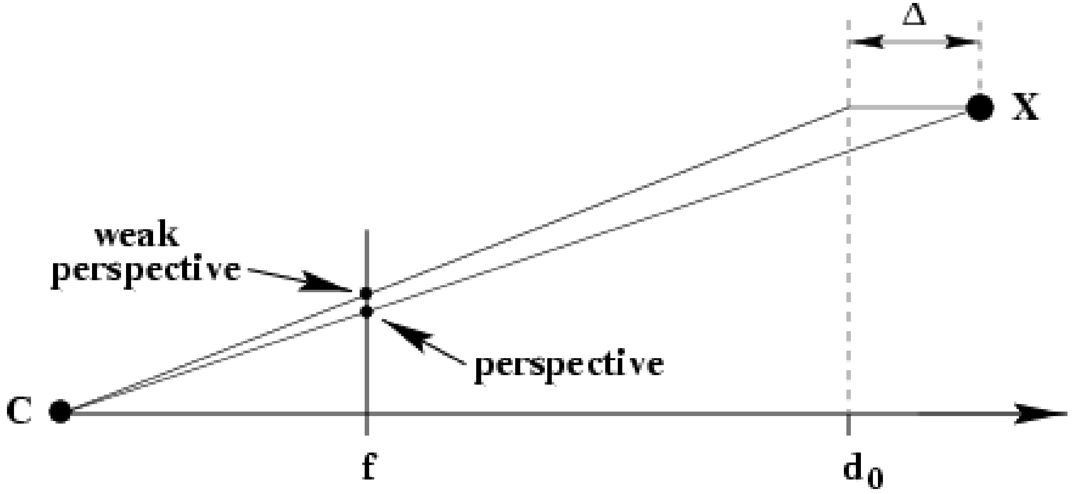
Note Z* is a fixed value, usually mean distance to scene

Weak perspective projection for an arbitrary camera pose R,t

Weak perspective projection

$$P_{\infty} = \begin{bmatrix} \alpha_x & & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathrm{r}^{1\mathrm{T}} & t_1 \\ \mathrm{r}^{2\mathrm{T}} & t_2 \\ 0 & 1/k \end{bmatrix}$$

(7dof)



Full Affine linear camera

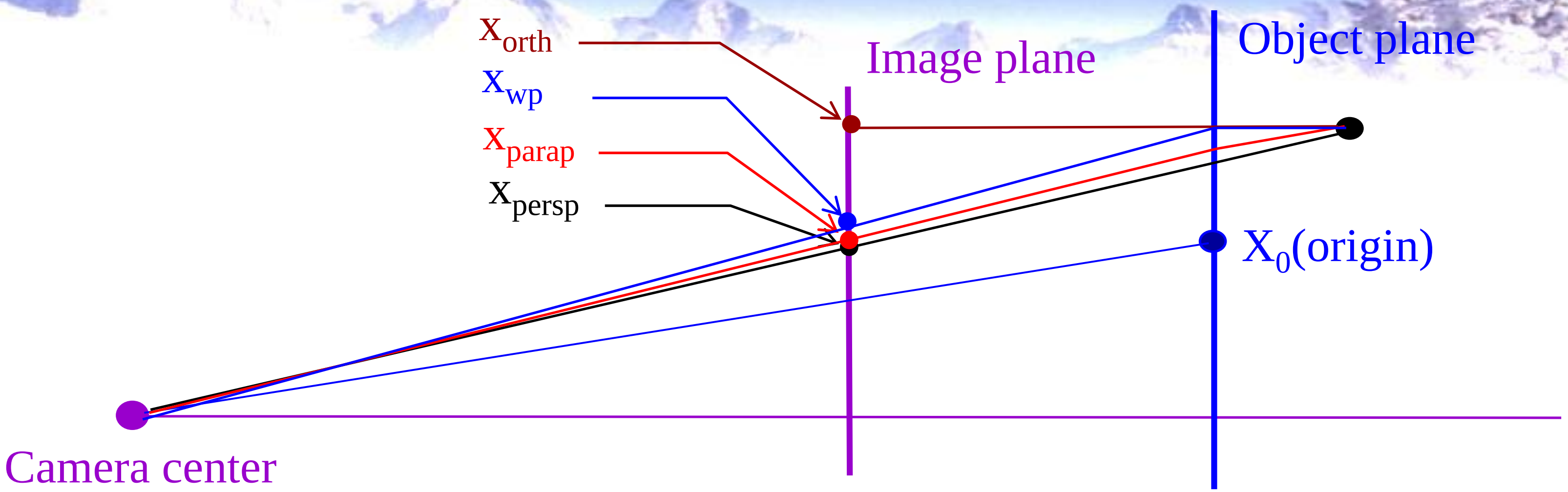
Affine camera (8dof)

$$P_A = \begin{bmatrix} \alpha_x & s & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} r^{1T} & t_1 \\ r^{2T} & t_2 \\ 0 & 1/k \end{bmatrix} \qquad P_A = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_A = \left[3\times 3 \text{ affine}\right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left[4\times 4 \text{ affine}\right]$$

