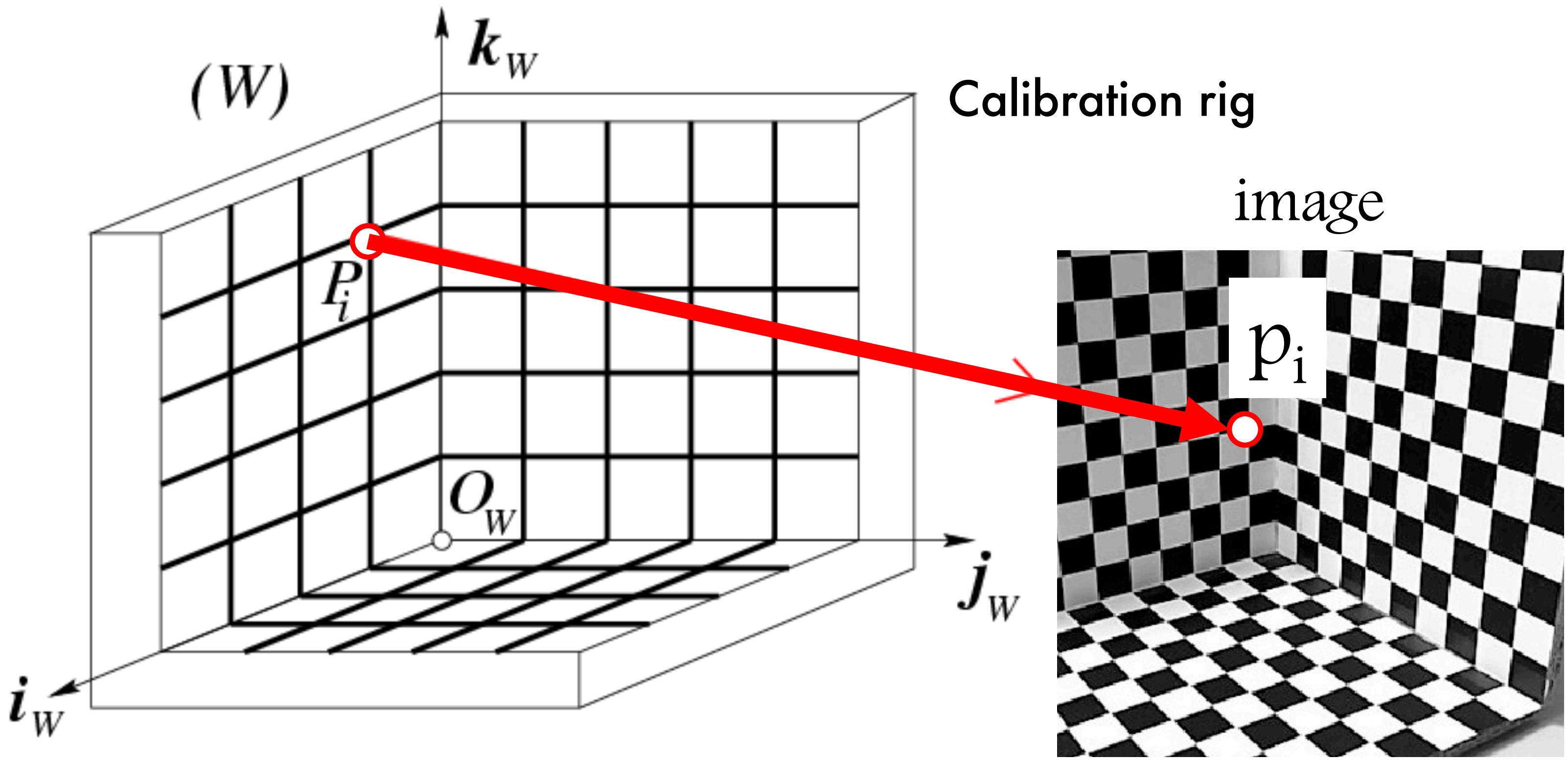
Lecture 4
Single View Metrology

- Review calibration and 2D transformations
- Vanishing points and lines
- Estimating geometry from a single image
- Extensions

Reading:
[HZ] Chapter 2 “Projective Geometry and Transformation in 2D”
[HZ] Chapter 3 “Projective Geometry and Transformation in 3D”
[HZ] Chapter 8 “More Single View Geometry”
[Hoeim & Savarese] Chapter 2

Silvio Savarese

Calibration Problem



$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M \; P_i$$

In pixels

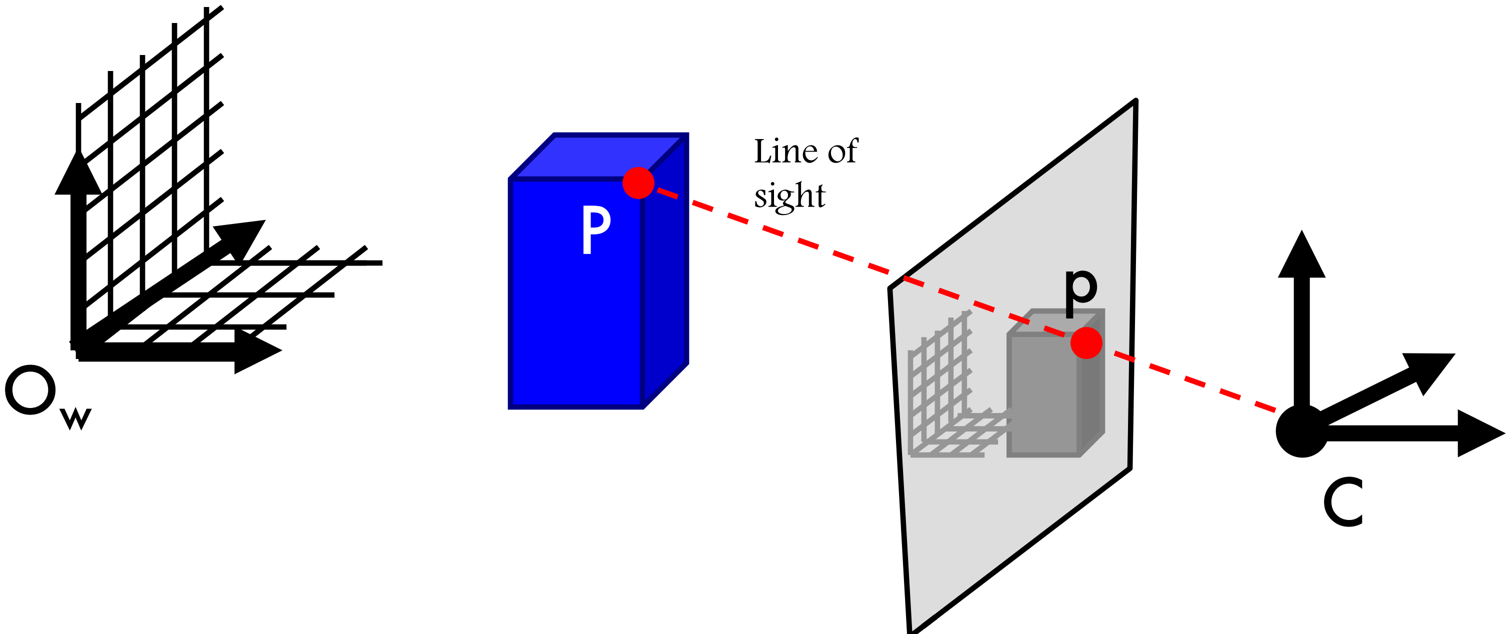
World ref. system

$$M = K\begin{bmatrix} R & T \end{bmatrix}$$

11 unknowns

Need at least 6 correspondences

Once the camera is calibrated...



$$M = K\begin{bmatrix} R & T \end{bmatrix}$$

-Internal parameters K are known
-R, T are known – but these can only relate C to the calibration rig

Can I estimate P from the measurement p from a single image?

No - in general ☹ (P can be anywhere along the line defined by C and p)

Recovering structure from a single view

http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/hut/hutme.wrl

Transformation in 2D

-Isometries

-Similarities

-Affinity

-Projective

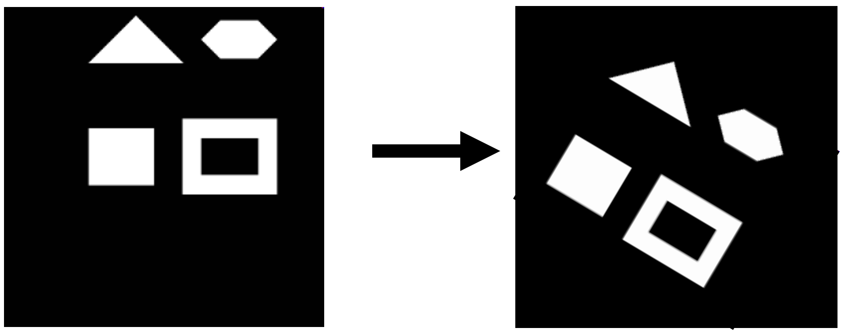
Transformation in 2D

Isometries:

[Euclideans]

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_e \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{[Eq. 4]}$$

- Preserve distance (areas)
- 3 DOF
- Regulate motion of rigid object



Class I: Isometries: preserve Euclidean distance

(*iso*=same, *metric*=measure)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \varepsilon\cos\theta & -\sin\theta & t_x \\ \varepsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \varepsilon = \pm 1$$

- orientation preserving: $\varepsilon = 1$ ➔ Euclidean transf. i.e. composition of translation and rotation ➔ forms a group
- orientation reversing: $\varepsilon = -1$ ➔ reflection ➔ does not form a group

$$\mathbf{x}' = \mathbf{H}_\varepsilon \mathbf{x} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \qquad \mathbf{R}^T\mathbf{R} = \mathbf{I}$$

R is 2x2 rotation matrix; (orthogonal, t is translation 2-vector, 0 is a null 2-vector

3DOF (1 rotation, 2 translation) ➔ trans. Computed from two point correspondences

special cases: pure rotation, pure translation

Invariants: length (distance between 2 pts), angle between 2 lines, area

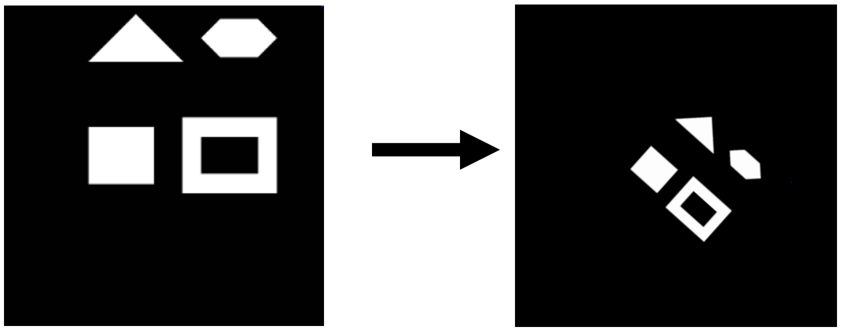
Transformation in 2D

Similarities:
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S\,R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$S = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$

[Eq. 5]

- Preserve
    - ratio of lengths
    - angles
- 4 DOF



Class II: Similarities: isometry composed with an isotropic scaling

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (\textit{isometry} + \textit{scale})$$

$$\mathbf{x}' = \mathbf{H}_S \mathbf{x} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \qquad \mathbf{R}^T\mathbf{R} = \mathbf{I}$$

4DOF (1 scale, 1 rotation, 2 translation) ➔ 2 point correspondences

Scalar s: isotropic scaling

also known as *equi-form* (shape preserving)

metric structure = structure up to similarity (in literature)

Invariants: ratios of length, angle, ratios of areas, parallel lines

Metric Structure means structure is defined up to a similarity

Transformation in 2D

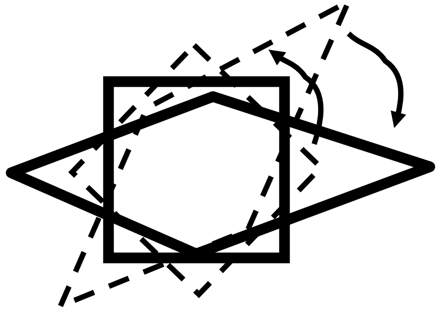
Affinities:

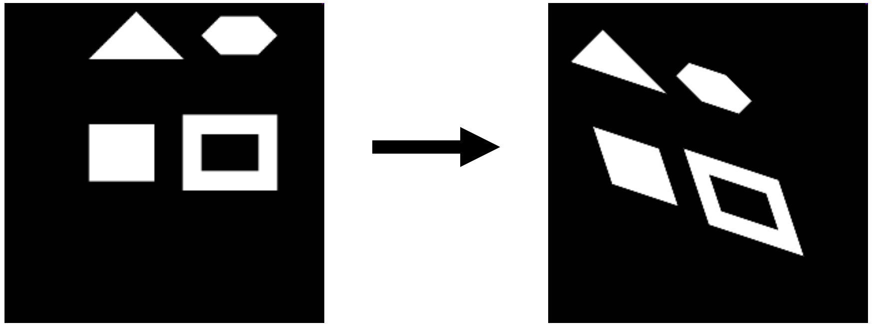
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

[Eq. 6]

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta) \cdot R(-\phi) \cdot D \cdot R(\phi) \qquad D = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

[Eq. 7]





Transformation in 2D

Affinities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

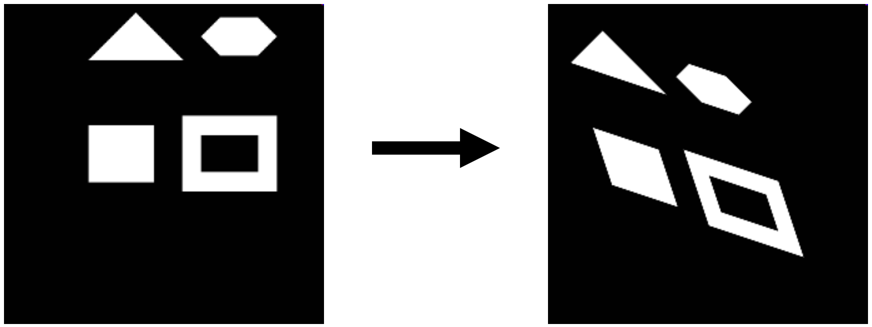
[Eq. 6]

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta) \cdot R(-\phi) \cdot D \cdot R(\phi) \qquad D = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

[Eq. 7]

-Preserve:
- Parallel lines
- Ratio of areas
- Ratio of lengths on collinear lines
- others…

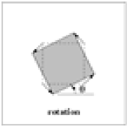
- 6 DOF



Class III: Affine transformations: non singular linear transformation followed by a translation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathbf{x}' = \mathbf{H}_A \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \mathbf{x}$$



Rotation by theta

R(-phi) D R(phi) scaling directions in the deformation are orthogonal

Can show: $\mathbf{A} = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{D}\mathbf{R}(\phi)$ $\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$

- Rotation by phi, scale by D, rotation by – phi, rotation by theta
- 6DOF (2 scale, 2 rotation, 2 translation) ➔ 3 point correspondences
  non-isotropic scaling!

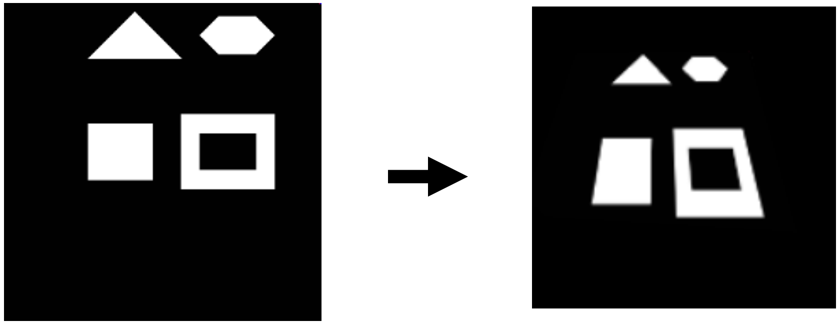
Invariants: parallel lines, ratios of parallel lengths, ratios of areas

Affinity is orientation preserving if det (A) is positive ➔ depends on the sign of the scaling

Transformation in 2D

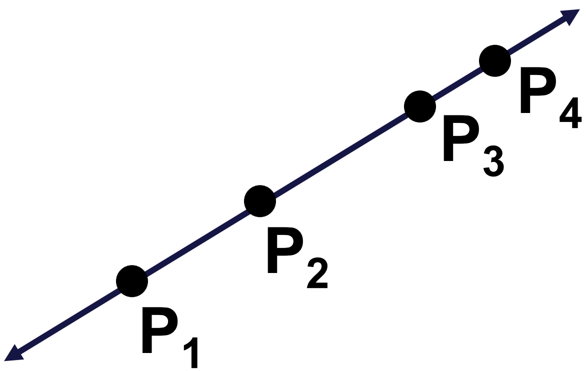
Projective:
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ v & b \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_p \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
[Eq. 8]

- 8 DOF
- Preserve:
  - collinearity
  - cross ratio of 4 collinear points
  - and a few others…



The cross ratio

The cross-ratio of 4 collinear points is defined as



[Eq. 9]

$$\frac{\left\|\mathbf{P}_3 - \mathbf{P}_1\right\| \left\|\mathbf{P}_4 - \mathbf{P}_2\right\|}{\left\|\mathbf{P}_3 - \mathbf{P}_2\right\| \left\|\mathbf{P}_4 - \mathbf{P}_1\right\|}$$

$$\mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Class IV: Projective transformations: general non singular linear transformation of homogenous coordinates

$$\mathbf{x}' = \mathbf{H}_P\,\mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^{\mathsf{T}} & v \end{bmatrix} \mathbf{x} \qquad \mathbf{v} = (v_1, v_2)^{\mathsf{T}}$$

Hp has nine elements; only their ratio significant ➔ 8 Dof ➔ 4 correspondences
Not always possible to scale the matrix to make v unity: might be zero

Action non-homogeneous over the plane

Invariants: cross-ratio of four points on a line
(ratio of ratio of length)

Projective transformations

Definition:

A *projectivity* is an invertible mapping h from P^2 to itself such that three points x_1, x_2, x_3 lie on the same line if and only if $h(x_1), h(x_2), h(x_3)$ do. (i.e. maps lines to lines in P^{2})

Theorem:

A mapping $h: \mathbf{P}^2 \rightarrow \mathbf{P}^2$ is a projectivity if and only if there exist a non-singular 3x3 matrix $\mathbf{H}$ such that for any point in $\mathbf{P}^2$ reprented by a vector x it is true that $h(\mathrm{x})=\mathbf{H}\mathrm{x}$

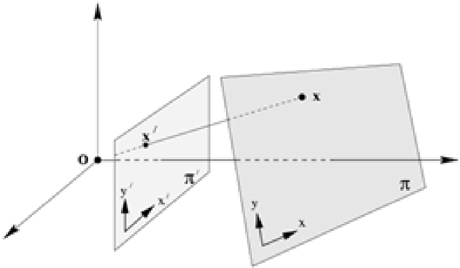
Definition: Projective transformation: linear transformation on homogeneous 3 vectors represented by a non singular matrix H

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{or} \quad \mathbf{x}' = \mathbf{H}\,\mathbf{x}$$

8DOF

- projectivity=collineation=projective transformation=homography
- Projectivity form a group: inverse of projectivity is also a projectivity; so is a composition of two projectivities.

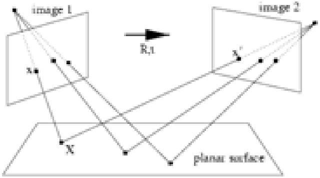
Projection along rays through a common point, (center of projection) defines a mapping from one plane to another



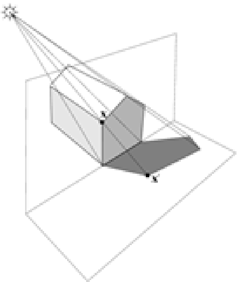
•Central projection maps points on one plane to points on another plane
•Projection also maps lines to lines : consider a plane through projection center that intersects the two planes ➔ lines mapped onto lines ➔ Central projection is a projectivity ➔

central projection may be expressed by x'=Hx
(application of theorem)

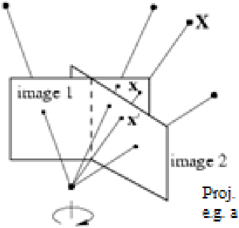
More examples



Projective transformation between two images
Induced by a world plane ➔ concatenation of two
Pojective transformations is also a proj. trans.