1. Affine camera=camera with principal plane coinciding with Π_∞
2. Affine camera maps parallel lines to parallel lines
3. No center of projection, but direction of projection $P_A D=0$ (point on Π_∞)

Hierarchy of camera models



Perspective:

$$P_{persp} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{i} & t_x \\ \mathbf{j} & t_y \\ \mathbf{k} & t_z \end{bmatrix}$$

Weak perspective:

$$P_{wp} = \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{i} & t_x \\ \mathbf{j} & t_y \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Orthographic:

$$P_{orth} = \begin{bmatrix} \mathbf{i} & t_x \\ \mathbf{j} & t_y \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Para-perspective:

First order approximation of perspective

Camera Models

- Internal calibration:
- Weak calibration:
- Affine calibration:
- Stratification of stereo vision:
  - characterizes the reconstructive certainty of weakly, affinely, and internally calibrated stereo rigs

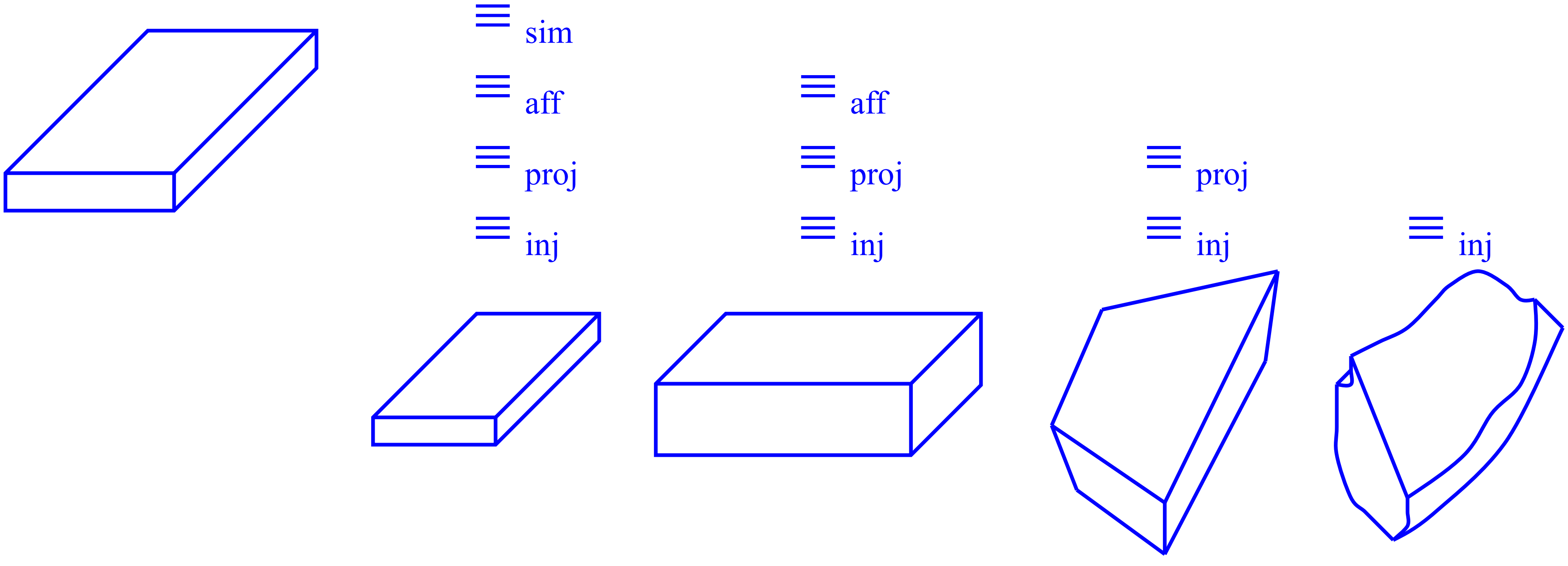
C_{sim} up to a similarity (scaled Euclidean transformation)

C_{aff} up to an affine transformation of task space

C_{proj} up to a projective transformation of task space

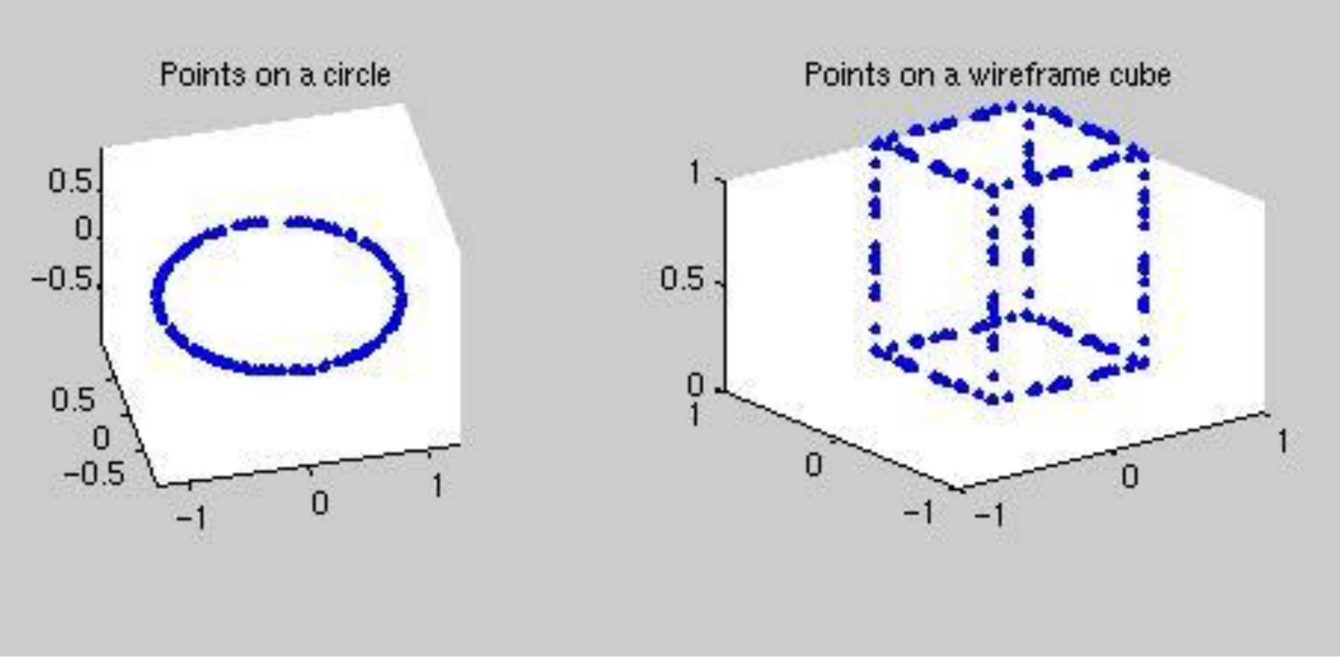
C_{inj} reconstruction up to a bijection of task space

Visual Invariance



Lab: Try these camera models

How do these pointclouds look when projected by different types of cameras



Try different Extrinsics
- Camera locations
- Camera rotations

Camera parameters

Compare to Matlab's built in 3D plotting

Perspective Camera Model Structure

Assume R and T express camera *in world coordinates, then*

$$^{c}p = \begin{pmatrix} & R' & & -R'T \\ 0 & 0 & 0 & 1 \end{pmatrix} {}^{w}p$$

Combining with a perspective model (and neglecting internal parameters) yields

$$^{c}u = M\,^{w}p = \begin{pmatrix} -R'_x & R'_x T \\ -R'_y & R'_y T \\ \dfrac{R_z}{f} & \dfrac{-R_z T}{f} \end{pmatrix} {}^{w}p$$

Note the M is defined only up to a scale factor at this point! If M is viewed as a 3x4 matrix defined up to scale, it is called the *projection matrix.*

Perspective Camera Model Structure

Assume R and T express camera *in world coordinates, then*

$$^{c}p = \begin{pmatrix} & R' & & -R'T \\ 0 & 0 & 0 & 1 \end{pmatrix} {}^{w}p$$

Combining with a weak perspective model (and neglecting internal parameters) yields

$$^{c}u = M\,{}^{w}p = \begin{pmatrix} -R'_x & R'_x T \\ -R'_y & R'_y T \\ 0 & \dfrac{R_z(\overline{P} - T)}{f} \end{pmatrix} {}^{w}p$$