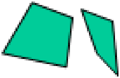
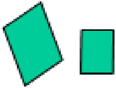
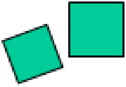
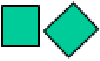


Proj. trans. Between the image of a plane
(end of the building) and the image of its
Shadow onto another plane (ground plane)

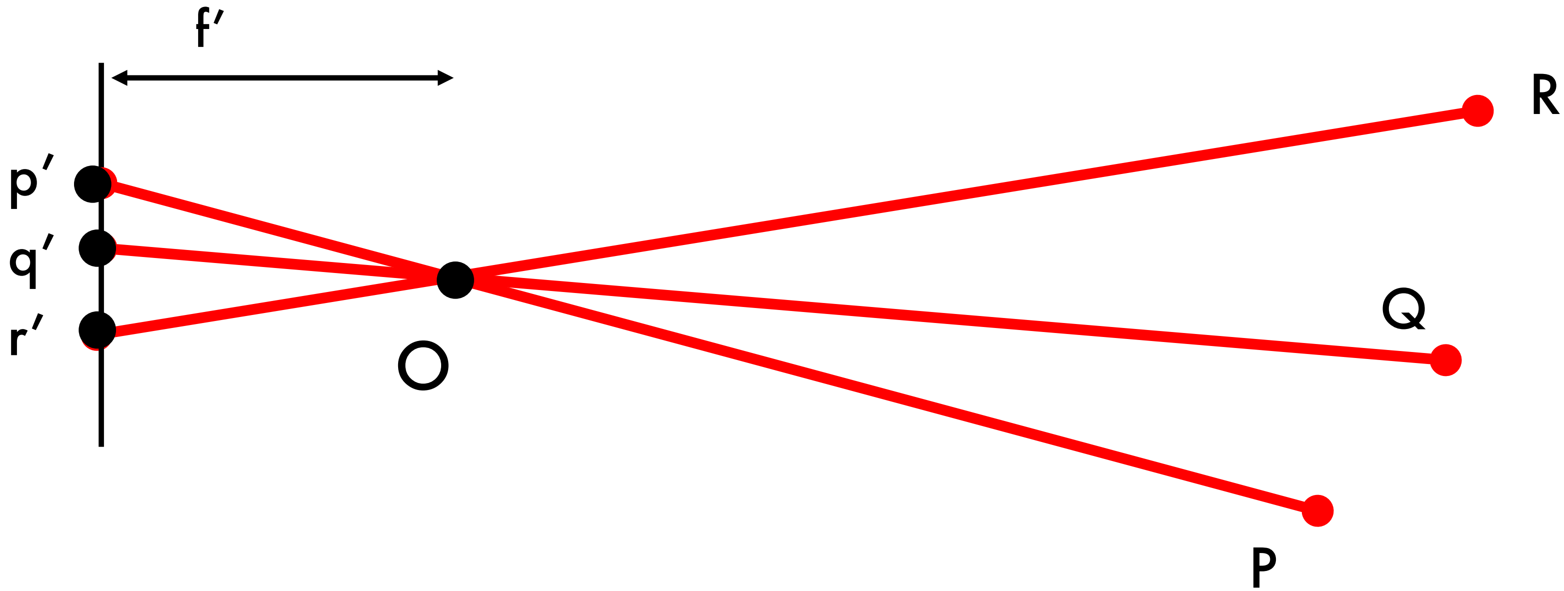


Proj. trans. Between two images with the same camera center
e.g. a camera rotating about its center

Overview transformations

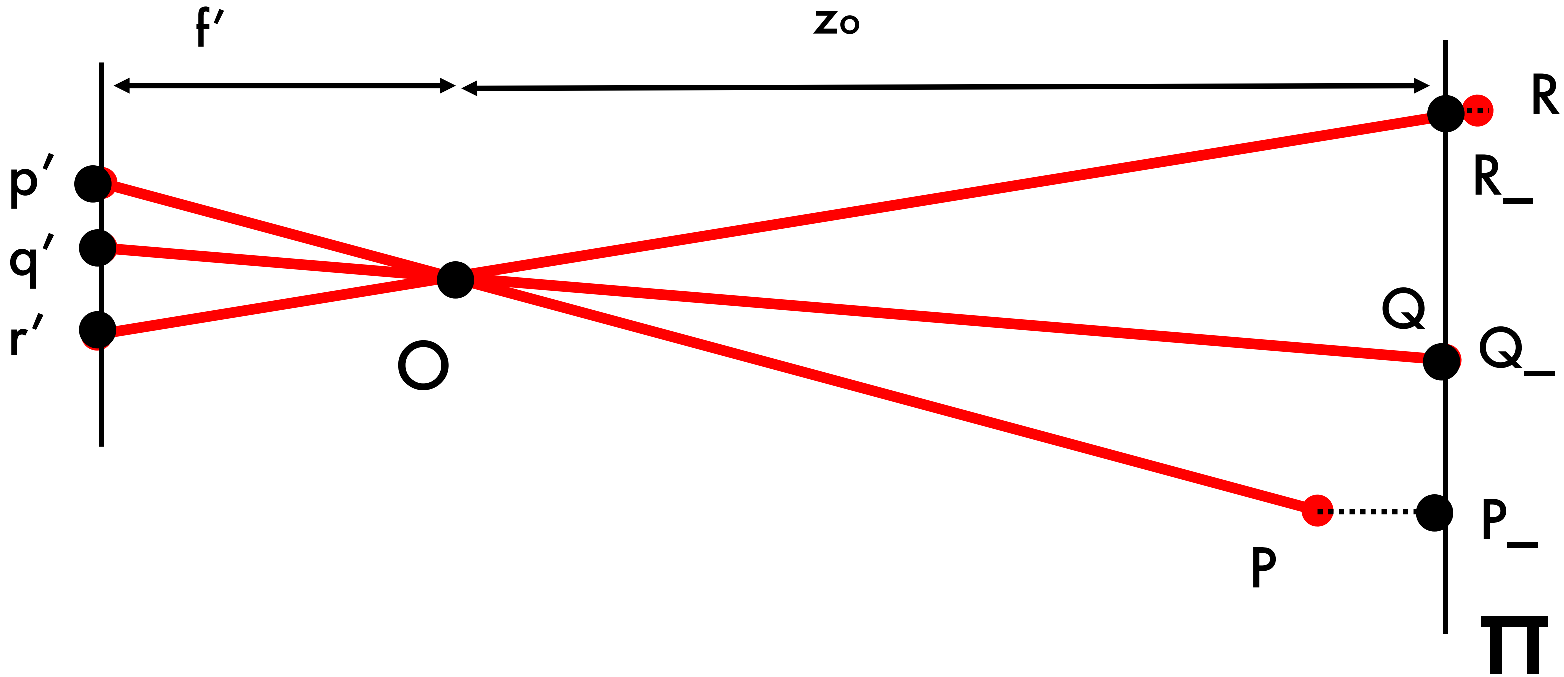
Transformation	Matrix	Example	Invariant properties
Projective 8dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio
Affine 6dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallellism, ratio of areas, ratio of lengths on parallel lines (e.g midpoints), linear combinations of vectors (centroids). **The line at infinity l_∞**
Similarity 4dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratios of lengths, angles. **The circular points I,J**
Euclidean 3dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		lengths, areas.

Projective camera

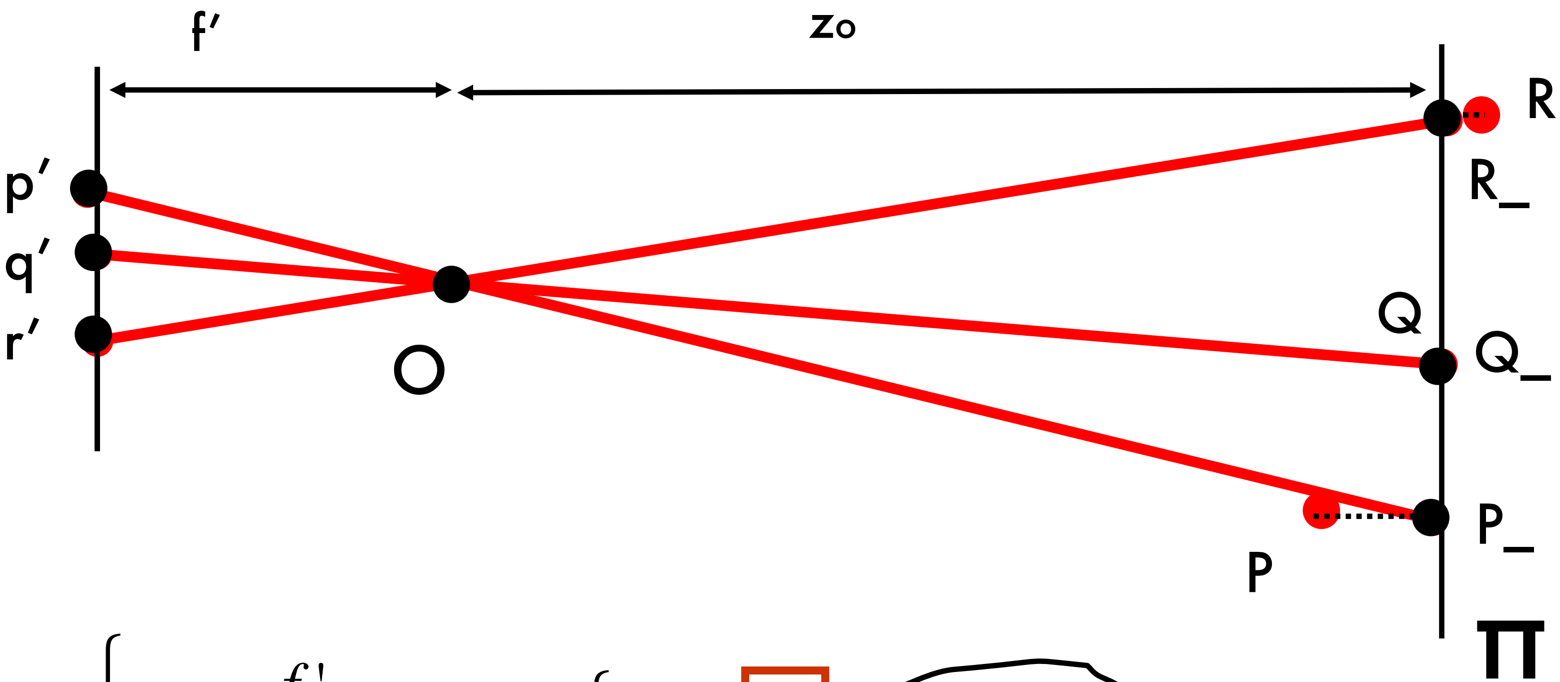


Weak perspective projection

When the relative scene depth is small compared to its distance from the camera



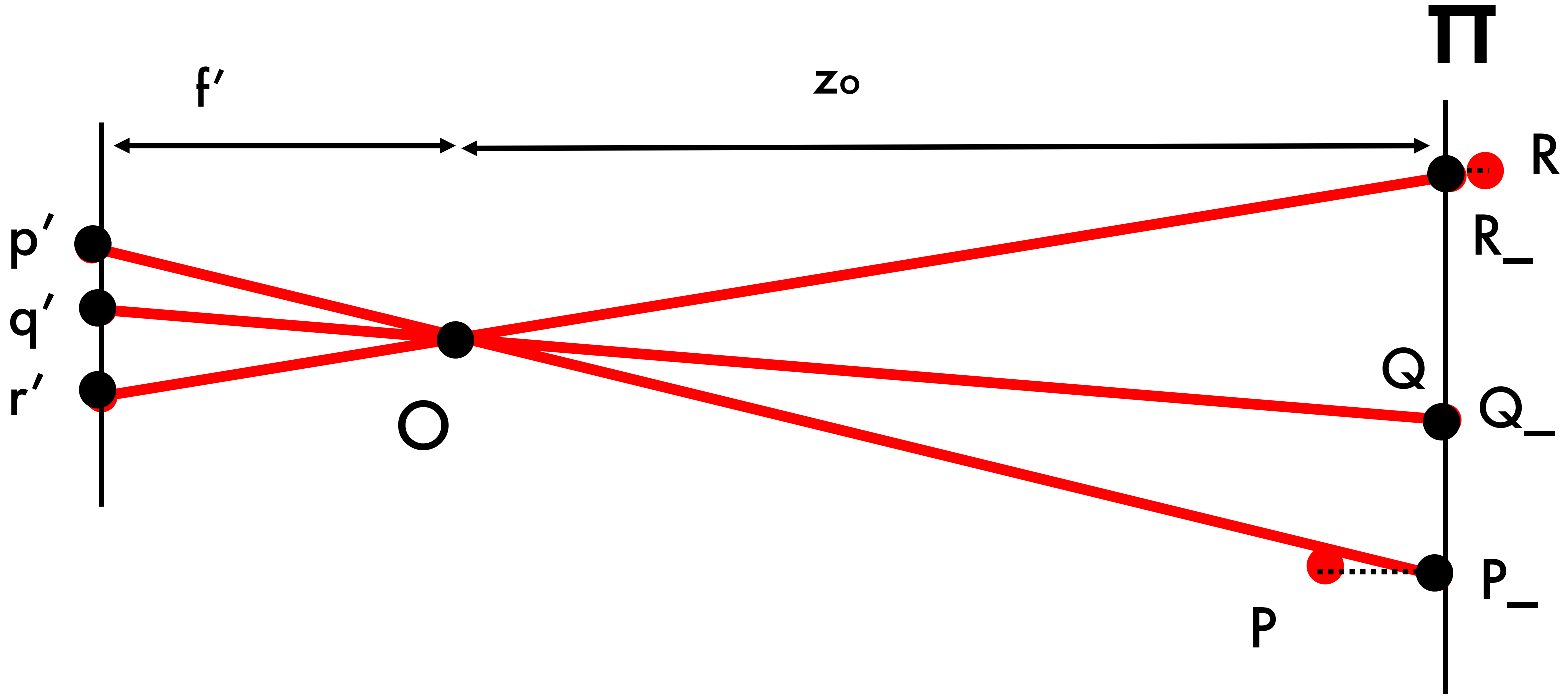
Weak perspective projection



$$\begin{cases} x' = \dfrac{f'}{z} x \\ y' = \dfrac{f'}{z} y \end{cases} \rightarrow \begin{cases} x' = \boxed{\dfrac{f'}{z_0}} x \\ y' = \dfrac{f'}{z_0} y \end{cases}$$

Magnification m

Weak perspective projection



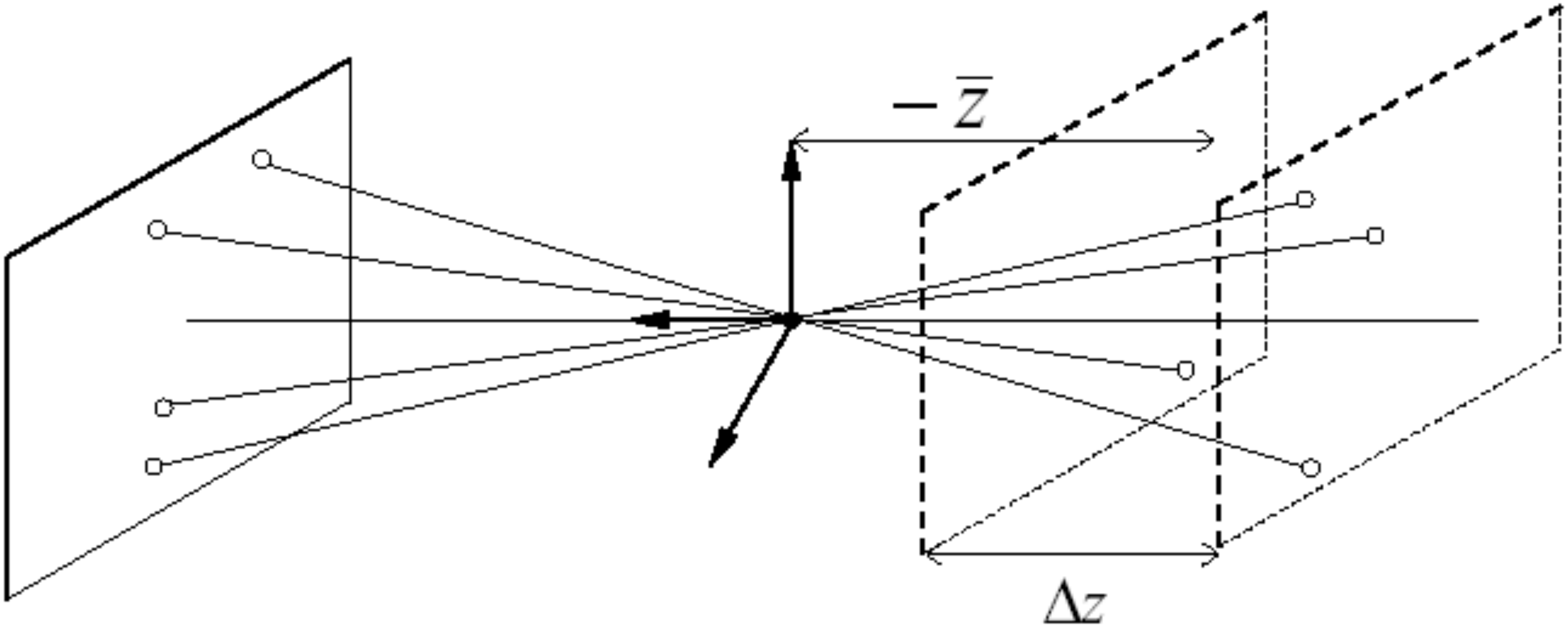
Projective (perspective)

$$M = K\begin{bmatrix} R & T \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{v} & 1 \end{bmatrix} \rightarrow$$

Weak perspective

$$M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Special Case: Weak Perspective (Affine Projection)



If $\Delta z << -\bar{z}$: $\begin{array}{l} x' \approx -mx \\ y' \approx -my \end{array}$ $\quad m = -\frac{f}{\bar{z}}$

Justified if scene depth is small relative to average distance from camera

$$P' = M\,P_w = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_W = \begin{bmatrix} \mathbf{m}_1 P_W \\ \mathbf{m}_2 P_W \\ \mathbf{m}_3 P_W \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{v} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

$$\xrightarrow{\mathbf{E}} \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w}\right)$$

Perspective: projective transformation

$$P' = M\,P_w = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_W = \begin{bmatrix} \mathbf{m}_1 P_W \\ \mathbf{m}_2 P_W \\ 1 \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = \begin{bmatrix} & \mathbf{m}_1 & & \\ & \mathbf{m}_2 & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

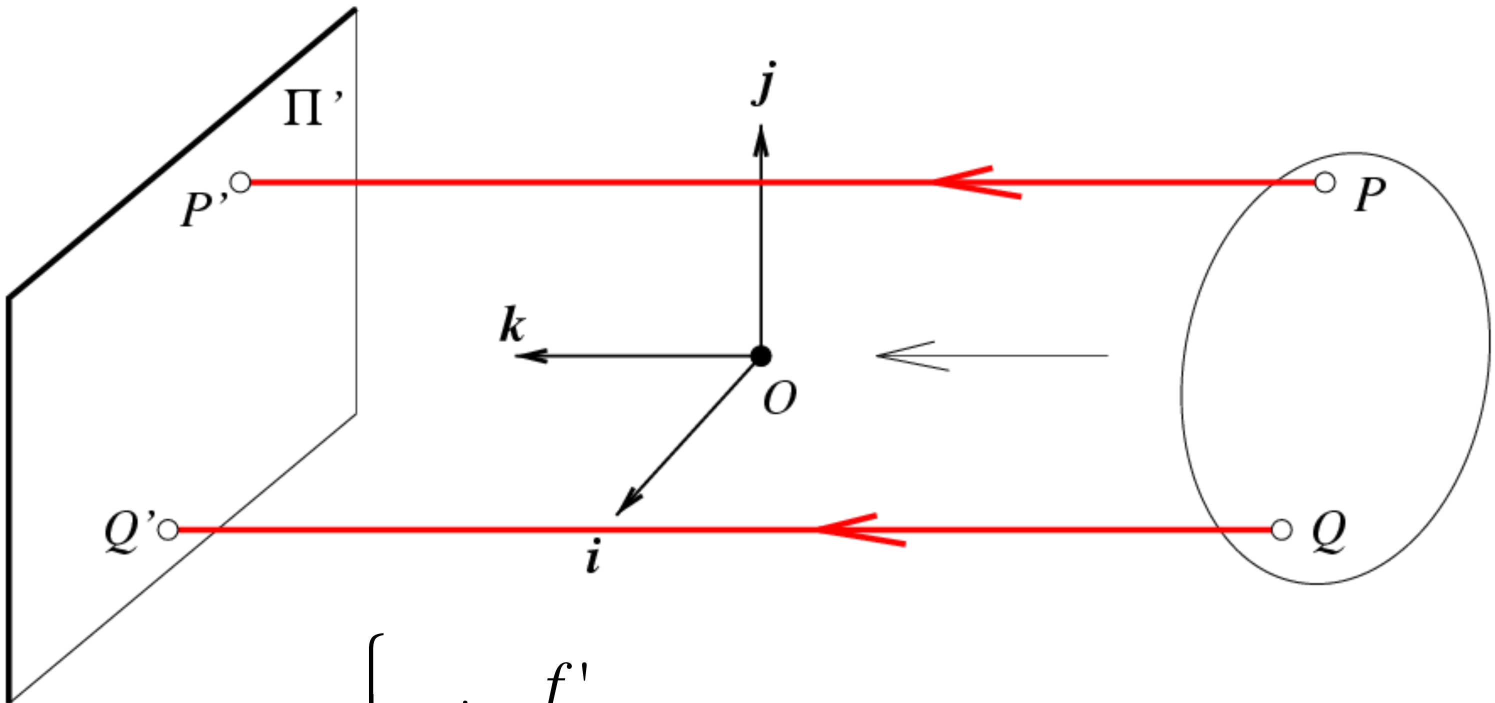
$$\xrightarrow{\mathbf{E}} (\mathbf{m}_1 P_w, \mathbf{m}_2 P_w)$$

↑ ↑ magnification

Weak Prospective: Affine Transformatoin

Orthographic (affine) projection

Distance from center of projection to image plane is infinite



$$\begin{cases} x' = \frac{f'}{z} x \\ y' = \frac{f'}{z} y \end{cases} \rightarrow \begin{cases} x' = x \\ y' = y \end{cases}$$

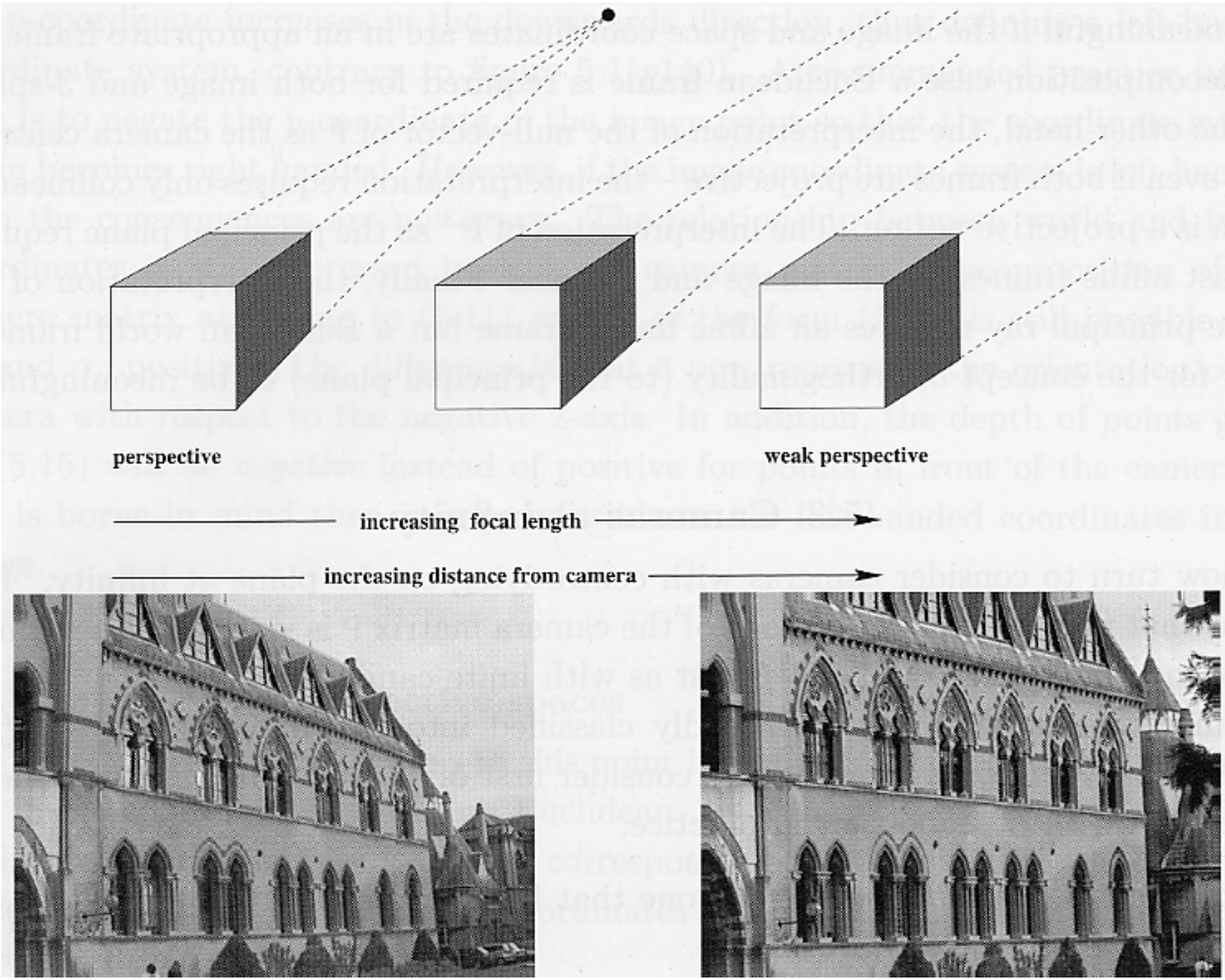
Affine Transformation

Pros and Cons of These Models

- Weak perspective results in much simpler math.
  - Accurate when object is small and distant.
  - Most useful for recognition.

- Pinhole perspective is much more accurate for modeling the 3D-to-2D mapping.
  - Used in structure from motion or SLAM.

Strong perspective:
Angles are not preserved
The projections of parallel lines intersect at one point



From Zisserman & Hartley

Strong perspective:
Angles are not preserved
The projections of parallel
lines intersect at one point

Weak perspective:
Angles are better preserved
The projections of parallel
lines are (almost) parallel

A hierarchy of transformations

- Group of invertible nxn matrices with real elements ➔ general linear group on n dimensions GL(n);
- Projective linear group: matrices related by a scalar multiplier PL(n); three subgroups:
  - Affine group (last row (0,0,1))
  - Euclidean group (upper left 2x2 orthogonal)
  - Oriented Euclidean group (upper left 2x2 det 1)

- Alternative, characterize transformation in terms of elements or quantities that are preserved or *invariant*
  - e.g. Euclidean transformations (rotation and translation) leave distances unchanged



Similarity Affine projective

•Similarity: circle imaged as circle; square as square; parallel or perpendicular lines have same relative orientation

•Affine: circle becomes ellipse; orthogonal world lines not imaged as orthogonoal; But, parallel lines in the square remain parallel

•Projective: parallel world lines imaged as converging lines; tiles closer to camera larger image than those further away.

Lecture 4

Single View Metrology

- Review calibration and 2D transformations
- Vanishing points and lines
- Estimating geometry from a single image
- Extensions

Reading:
[HZ] Chapter 2 “Projective Geometry and Transformation in 2D”
[HZ] Chapter 3 “Projective Geometry and Transformation in 3D”
[HZ] Chapter 8 “More Single View Geometry”
[Hoeim & Savarese] Chapter 2

Euclidean Geometry

1. Euclidean Geometry

- Euclidean geometry describes the world well.
- It allows to measure lengths and angles.
- Length, angles, parallelism, orthogonality, and all other properties that are related via a linear/Euclidean transform are preserved.
- Euclidean coordinates of a point in a plane are given by a 2-tuple ~ $[u, v]^T$.

- Ex: Consider the transformation that rotates 2 points, P_1, P_2, in a plane counter-clockwise θ° with respect to the origin as shown in Fig. 1. The transformation can be represented by the linear equations,

$$P_1' = R.P_1 = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} . P_1 \quad 1.1$$

$$P_2' = R.P_2 = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} . P_2 \quad 1.2$$

Since the transformation is Euclidean, the length between the two points, and the angle subtended at the origin, before and after the transformation remains the same.

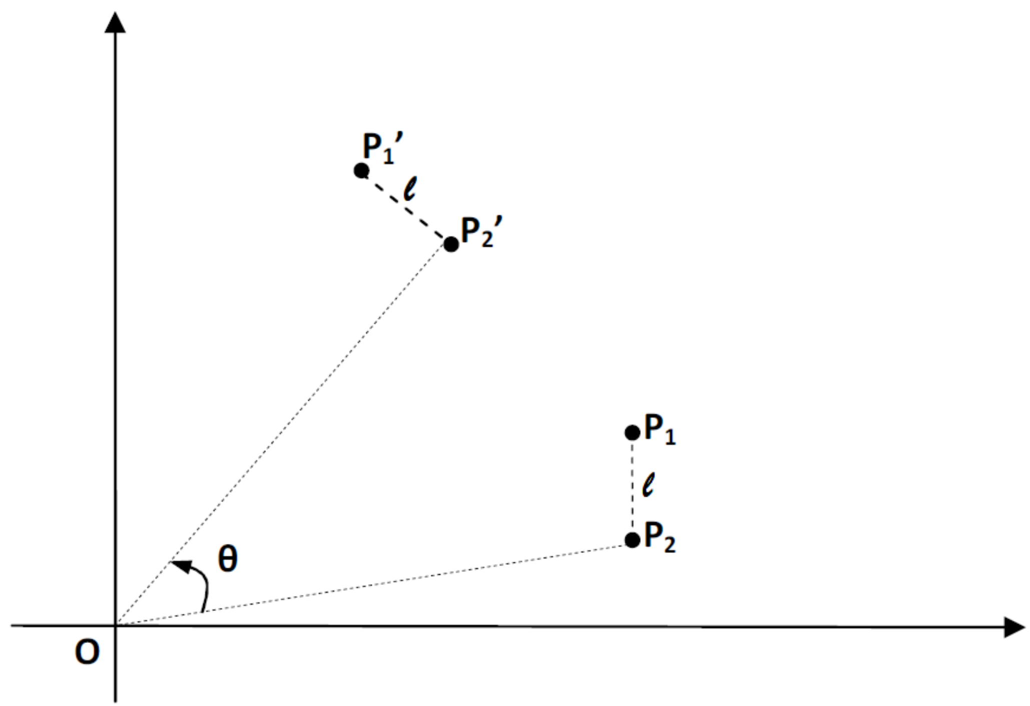


Fig. 1. – Rotating a point about the origin is a Euclidean transformation.

Q – Why do we need Projective geometry?

A – Because 3D objects are projected on to a 2D plane on capturing an image.

Projective Geometry

- Describes projection to lower dimensions well. For instance, parallel lines in 3D space are no longer parallel in a 2D image projection, and appear to meet. Such properties are captured well by projective geometry.
- The horizon has the same projection.
- Since parallelism between lines is not preserved, distances or angles are not preserved either.
- Projective geometry describes a larger class of transformations. It is an extension of Euclidean geometry and deals with the perspective projection of a camera.
- Projective coordinates of a point in a plane are homogenized and represented by a 3-tuple: $[u, v, 1]^T$.
- Rule: Scaling the projective coordinates by a non-zero factor does not change the Euclidean point it represents as it is homogenized. i.e., $[u, v, 1]^T \equiv [\lambda u, \lambda v, \lambda]^T$.

3. Projective Space

The Euclidean coordinates of a point in a plane can be represented by a 2-tuple: $[u, v]^T$. It's projective coordinates are obtained by appending a 1 to the vector: $[u, v, 1]^T$. By representing the point by this 3-tuple in projective coordinates, a one-to-one mapping is established between the 2D point in Euclidean coordinates and the corresponding point in projective coordinates. Thus, scaling the point by a non-zero zero factor does not change the Euclidean point it represents as it is homogenized. i.e., $[u, v, 1]^T \equiv [\lambda u, \lambda v, \lambda]^T$. Thus, projective coordinates represent naturally the operations performed by cameras.

Definition: The space of $(n+1)$-tuples of coordinates, with the rule that proportional (or scaled) $(n+1)$-tuples represent the same point, is called a *projective space* of dimension n, and is denoted $\mathbf{P}^n$.

In general, given coordinates in $\mathbf{R}^n$, the corresponding projective coordinates are obtained as,

$$[x_1, x_2, \dots, x_n]^T \xrightarrow{\mathbf{R}^n \to \mathbf{P}^n} [x_1, x_2, \dots, x_n, 1]^T. \qquad 3.1$$

To transform a point from projective coordinates back to Euclidean coordinates, we just need to divide by the last coordinate and the drop the last coordinate,

$$[x_1, x_2, \dots, x_n, x_{n+1}]^T \xrightarrow{\mathbf{P}^n \to \mathbf{R}^n} \left[\frac{x_1}{x_{n+1}}, \frac{x_2}{x_{n+1}}, \dots, \frac{x_n}{x_{n+1}}\right]^T. \qquad 3.2$$

Points with last coordinate $x_{n+1} \neq 0$ are usual points with representations in $\mathbf{R}^n$, but points of the form $[x_1, x_2, \dots, x_n, 0]^T$, do not have an equivalent representation in Euclidean coordinates. If we consider them as the limit of $[x_1, x_2, \dots, x_n, \lambda]^T$, when $\lambda \to 0$, (i.e. the limit of $[x_1/\lambda, x_2/\lambda, \dots, x_n/\lambda, 1]^T$) then they represent the limit of a point in $\mathbf{R}^n$ going to infinity in the direction $[x_1, x_2, \dots, x_n]^T$. Such points are called *points at infinity*.

Thus projective space contains more points than the Euclidean space of same dimensions, and is a union of the usual space $\mathbf{R}^n$ and the set of points at infinity. i.e.,

$$\mathbf{P}^n = \mathbf{R}^n \cup \{[x_1, x_2, \dots, x_n, 0]^T\}. \qquad 3.3$$

As a result of this formalism, points at infinity are represented without exceptions in projective space.

Once the projection has been captured by the image, the true 3D depth of the point ***M***, can no longer be inferred from a single image due to the inherent nature of 3D to 2D projection. Thus any other point, ***M'***=$[\lambda X, \lambda Y, \lambda Z]^T$, that lies on the optical ray (***C, M***), also has the same 2D-projection, ***m***. This depth ambiguity cannot be inferred from a single image of the point using geometry alone, and the only information available from the single image projection is the vector along which the 3D point lies in space.

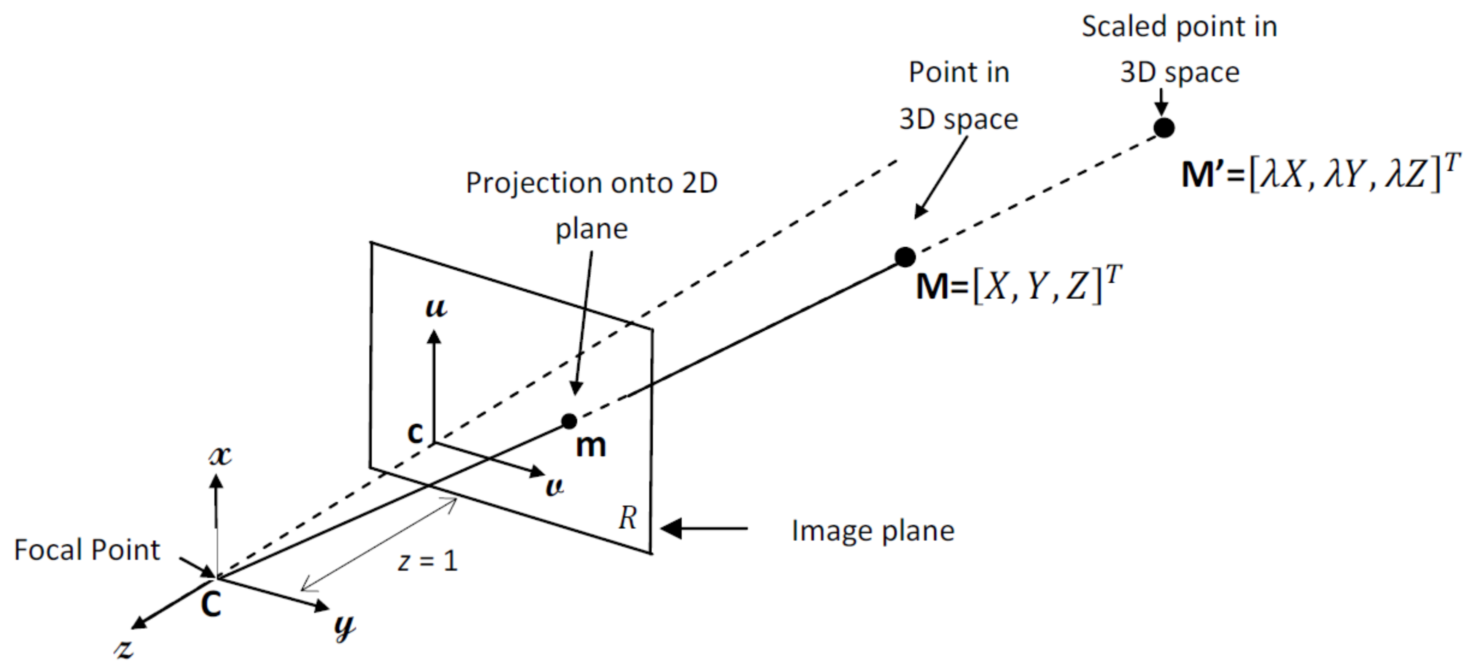


Fig.2. – Perspective projection of a 3D point onto a 2D image plane

Robot Mapping

A Short Introduction to Homogeneous Coordinates

Cyrill Stachniss

Motivation

- Cameras generate a projected image of the world
- Euclidian geometry is suboptimal to describe the central projection
- In Euclidian geometry, the math can get difficult
- Projective geometry is an alternative algebraic representation of geometric objects and transformations
- Math becomes simpler

Projective Geometry

- Projective geometry does not change the geometric relations
- Computations can also be done in Euclidian geometry (but more difficult)

Homogeneous Coordinates

- H.C. are a system of coordinates used in projective geometry
- Formulas involving H.C. are often simpler than in the Cartesian world
- Points at infinity can be represented using finite coordinates
- A single matrix can represent affine transformations and projective transformations

Homogeneous Coordinates

- H.C. are a system of coordinates used in projective geometry
- Formulas involving H.C. are often simpler than in the Cartesian world
- Points at infinity can be represented using finite coordinates
- **A single matrix can represent affine transformations and projective transformations**

Homogeneous Coordinates

Definition

- The representation $\mathbf{x}$ of a geometric object is homogeneous if $\mathbf{x}$ and $\lambda\mathbf{x}$ represent the same object for $\lambda \neq 0$