Where $\overline{P}$ is the nominal distance to the viewed object

Other Models

- The *affine camera* is a generalization of weak perspective.
- The *projective camera* is a generalization of the perspective camera.
- Both have the advantage of being linear models on real and projective spaces, respectively.
- But in general will recover structure up to an affine or projective transform only. (ie distorted structure)

Camera Internal Calibration Recall: Intrinsic Parameters

Intrinsic Parameters describe the conversion from metric to pixel coordinates (and the reverse)

$x_{mm} = - (x_{pix} – o_x)\, s_x$
$y_{mm} = - (y_{pix} – o_y)\, s_y$

or

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix}_{pix} = \begin{pmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}_{mm} = M_{\text{int}} p$$

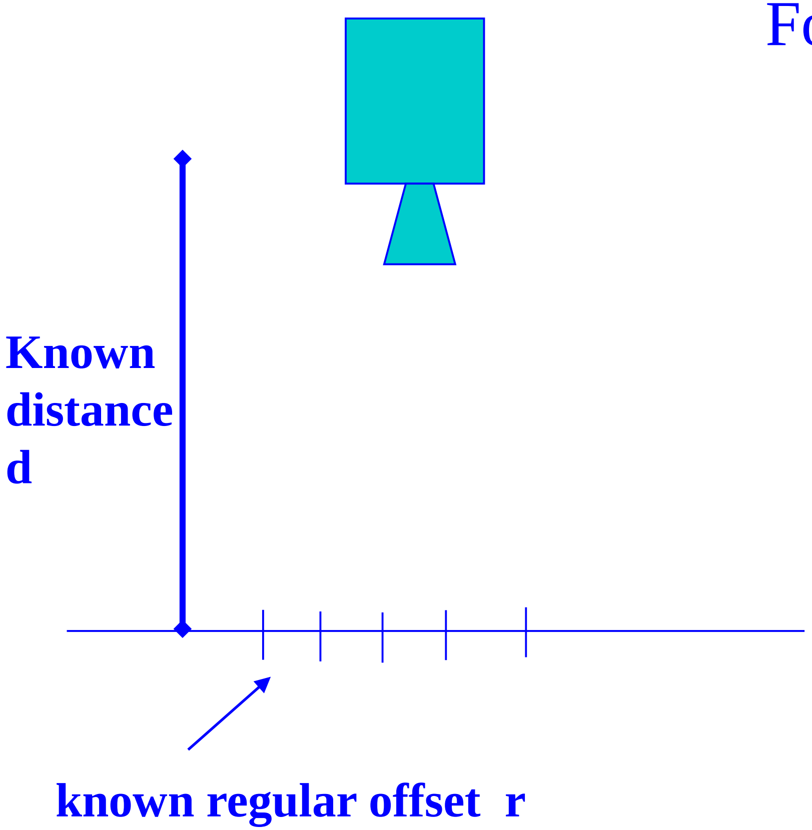
CAMERA INTERNAL CALIBRATION

Compute Sx

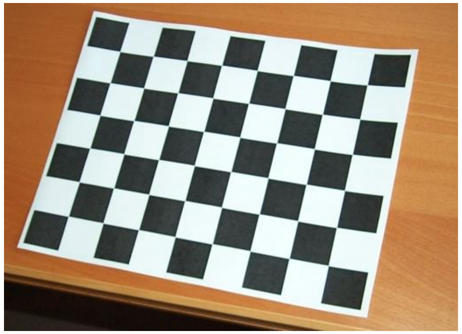
Focal length = 1/ Sx

$$\frac{rk_i}{d} = (x_i - o_x)s_x$$

$$\frac{r}{d} = (x_{i+1} - x_i)s_x$$



A simple way to get scale parameters; we can compute the optical center as the numerical center and therefore have the intrinsic parameters



Camera calibration

- Issues:
  - what are intrinsic parameters of the camera?
  - what is the camera matrix? (intrinsic+extrinsic)
- General strategy:
  - view calibration object
  - identify image points
  - obtain camera matrix by minimizing error
  - obtain intrinsic parameters from camera matrix
- Error minimization:
  - Linear least squares
    - easy problem numerically
    - solution can be rather bad
  - Minimize image distance
    - more difficult numerical problem
    - solution usually rather good, but can be hard to find
      - start with linear least squares
  - Numerical scaling is an issue