Example

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

From Homogeneous to Euclidian Coordinates

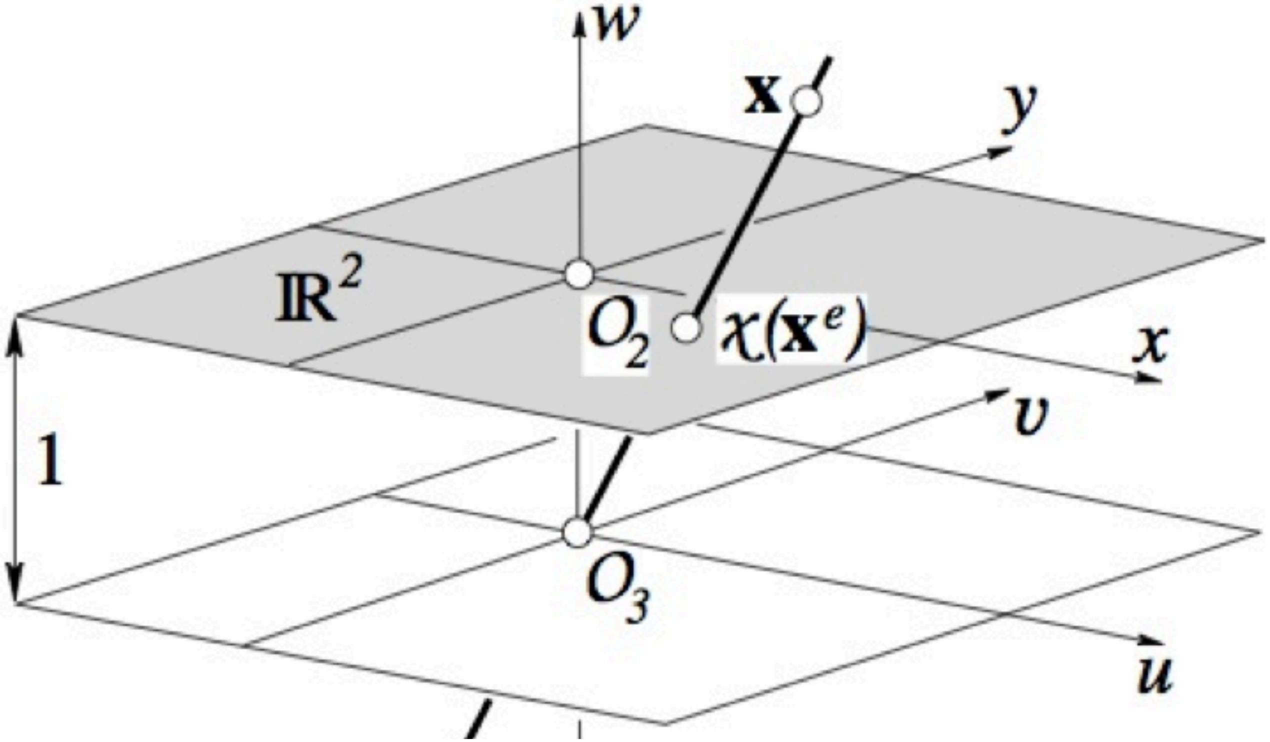
homogeneous

Euclidian

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u/w \\ v/w \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

From Homogeneous to Euclidian Coordinates



$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u/w \\ v/w \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

[Courtesy by K. Schindler]

Center of the Coordinate System

$$\mathbf{O}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \mathbf{O}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Infinitively Distant Objects

- It is possible to explicitly model infinitively distant points with finite coordinates

$$\mathbf{x}_\infty = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix}$$

- Great tool when working with bearing-only sensors such as cameras

3D Points

- Analogous for 3D points

homogeneous — Euclidian

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \\ t \end{bmatrix} = \begin{bmatrix} u/t \\ v/t \\ w/t \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u/t \\ v/t \\ w/t \end{bmatrix}$$

Transformations

- A projective transformation is a invertible linear mapping

$$\mathbf{x}' = M\mathbf{x}$$

Important Transformations ($\mathbb{P}^3$)

- General projective mapping

$$\mathbf{x}' = \underset{4\times 4}{M}\ \mathbf{x}$$

- Translation: 3 parameters
  (3 translations)

$$M = \lambda \begin{bmatrix} I & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Important Transformations ($\mathbb{P}^3$)

- Rotation: 3 parameters
  (3 rotation)

$$M = \lambda \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

rotation
matrix

Recap – Rotation Matrices

$$R^{2D}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$R_x^{3D}(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{bmatrix} \quad R_y^{3D}(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

$$R_z^{3D}(\kappa) = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^{3D}(\omega, \phi, \kappa) = R_z^{3D}(\kappa) R_y^{3D}(\phi) R_x^{3D}(\omega)$$

Important Transformations ($\mathbb{P}^3$)

- Rotation: 3 parameters
  (3 rotation)

$$M = \lambda \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

- Rigid body transformation: 6 params
  (3 translation + 3 rotation)

$$M = \lambda \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

Important Transformations ($\mathbb{P}^3$)

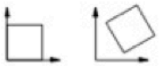
- Similarity transformation: 7 params
  (3 trans + 3 rot + 1 scale)

$$M = \lambda \begin{bmatrix} mR & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

- Affine transformation: 12 parameters
  (3 trans + 3 rot + 3 scale + 3 sheer)

$$M = \lambda \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

Transformations in $\mathbb{P}^2$

2D Transformation	Figure	d. o. f.	H	H
Translation		2	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} I & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Mirroring at y-axis		1	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Rotation		1	$\begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Motion		3	$\begin{bmatrix} \cos\varphi & -\sin\varphi & t_x \\ \sin\varphi & \cos\varphi & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Similarity		4	$\begin{bmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \lambda R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Scale difference		1	$\begin{bmatrix} 1+m/2 & 0 & 0 \\ 0 & 1-m/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} D & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Shear		1	$\begin{bmatrix} 1 & s/2 & 0 \\ s/2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} S & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Asym. shear		1	$\begin{bmatrix} 1 & s' & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} S' & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Affinity		6	$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
Projectivity		8	$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$	$\begin{bmatrix} A & \mathbf{t} \\ \mathbf{p}^\top & 1/\lambda \end{bmatrix}$

[Courtesy by K. Schindler]

Transformations

- Inverting a transformation

$$\begin{aligned} \mathbf{x}' &= M\mathbf{x} \\ \mathbf{x} &= M^{-1}\mathbf{x}' \end{aligned}$$

- Chaining transformations via matrix products (not commutative)

$$\begin{aligned} \mathbf{x}' &= M_1 M_2 \mathbf{x} \\ &\neq M_2 M_1 \mathbf{x} \end{aligned}$$

Motions

- We will express motions (rotations and translations) using H.C.

$$M = \lambda \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

- Chaining transformations via matrix products (not commutative)

$$\begin{aligned} \mathbf{x}' &= M_1 M_2 \mathbf{x} \\ &\neq M_2 M_1 \mathbf{x} \end{aligned}$$

Conclusion

- Homogeneous coordinates are an alternative representation for geometric objects
- Equivalence up to scale

$$\mathbf{x} \equiv \lambda \mathbf{x} \text{ with } \lambda \neq 0$$

- Modeled through an extra dimension
- Homogeneous coordinates can simplify mathematical expressions
- We often use it to represent the motion of objects

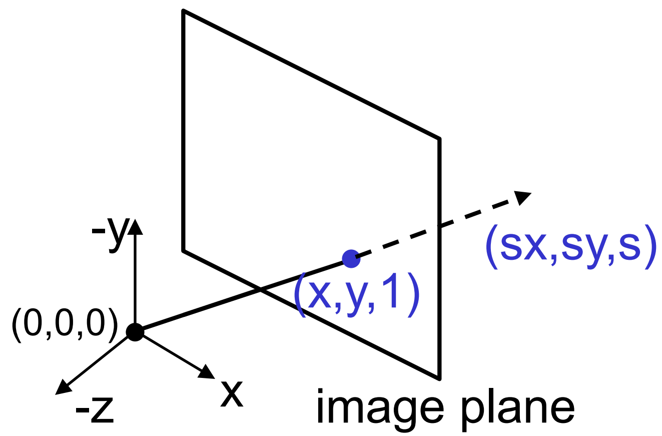
Literature

TOPIC

- Wikipedia as a good summary on homogeneous coordinates:
  http://en.wikipedia.org/wiki/Homogeneous_coordinates

The projective plane

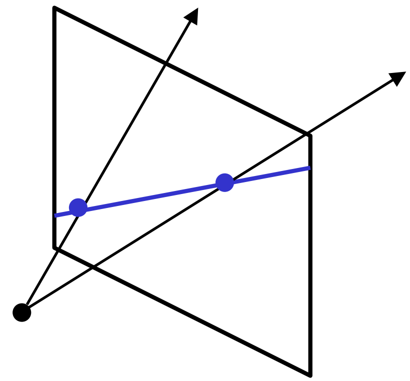
- Why do we need homogeneous coordinates?
  - represent points at infinity, homographies, perspective projection, multi-view relationships
- What is the geometric intuition?
  - a point in the image is a *ray* in projective space



- Each *point* (x,y) on the plane is represented by a *ray* (sx,sy,s)
  – all points on the ray are equivalent: $(x, y, 1) \cong (sx, sy, s)$

Projective lines

- What does a line in the image correspond to in projective space?



- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

in vector notation :

$$0 = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

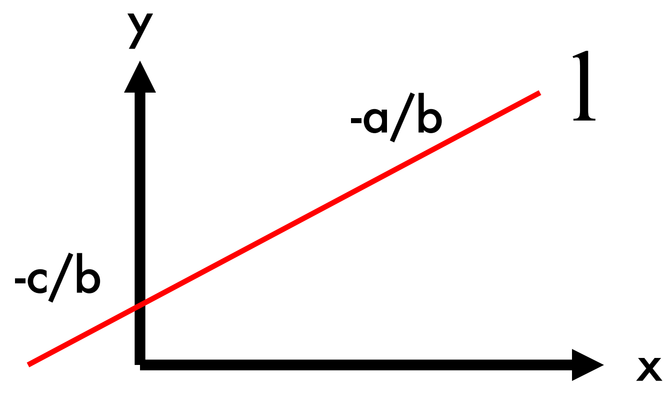
$$\qquad \mathbf{l} \qquad \mathbf{p}$$

- A line is also represented as a homogeneous 3-vector **l**

Lines in a 2D plane

$$ax + by + c = 0$$

$$\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



$$\text{If } \; x = [\, x_1, x_2]^T \in \mathbf{l}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}^T \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

[Eq. 10]

Lines in 2-D

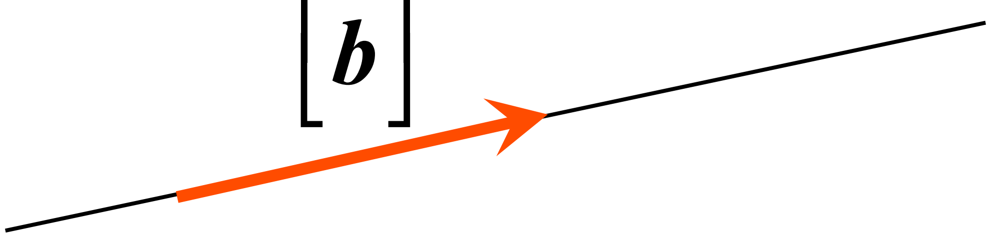
- General equation of a line in 2-D:

$$\boldsymbol{ax} + \boldsymbol{by} + \boldsymbol{c} = 0$$

- In homogeneous coordinates:

$$\boldsymbol{l}^T \boldsymbol{p} = \boldsymbol{l} \bullet \boldsymbol{p} = 0 \quad \boldsymbol{l} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \end{bmatrix} \quad \boldsymbol{p} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

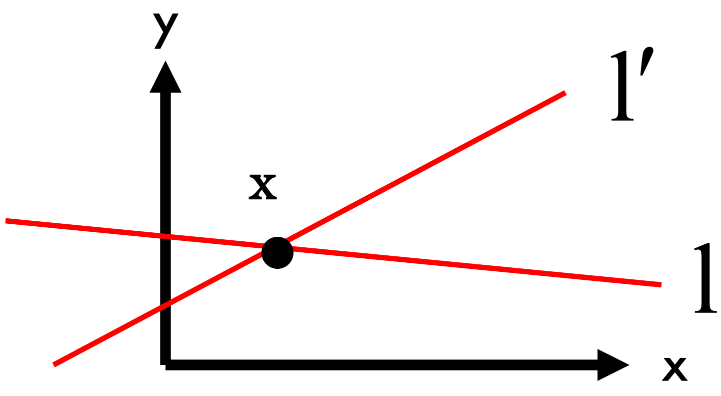
$$\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}$$



Lines in a 2D plane

Intersecting lines

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' \quad \text{[Eq. 11]}$$



Proof

$$\mathbf{l} \times \mathbf{l}' \perp \mathbf{l} \quad \rightarrow (\mathbf{l} \times \mathbf{l}') \cdot \mathbf{l} = 0 \quad \rightarrow x \in l \quad \text{[Eq. 12]}$$

$$\mathbf{l} \times \mathbf{l}' \perp \mathbf{l}' \quad \rightarrow \underbrace{(\mathbf{l} \times \mathbf{l}')}_{\mathbf{x}} \cdot \mathbf{l}' = 0 \quad \rightarrow x \in l' \quad \text{[Eq. 13]}$$

$\rightarrow$ x is the intersection point

Points from lines and vice-versa

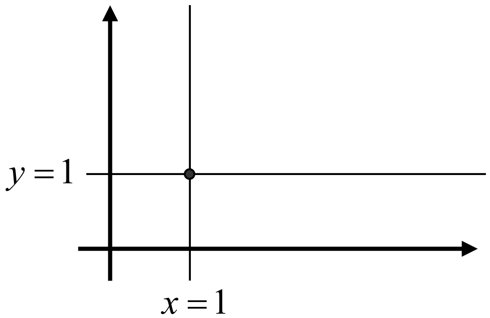
Intersections of lines

The intersection of two lines $\mathbf{l}$ and $\mathbf{l}'$ is $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$

Line joining two points

The line through two points $\mathbf{x}$ and $\mathbf{x}'$ is $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$

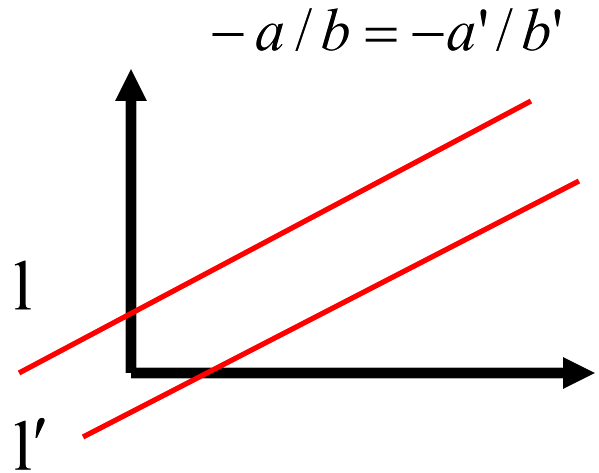
Example



2D Points at infinity (ideal points)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, x_3 \neq 0$$

$$x_\infty = \begin{bmatrix} x'_1 \\ x'_2 \\ 0 \end{bmatrix}$$



$$\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$l' = \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}$$

Let's intersect two parallel lines: $\rightarrow l \times l' \propto \begin{bmatrix} b \\ -a \\ 0 \end{bmatrix} = x_\infty$ [Eq.13]

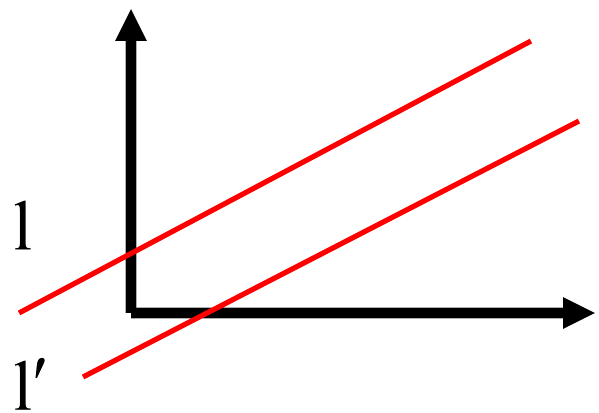
= ideal point!

- In Euclidian coordinates this point is at infinity
- Agree with the general idea of two lines intersecting at infinity

2D Points at infinity (ideal points)

$$\mathrm{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \mathrm{x}_3 \end{bmatrix}, \mathrm{x}_3 \neq 0$$

$$-a/b = -a'/b'$$



$$\mathrm{l} = \begin{bmatrix} \mathrm{a} \\ \mathrm{b} \\ \mathrm{c} \end{bmatrix}$$

$$l' = \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}$$

Note: the line l = [a b c]^T pass trough the ideal point x_∞

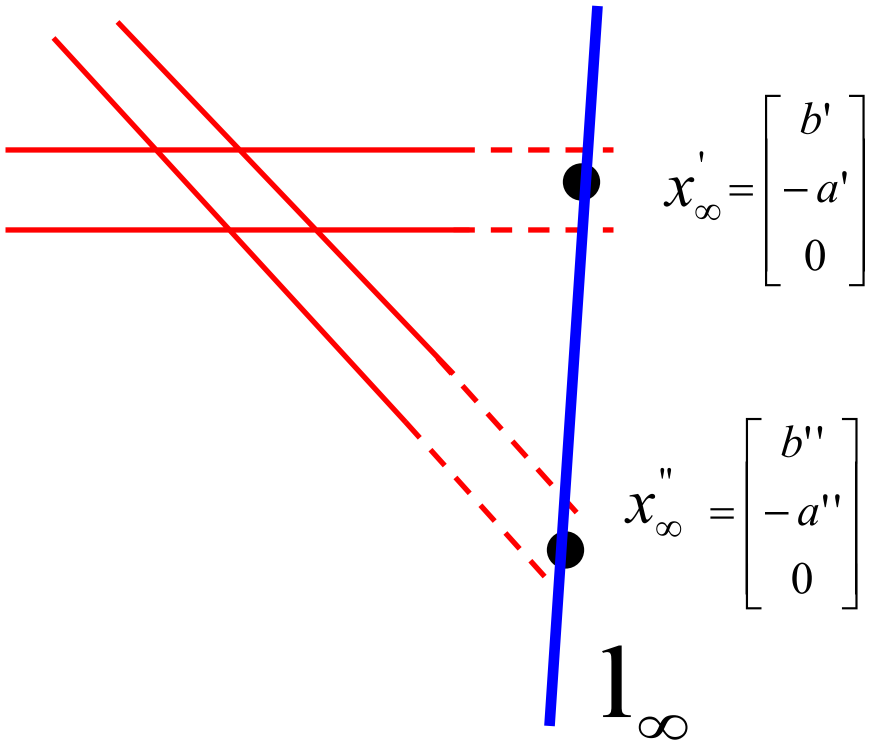
$$\mathrm{l}^{\mathrm{T}}\, x_\infty = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} b \\ -a \\ 0 \end{bmatrix} = 0 \quad \text{[Eq. 15]}$$

So does the line l′ since a b′ = a′ b

Lines infinity l_∞

Set of ideal points lies on a line called the line at infinity.
How does it look like?

$$l_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Indeed: $\begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$

A line at infinity can be thought of the set of "directions" of lines in the plane

Transformation of lines

For points on a line l, the transformed points under proj. trans. also lie on a line; if point x is on line l, then transforming x, transforms l

For a point transformation

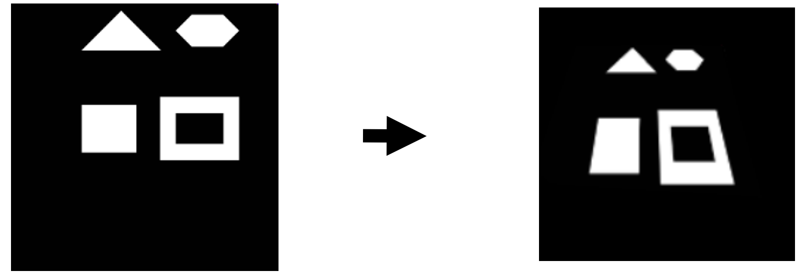
$$\mathbf{x}' = \mathbf{H}\,\mathbf{x}$$

Transformation for lines

$$\mathbf{l}' = \mathbf{H}^{-T}\,\mathbf{l}$$

Projective transformation of a point at infinity

$$H = \begin{bmatrix} A & t \\ v & b \end{bmatrix}$$



$$p' = H\ p$$

is it a point at infinity?

$$H\ p_\infty = ? \quad = \begin{bmatrix} A & t \\ v & b \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix}$$

...no!

[Eq. 17]

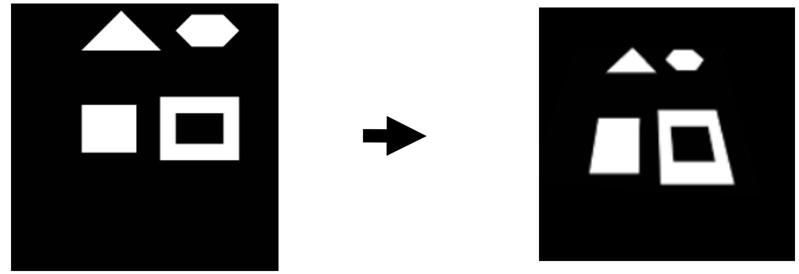
$$H_A\ p_\infty = ? \quad = \begin{bmatrix} A & t \\ 0 & b \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p'_x \\ p'_y \\ 0 \end{bmatrix}$$

[Eq. 18]

An affine transformation of a point at infinity is still a point at infinity

Projective transformation of a line (in 2D)

$$\mathrm{H} = \begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v} & \mathrm{b} \end{bmatrix}$$



$$\mathrm{l}' = \mathrm{H}^{-\mathrm{T}}\ \mathrm{l}$$

[Eq. 19]

is it a line at infinity?

$$\mathrm{H}^{-\mathrm{T}}\ \mathrm{l}_{\infty} = ? = \begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v} & \mathrm{b} \end{bmatrix}^{-\mathrm{T}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{t}_{\mathrm{x}} \\ \mathrm{t}_{\mathrm{y}} \\ \mathrm{b} \end{bmatrix}$$

...no!

[Eq. 20]

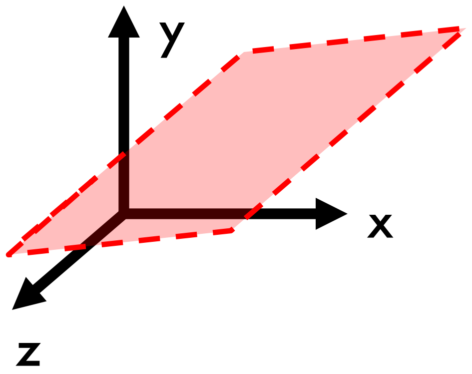
$$\mathrm{H}_{\mathrm{A}}^{-\mathrm{T}}\ \mathrm{l}_{\infty} = ? \quad = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}^{-T} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} A^{-T} & 0 \\ -t^{T} A^{-T} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

[Eq. 21]

Points and planes in 3D

$$\mathrm{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \mathrm{x}_3 \\ 1 \end{bmatrix}$$

$$\Pi = \begin{bmatrix} \mathrm{a} \\ \mathrm{b} \\ \mathrm{c} \\ \mathrm{d} \end{bmatrix}$$



$$x \in \Pi \leftrightarrow x^T \Pi = 0$$

[Eq. 22]

$$\mathrm{ax} + \mathrm{by} + \mathrm{cz} + \mathrm{d} = 0$$

[Eq. 23]

3D points

3D point

$$(X, Y, Z)^\mathsf{T} \text{ in } \mathbf{R}^3$$

$$\mathrm{X} = (X_1, X_2, X_3, X_4)^\mathsf{T} \text{ in } \mathbf{P}^3$$

$$\mathrm{X} = \left(\frac{X_1}{X_4}, \frac{X_2}{X_4}, \frac{X_3}{X_4}, 1 \right)^\mathsf{T} = (X, Y, Z, 1)^\mathsf{T} \qquad (X_4 \neq 0)$$

projective transformation

$$\mathbf{X}' = \mathbf{H}\,\mathbf{X} \qquad \text{(4x4-1=15 dof)}$$

Planes

3D plane

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0$$

$$\pi_1 X_1 + \pi_2 X_2 + \pi_3 X_3 + \pi_4 X_4 = 0$$

$$\pi^{\mathsf{T}} \mathrm{X} = 0$$

Transformation

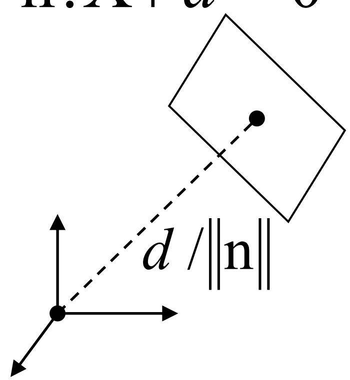
$$\mathrm{X}' = \mathbf{H}\,\mathrm{X}$$

$$\pi' = \mathbf{H}^{-\mathsf{T}}\,\pi$$

Euclidean representation

$$\mathrm{n}.\tilde{\mathrm{X}} + d = 0 \qquad \mathrm{n} = (\pi_1, \pi_2, \pi_3)^{\mathsf{T}} \qquad \tilde{\mathrm{X}} = (X, Y, Z)^{\mathsf{T}}$$

$$\pi_4 = d \qquad X_4 = 1$$



Dual: points ↔ planes, lines ↔ lines

Planes in 3-D

- General equation of a plane in 3D:

$$\boldsymbol{ax} + \boldsymbol{by} + \boldsymbol{cz} + \boldsymbol{d} = 0$$

- In homogeneous coordinates:

$$\Pi^T \boldsymbol{p} = \Pi \bullet \boldsymbol{p} = 0 \quad \Pi = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} \quad \boldsymbol{p} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ 1 \end{bmatrix}$$

Lines in 3D

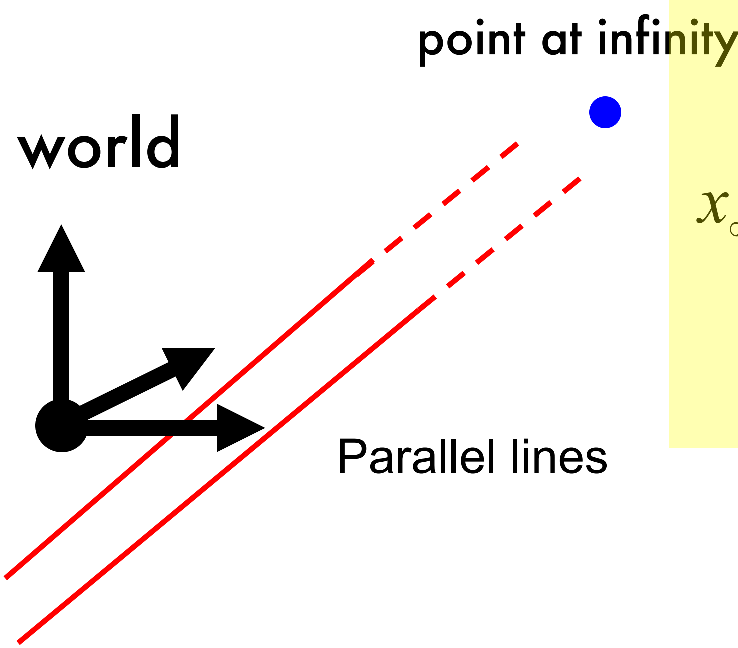
- Lines have 4 degrees of freedom - hard to represent in 3D-space

- Can be defined as intersection of 2 planes

$$\begin{aligned}\mathbf{d} &= \text{direction of the line} \\ &= [a, b, c]^T\end{aligned}$$

Points at infinity in 3D

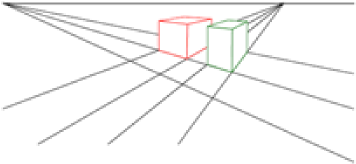
Points where parallel lines intersect in 3D

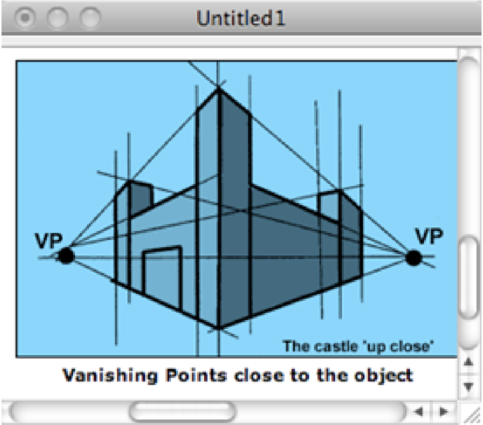


$$x_\infty = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \end{bmatrix}$$

Vanishing points

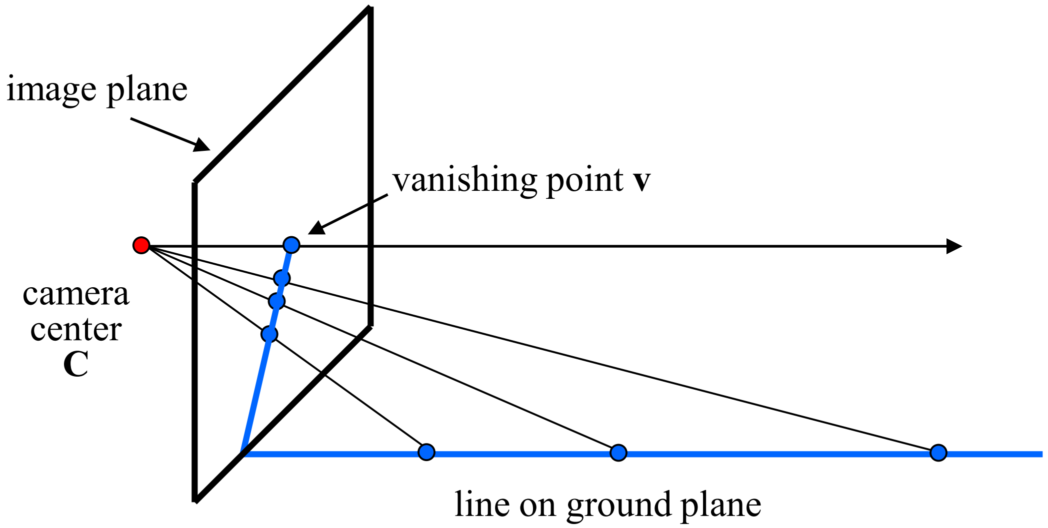
- Each set of parallel lines (=direction) meets at a different point
  - The *vanishing point* for this direction
- Sets of parallel lines on the same plane lead to *collinear* vanishing points.
  - The line is called the *horizon* for that plane





http://www.ider.herts.ac.uk/school/courseware/graphics/two_point_perspective.html

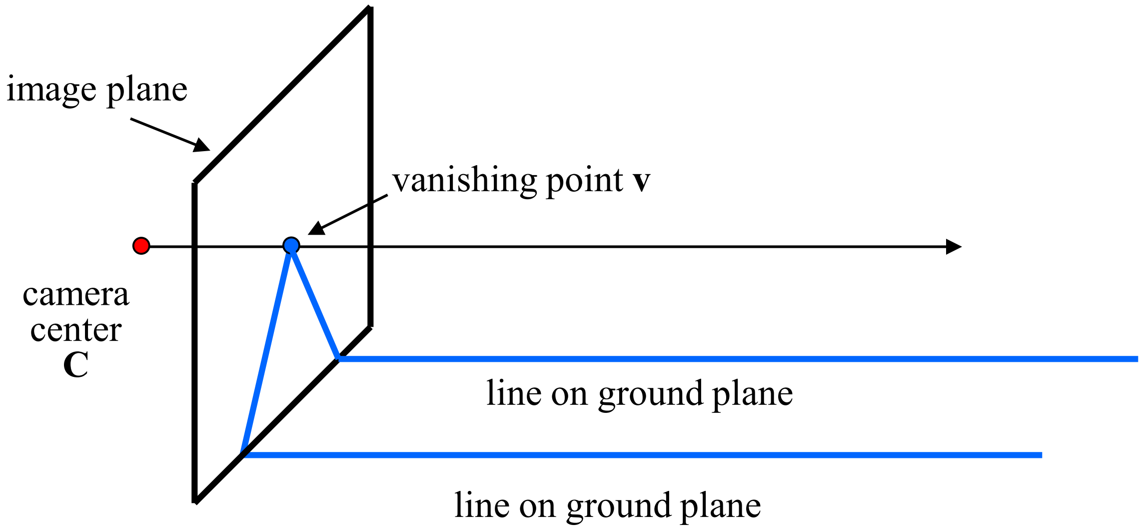
Vanishing points



Vanishing point

- projection of a point at infinity

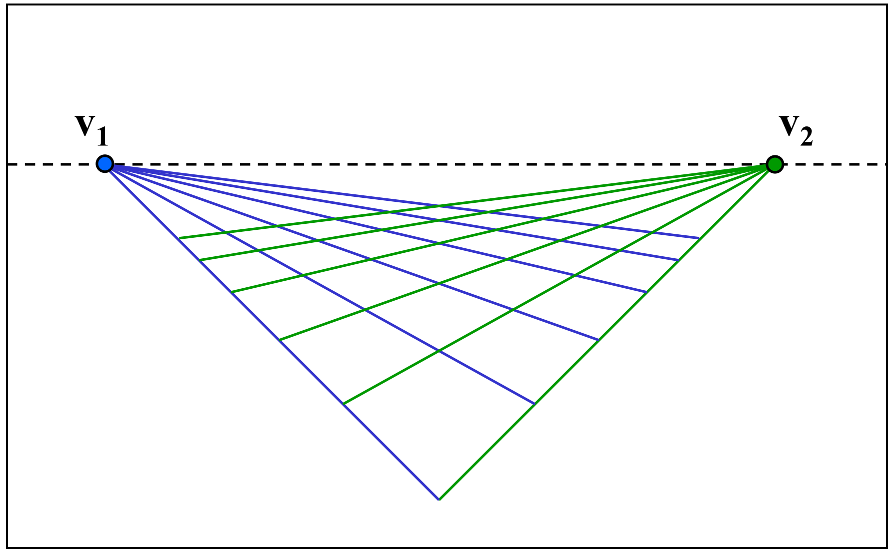
Vanishing points



Properties

- Any two parallel lines have the same vanishing point **v**
- The ray from **C** through **v** is parallel to the lines
- An image may have more than one vanishing point
  - in fact every pixel is a potential vanishing point

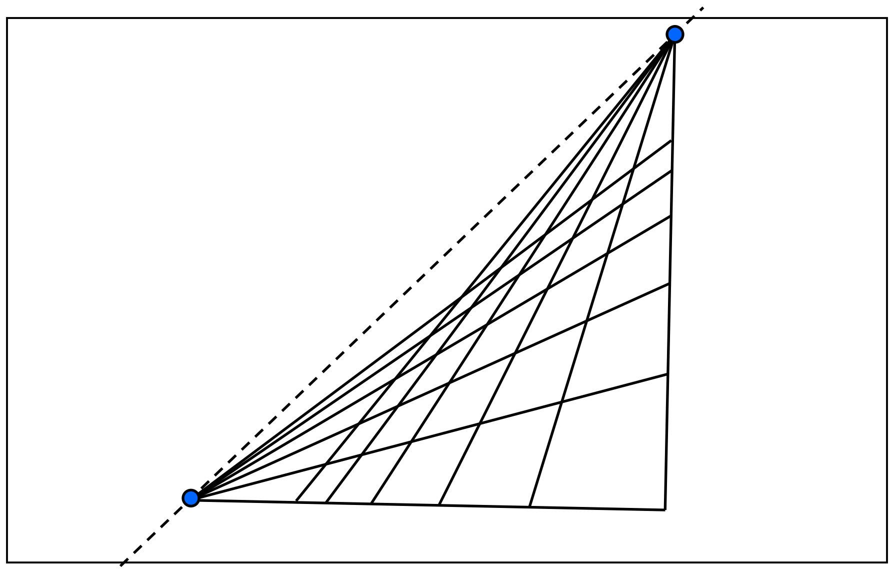
Vanishing lines



Multiple Vanishing Points

- Any set of parallel lines on the plane define a vanishing point
- The union of all of vanishing points from lines on the same plane is the *vanishing line*
  – For the ground plane, this is called the *horizon*

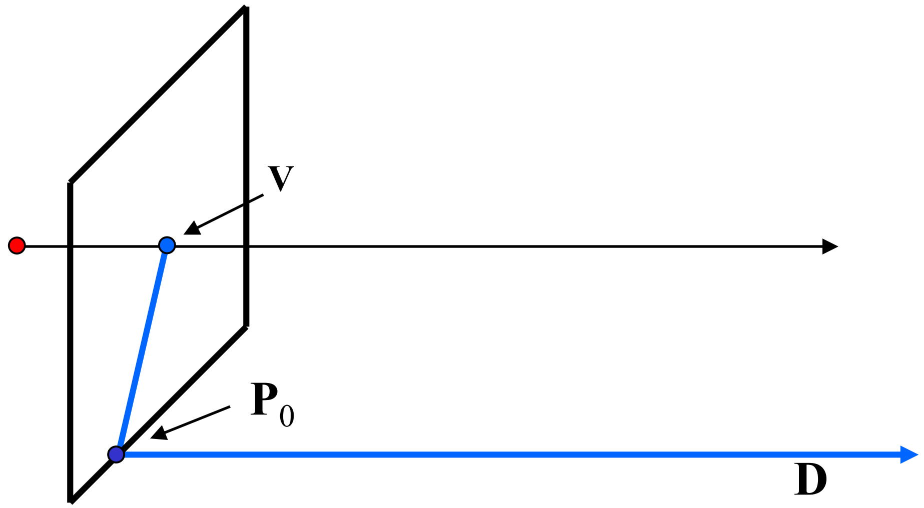
Vanishing lines



Multiple Vanishing Points

- Different planes define different vanishing lines

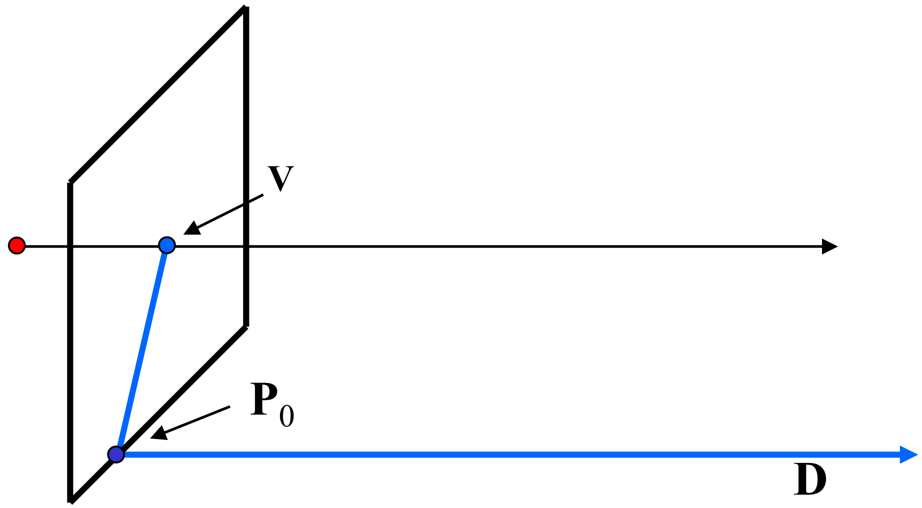
Computing vanishing points



$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix}$$

Computing vanishing points



$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X / t + D_X \\ P_Y / t + D_Y \\ P_Z / t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

Properties $\mathbf{v} = \mathbf{\Pi} \mathbf{P}_\infty$

- $\mathbf{P}_\infty$ is a point at *infinity*, **v** is its projection
- They depend only on line *direction*
- Parallel lines $\mathbf{P}_0$ + t**D**, $\mathbf{P}_1$ + t**D** intersect at $\mathbf{P}_\infty$

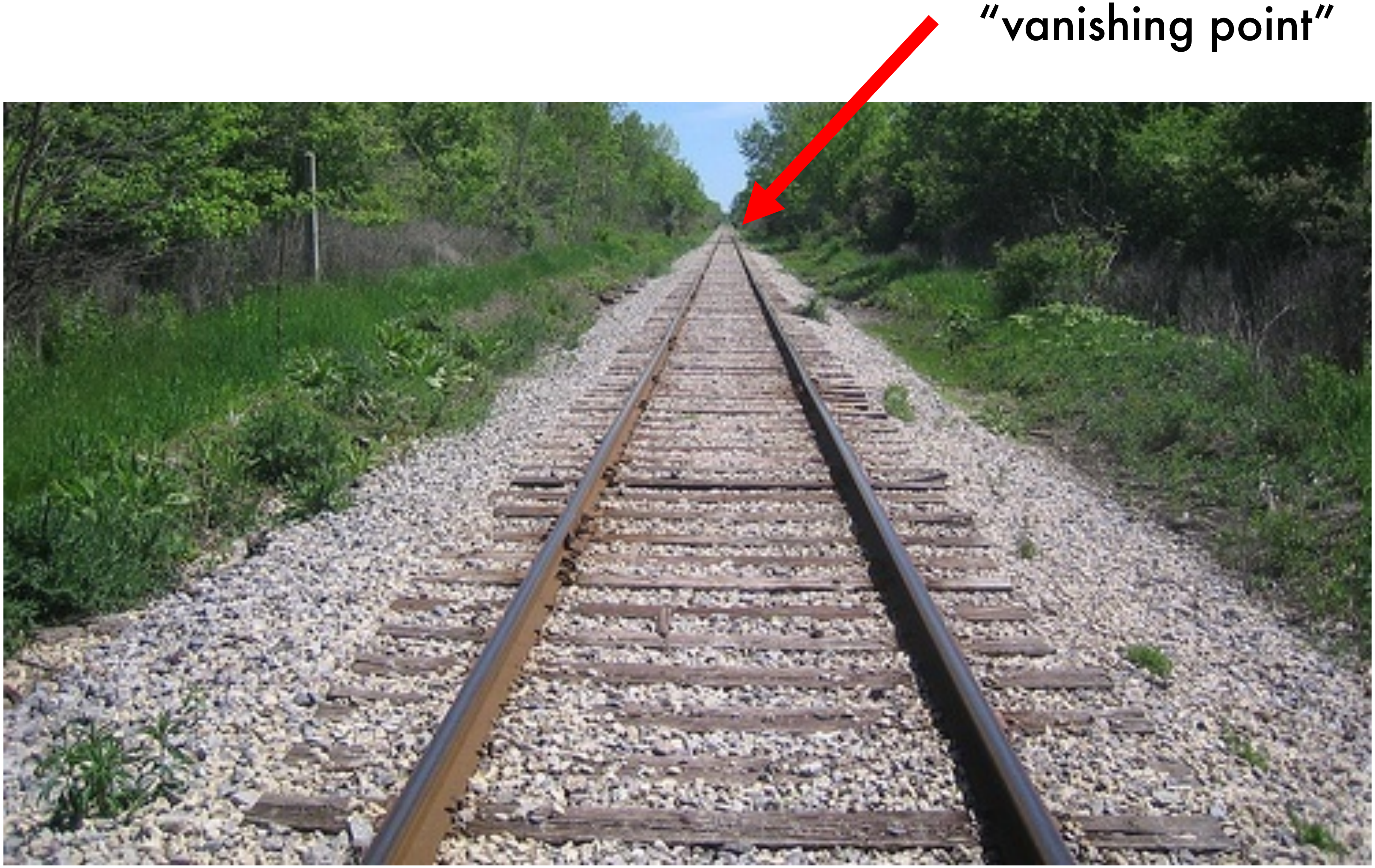
Properties of projective transformations

- Points project to points
- Lines project to lines
- Distant objects look smaller

Properties of Projection

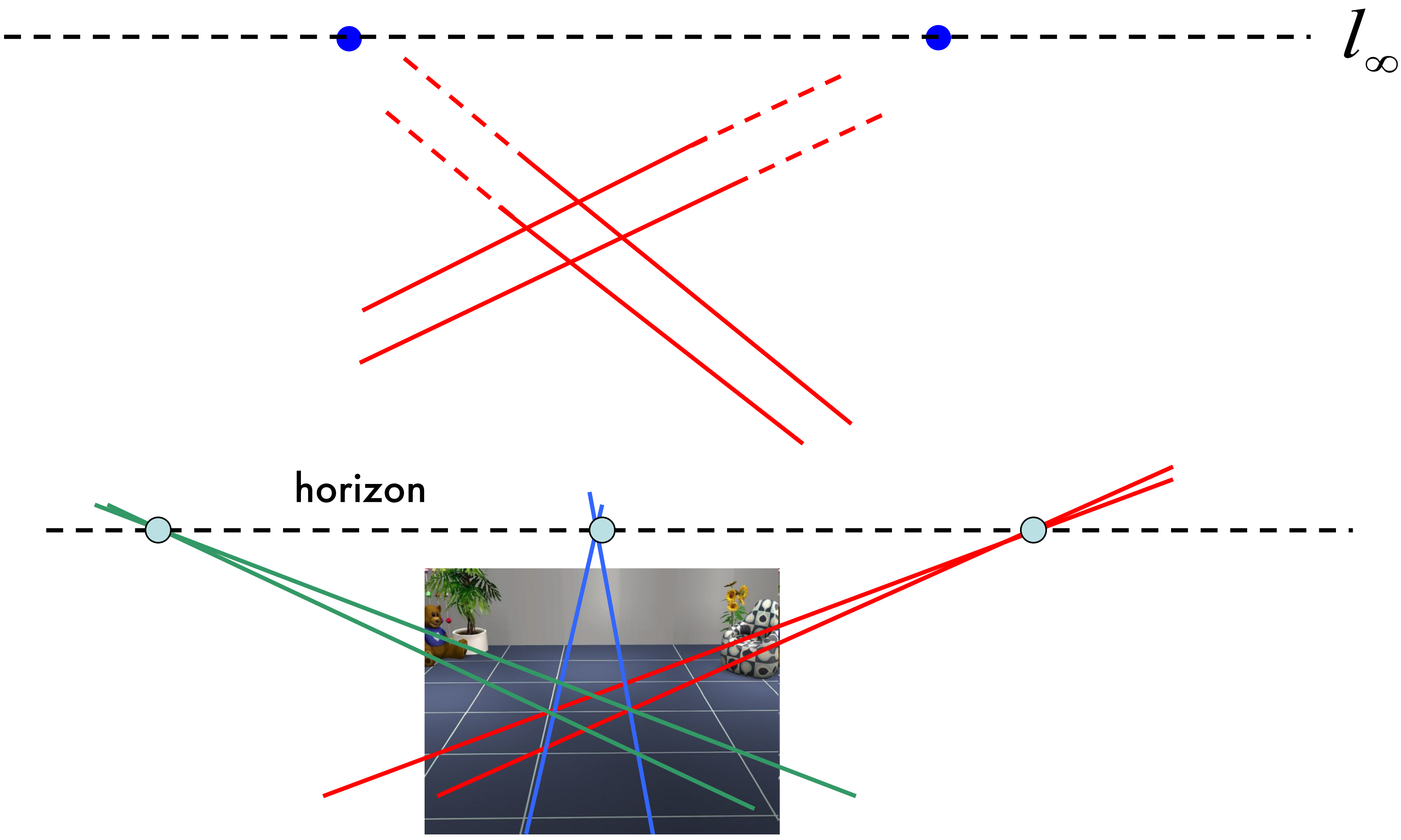
- Angles are not preserved
- Parallel lines meet!

Parallel lines in the world intersect in the image at a "vanishing point"



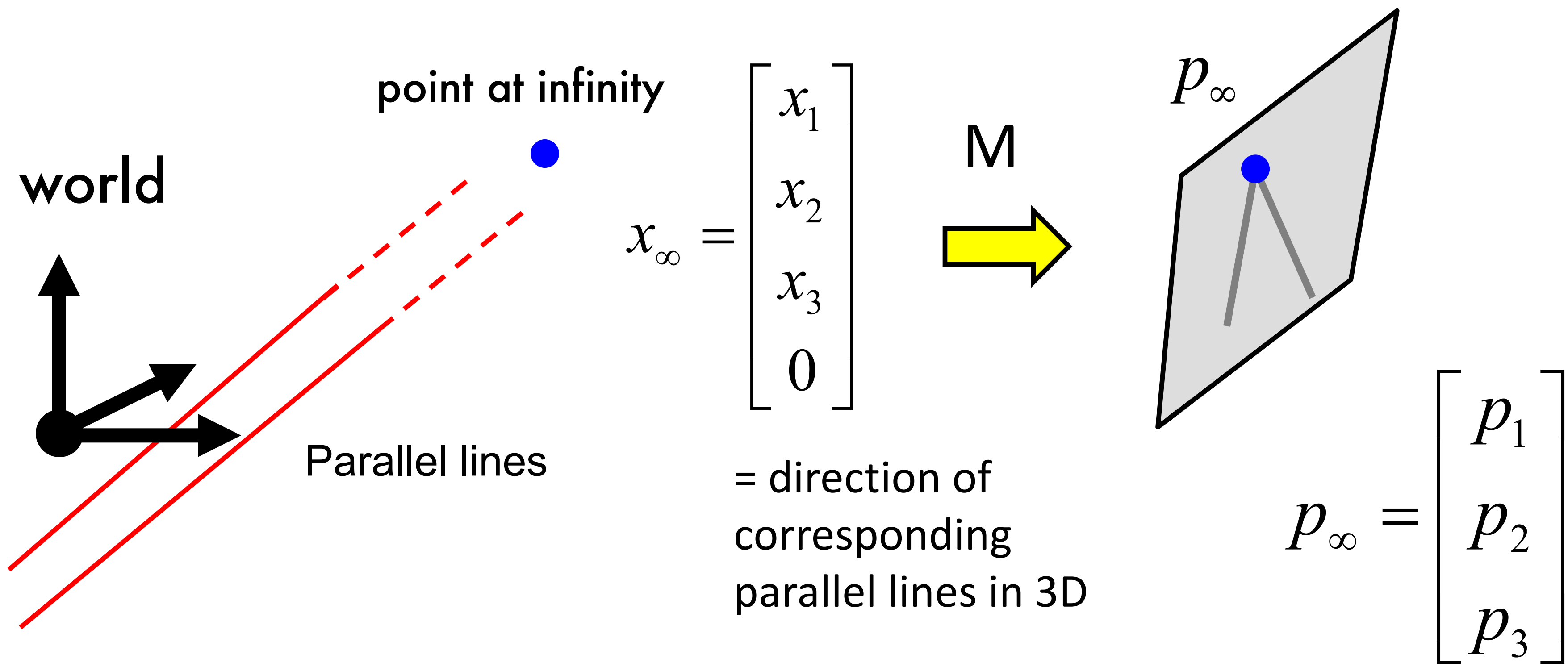
Horizon line (vanishing line)

Horizon line (vanishing line)

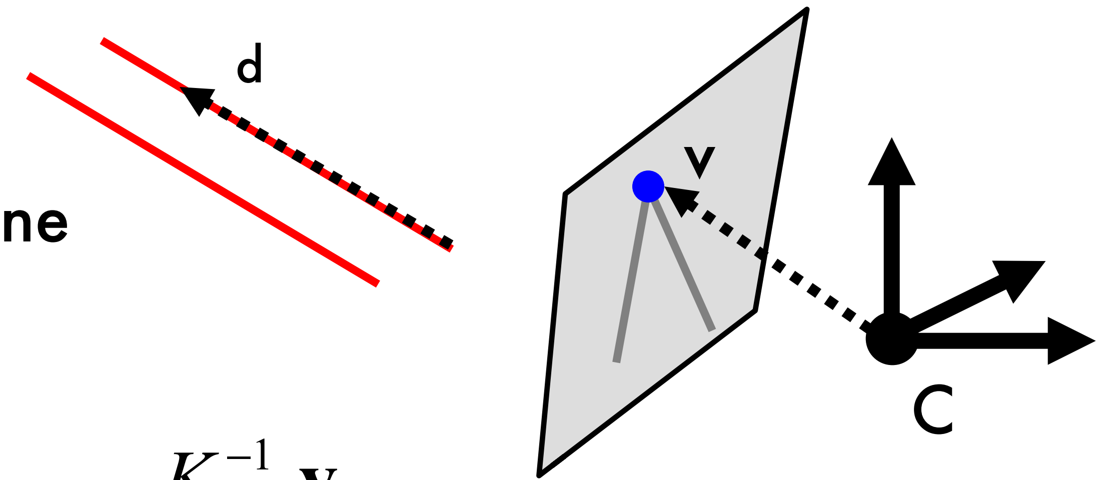


Vanishing points

The projective projection of a point at infinity into the image plane defines a vanishing point.



Vanishing points and directions



d = direction of the line
= [a, b, c]^T

$$\mathbf{v} = K\,\mathbf{d}$$

[Eq. 24]

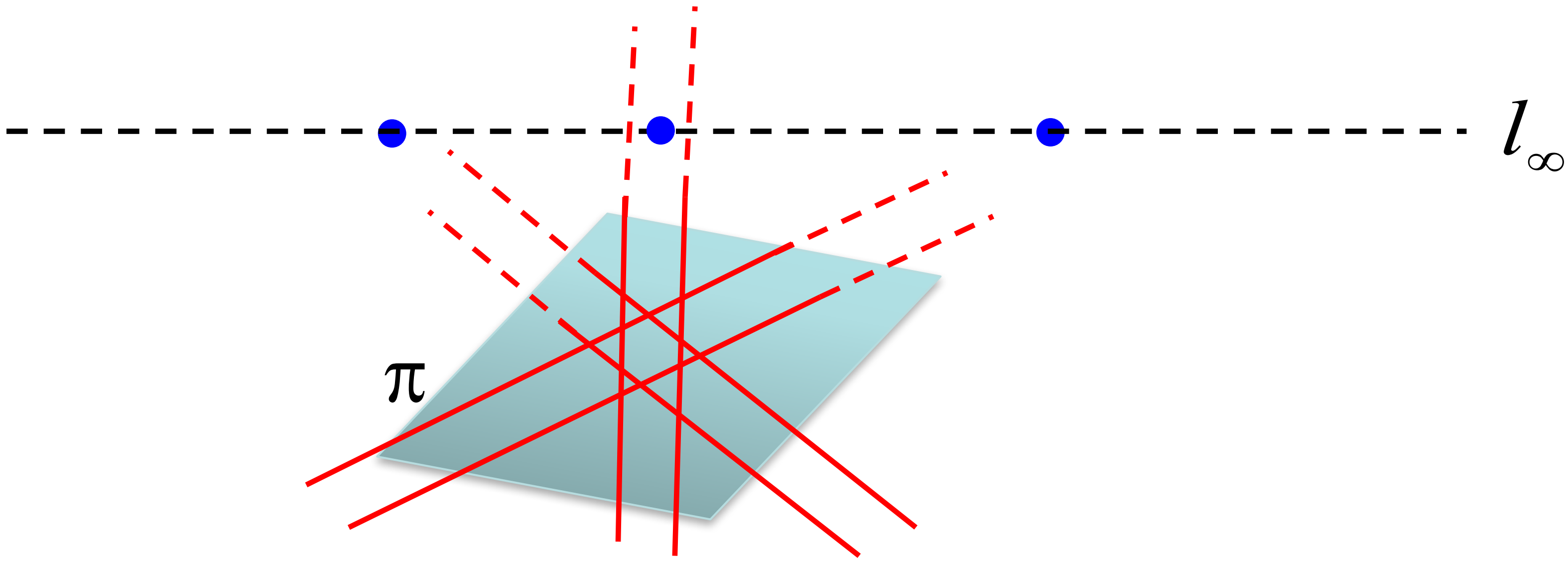
$$\mathbf{d} = \frac{K^{-1}\,\mathbf{v}}{\left\| K^{-1}\,\mathbf{v} \right\|}$$

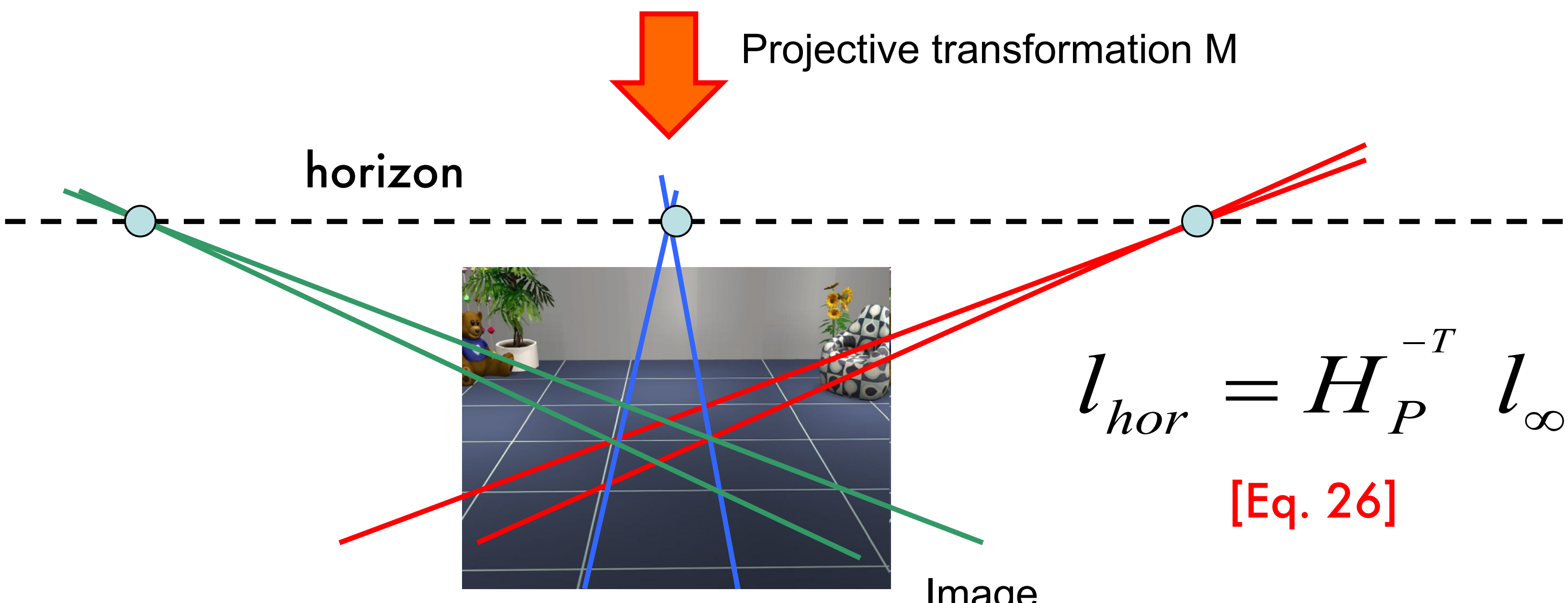
[Eq. 25]

Proof:

$$X_\infty = \begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix} \xrightarrow{M} \mathbf{v} = \mathbf{M}\,\mathbf{X}_\infty = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix} = \mathbf{K}\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Vanishing (horizon) line

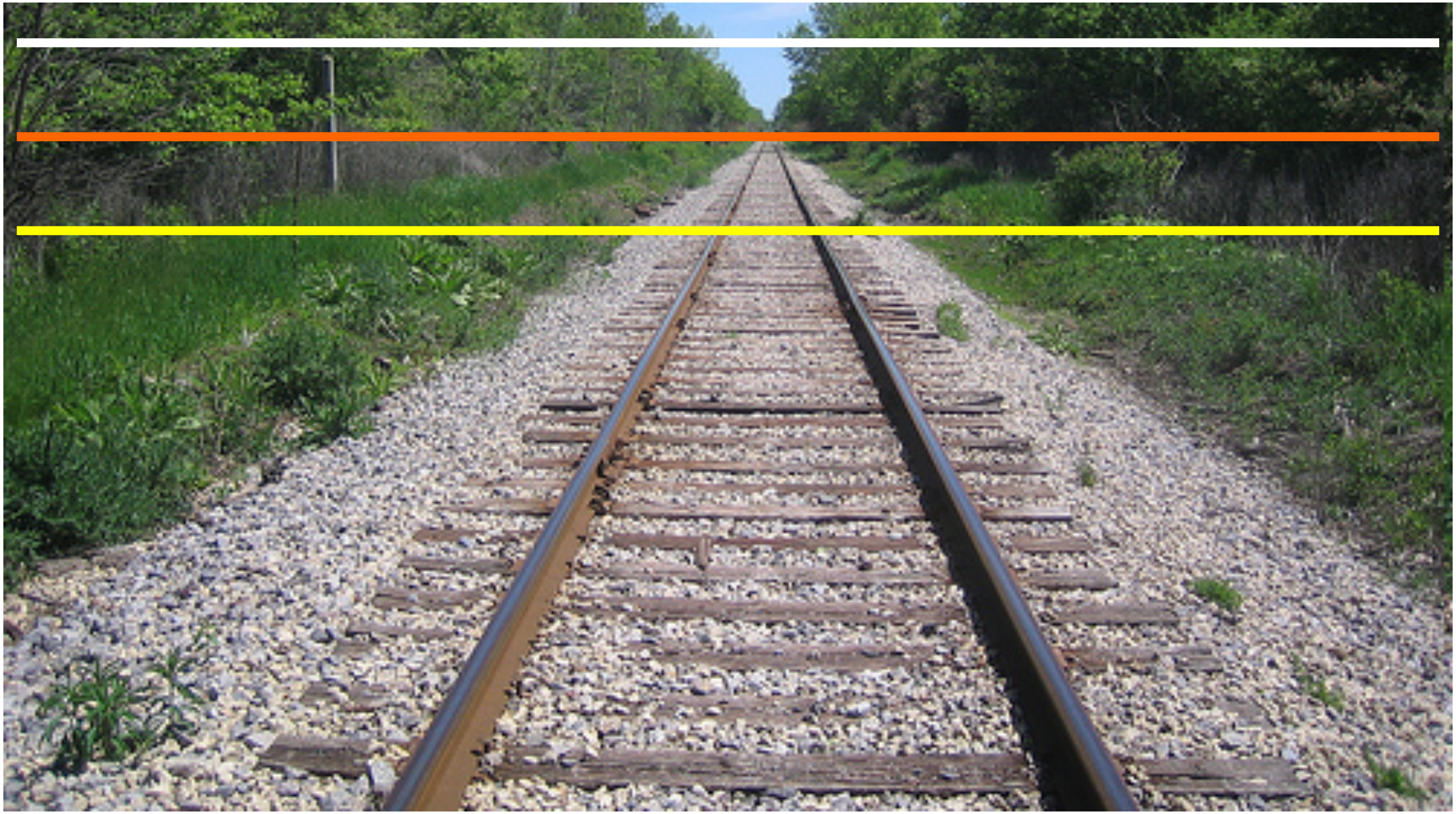




$$l_{hor} = H_P^{-T}\ l_\infty$$

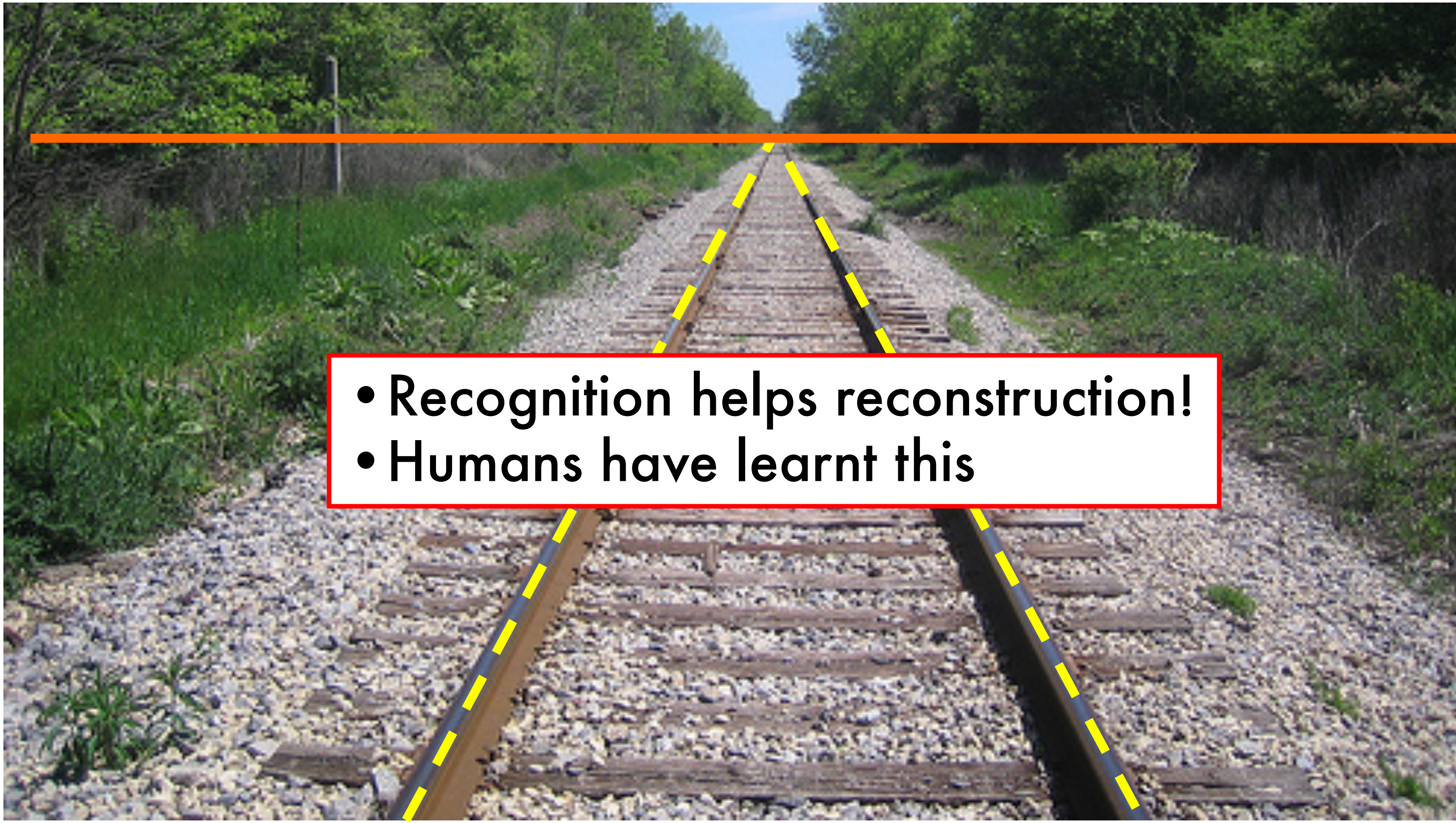
[Eq. 26]

Example of horizon line



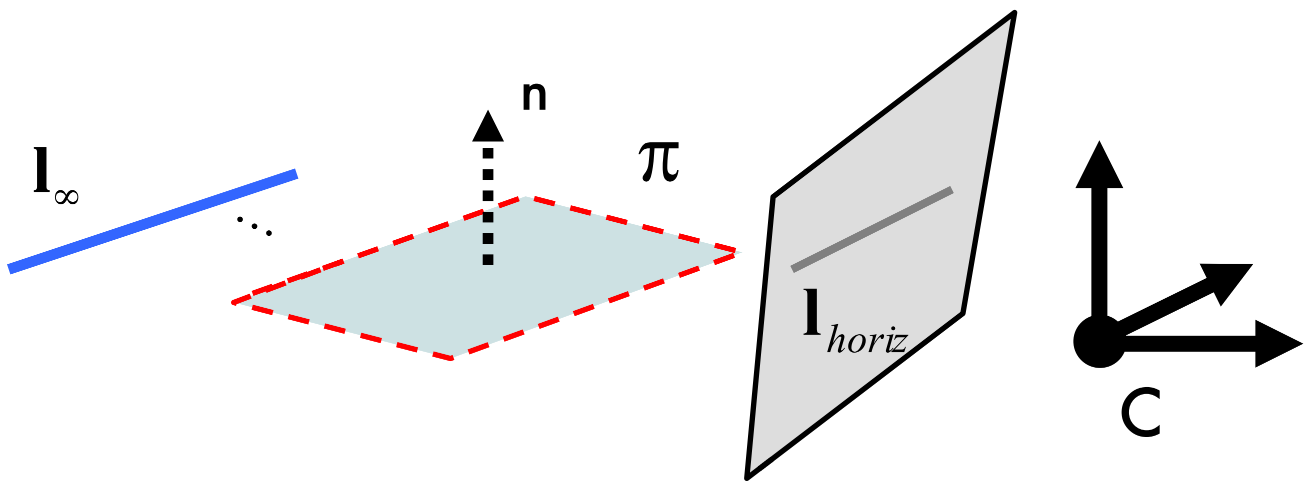
The orange line is the horizon!

Are these two lines parallel or not?



- Recognize the horizon line
- Measure if the 2 lines meet at the horizon
- if yes, these 2 lines are // in 3D

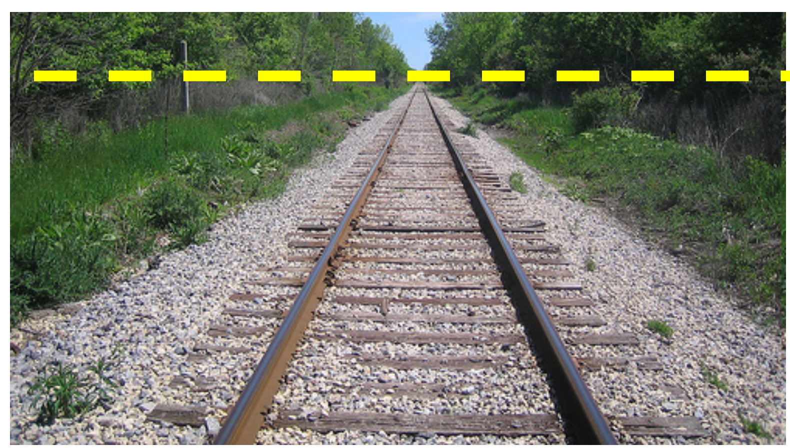
Vanishing points and planes



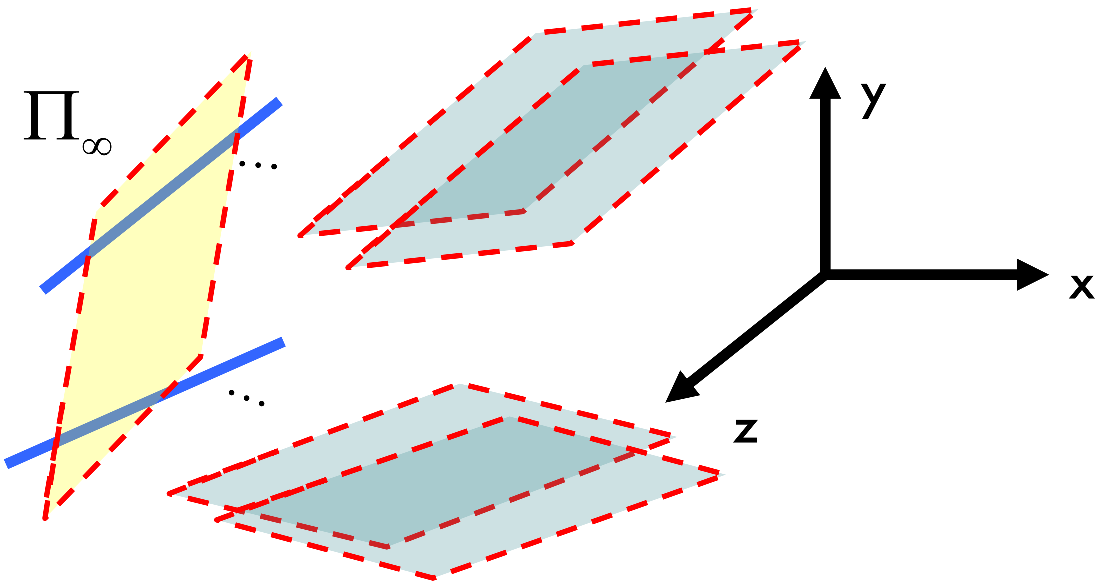
$$\mathbf{n} = K^T \mathbf{l}_{horiz}$$

[Eq. 27]

See sec. 8.6.2 [HZ] for details



Planes at infinity

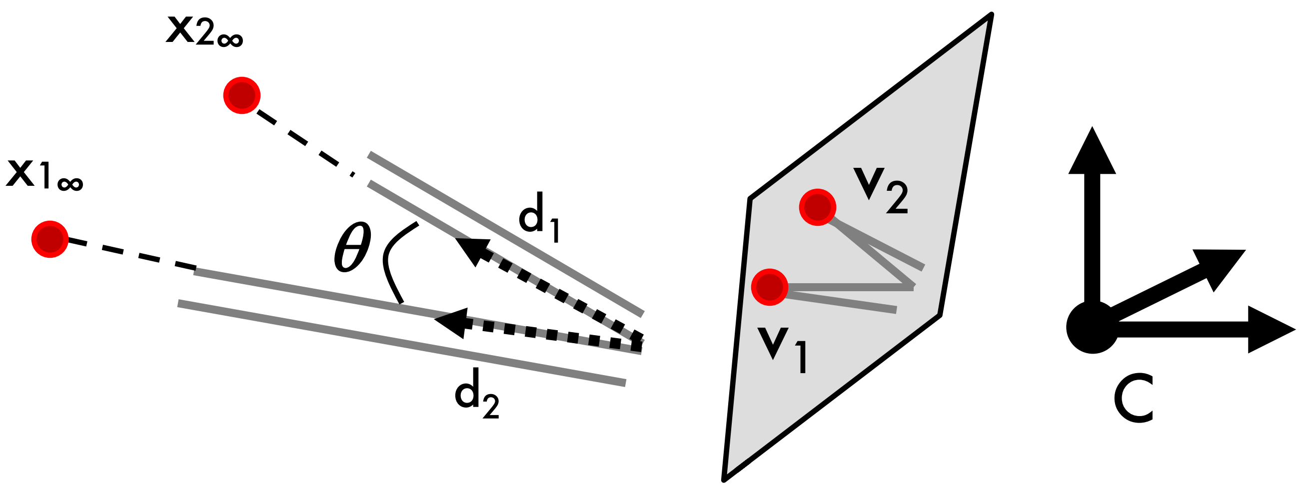


$$\Pi_\infty = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

plane at infinity

- Parallel planes intersect at infinity in a common line – **the line at infinity**
- A set of 2 or more lines at infinity defines the plane at infinity Π_∞

Angle between 2 vanishing points



$$\cos\theta = \frac{v_1^T \omega\, v_2}{\sqrt{v_1^T \omega\, v_1}\sqrt{v_2^T \omega\, v_2}}$$

[Eq. 28]

$$\omega = (K\,K^T)^{-1}$$

[Eq. 30]

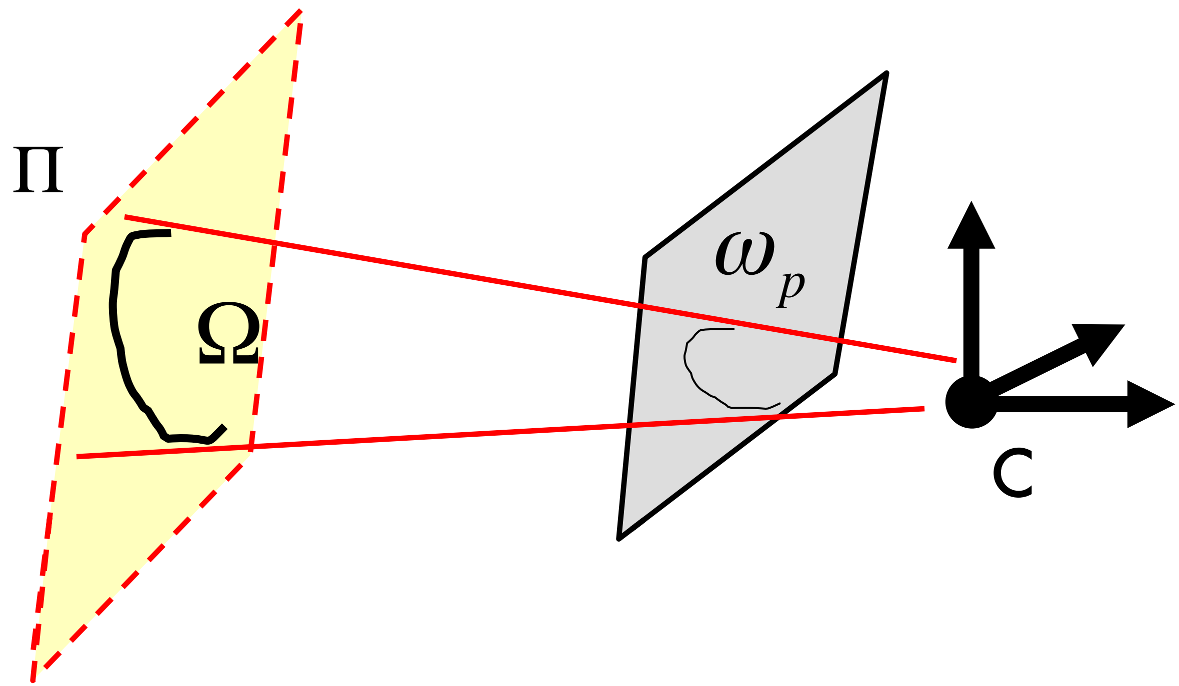
If $\theta = 90 \rightarrow$

$$v_1^T \omega\, v_2 = 0$$

[Eq. 29]

Scalar equation

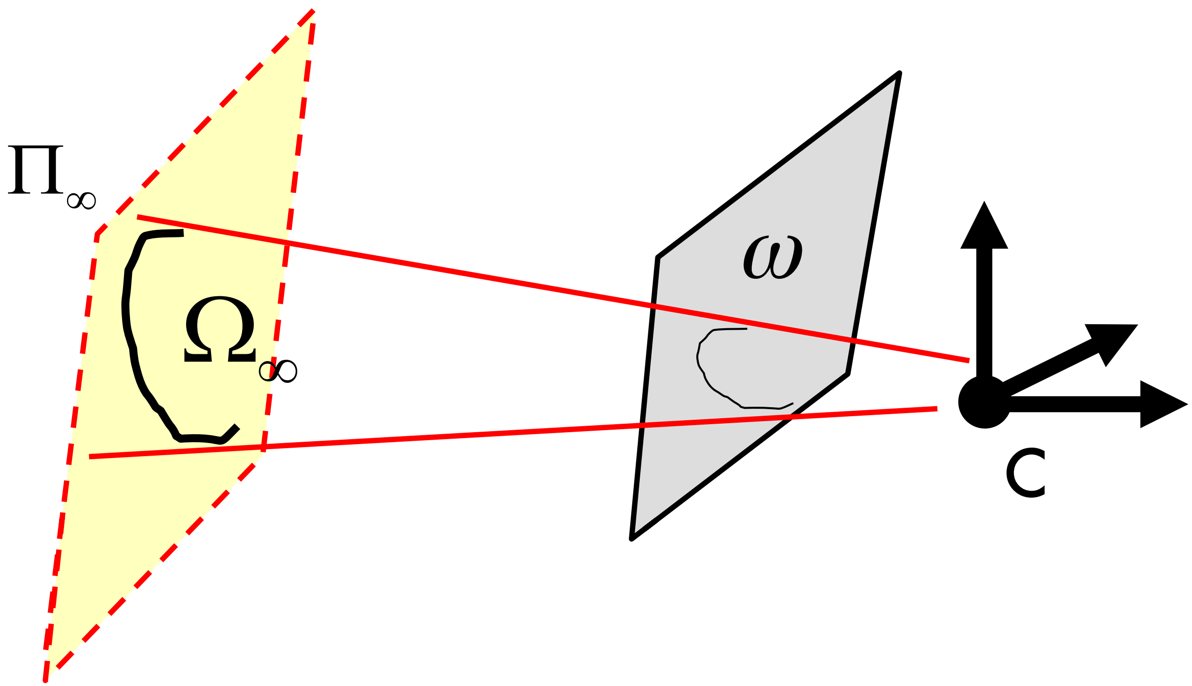
Projective transformation of a conic Ω



$$\omega_p = M^{-T} \Omega\, M^{-1}$$

HZ page 73, eq. 3.16

Projective transformation of Ω_∞ Absolute conic



$$\omega = M^{-T}\Omega_\infty M^{-1} = (K\,K^T)^{-1}$$

HZ page 73

Properties of ω

$$\omega = (K K^T)^{-1}$$

[Eq. 30]

$$M = K\begin{bmatrix} R & T \end{bmatrix}$$

1. $$\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_4 \\ \omega_2 & \omega_3 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}$$ symmetric and known up scale

2. $\omega_2 = 0$ zero-skew

3. $\omega_2 = 0$, $\omega_1 = \omega_3$ square pixel

Summary

$$\mathbf{v} = K\,\mathbf{d}$$
[Eq. 24]

$$\mathbf{n} = \mathrm{K}^{\mathrm{T}}\mathbf{l}_{\mathrm{horiz}}$$
[Eq. 27]

$$\cos\theta = \frac{\mathrm{v}_1^{\mathrm{T}}\,\boldsymbol{\omega}\,\mathrm{v}_2}{\sqrt{\mathrm{v}_1^{\mathrm{T}}\,\boldsymbol{\omega}\,\mathrm{v}_1}\,\sqrt{\mathrm{v}_2^{\mathrm{T}}\,\boldsymbol{\omega}\,\mathrm{v}_2}}$$
[Eq. 28]

$$\xrightarrow{\theta = 90}$$

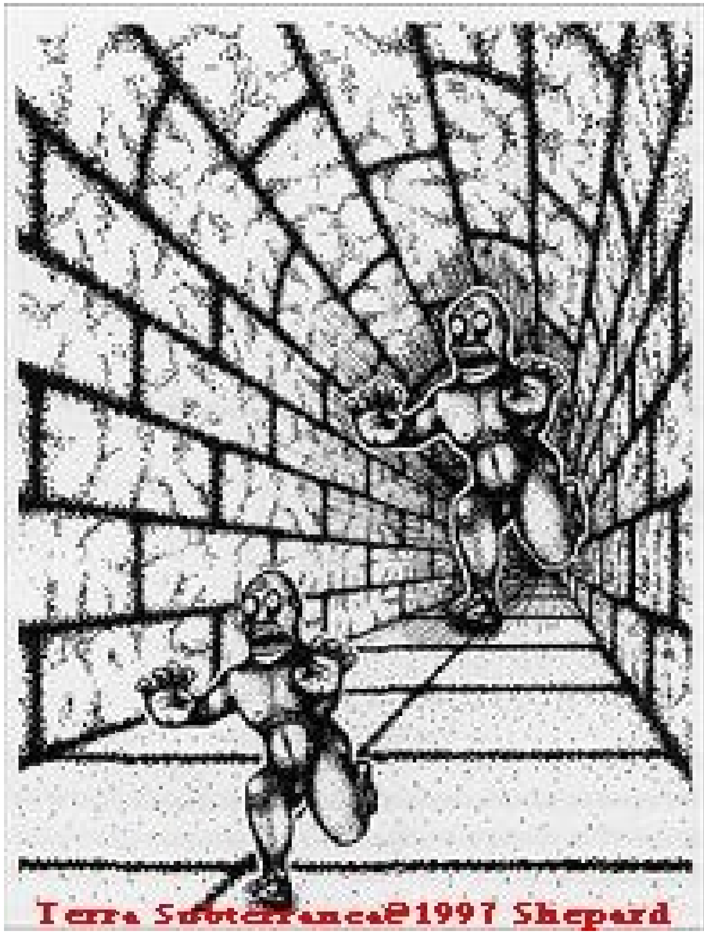
$$\mathrm{v}_1^{\mathrm{T}}\,\boldsymbol{\omega}\,\mathrm{v}_2 = 0$$
[Eq. 29]

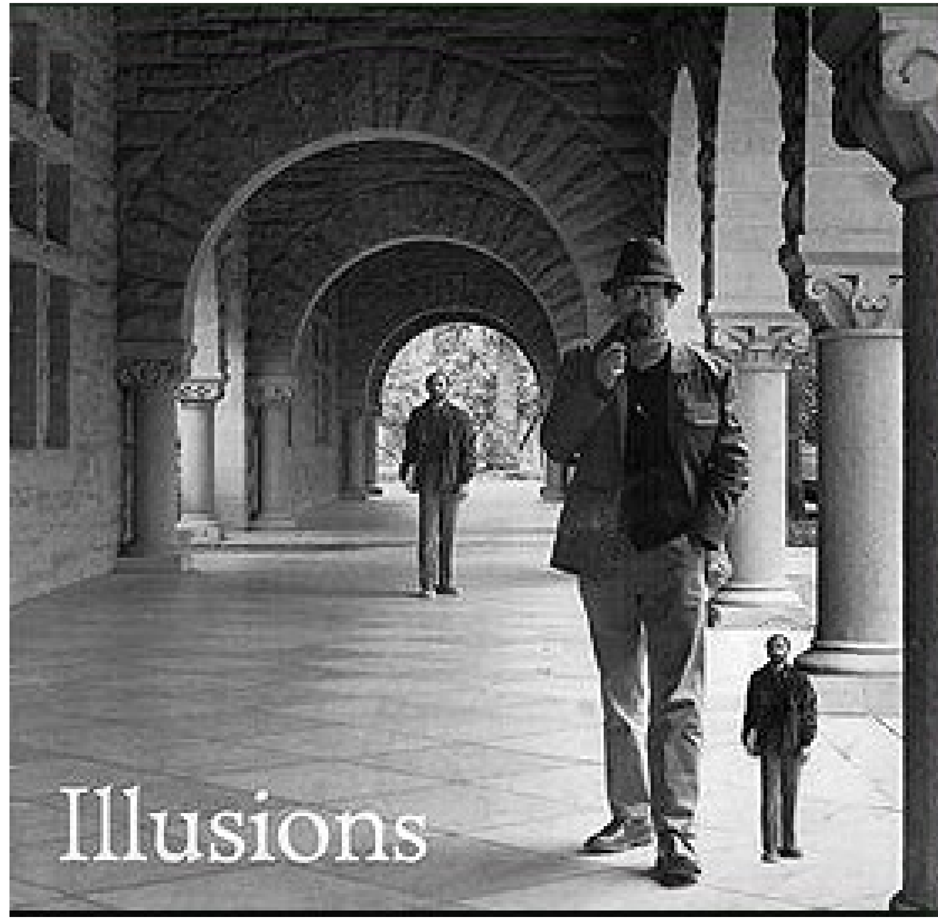
$$\omega = (K\,K^{T})^{-1}$$
[Eq. 30]

Useful to:

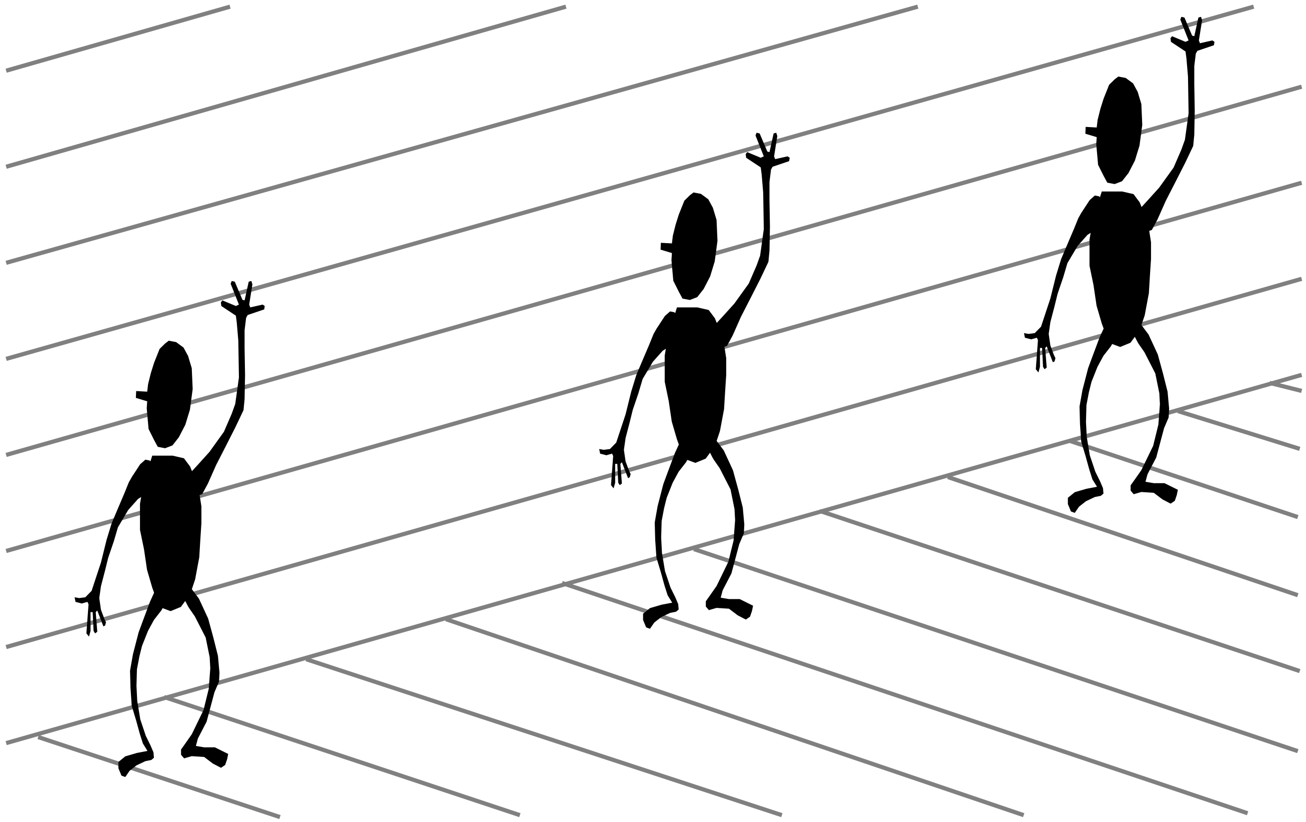
- To calibrate the camera
- To estimate the geometry of the 3D world

Fun with vanishing points

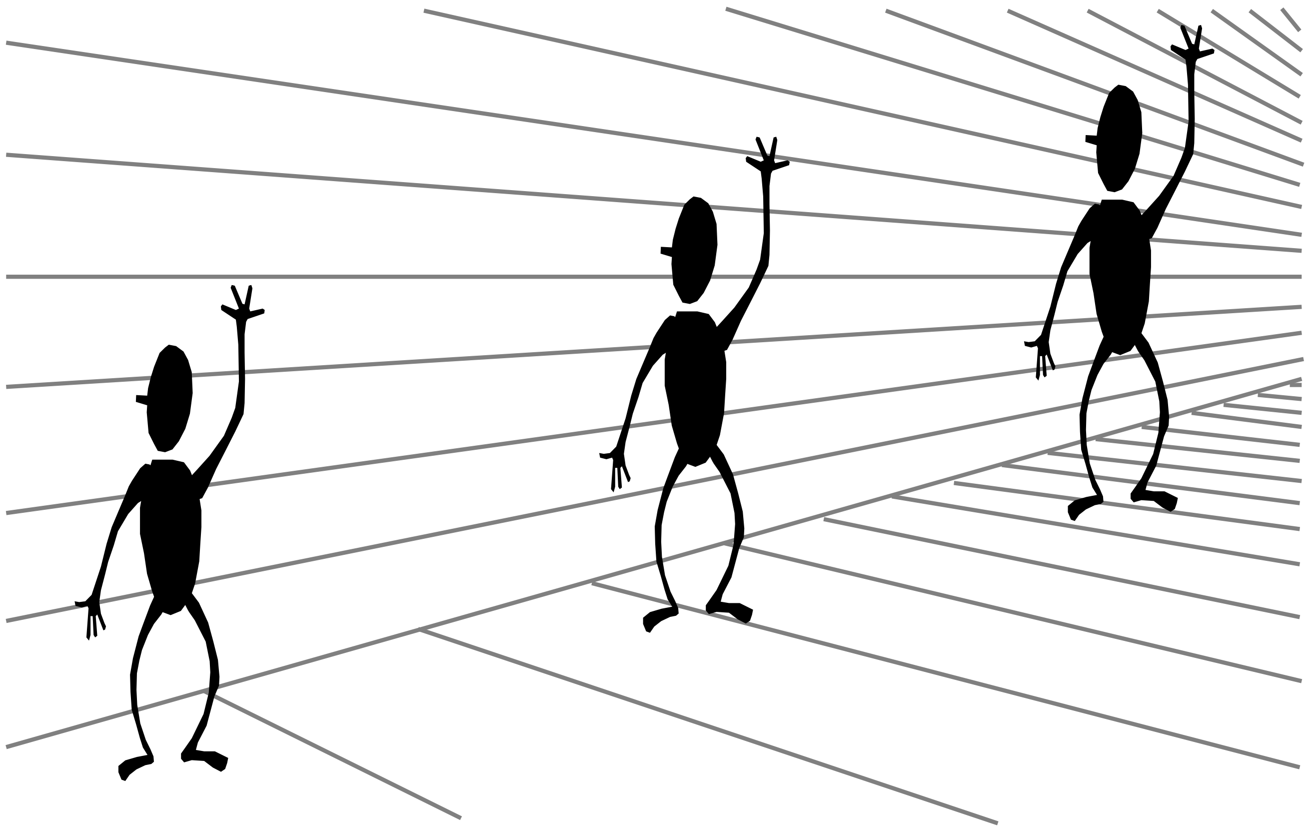




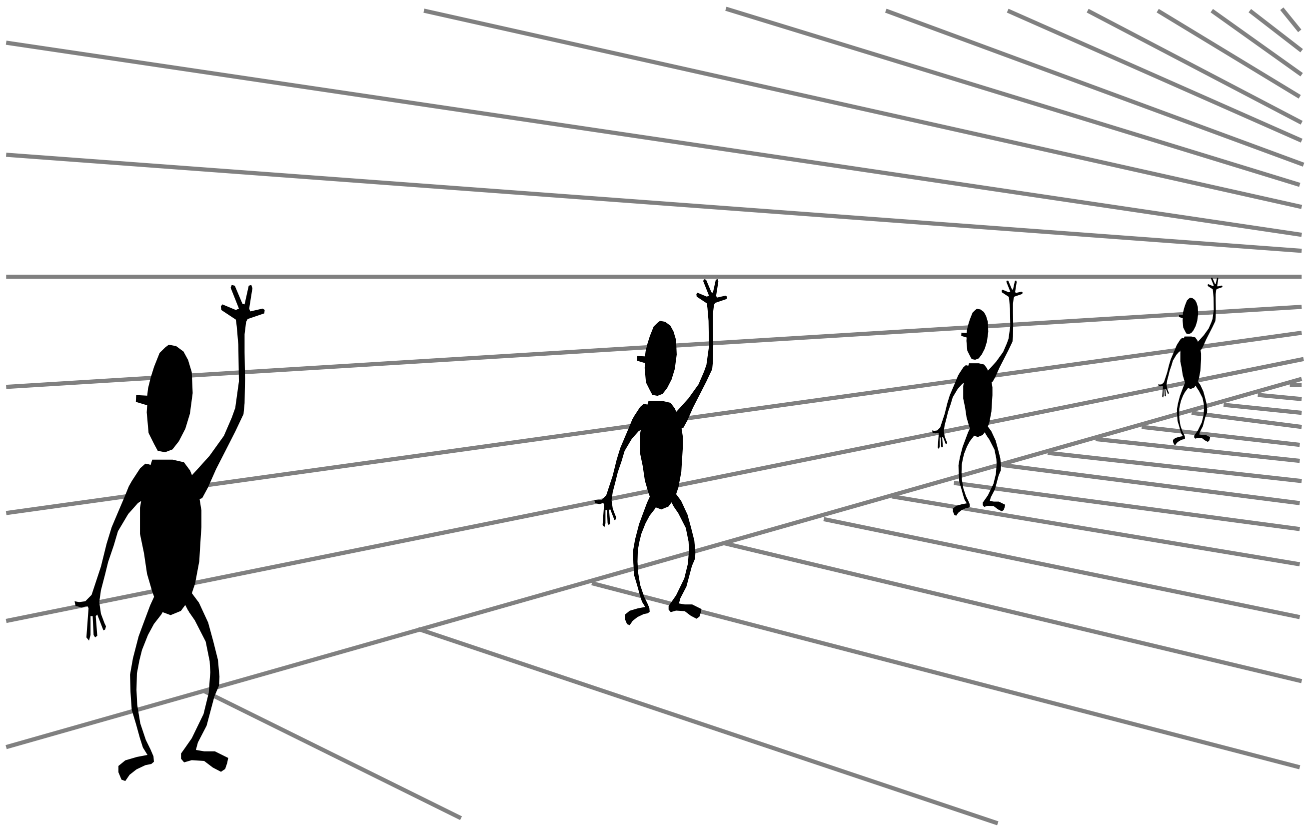
Perspective cues



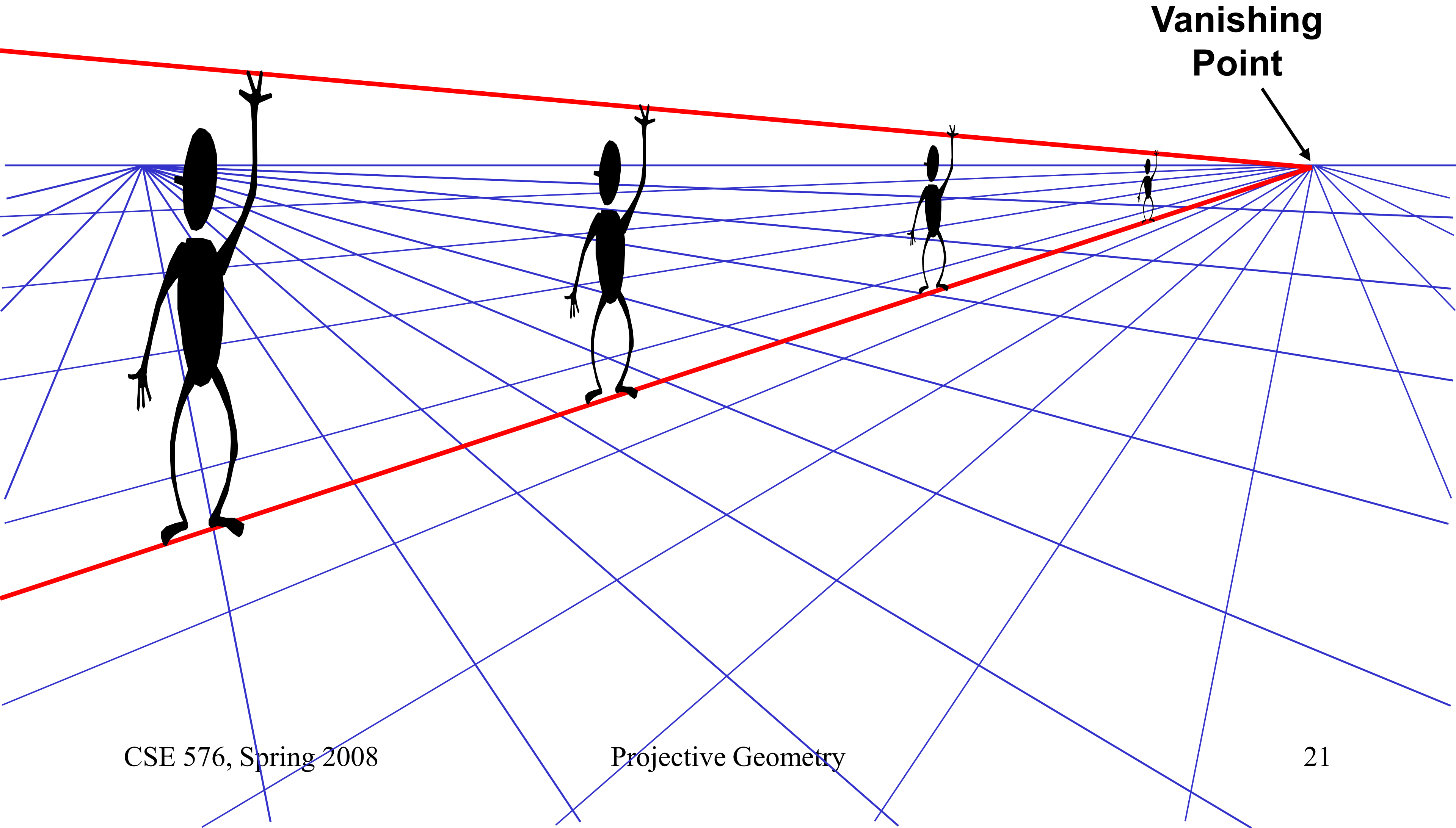
Perspective cues



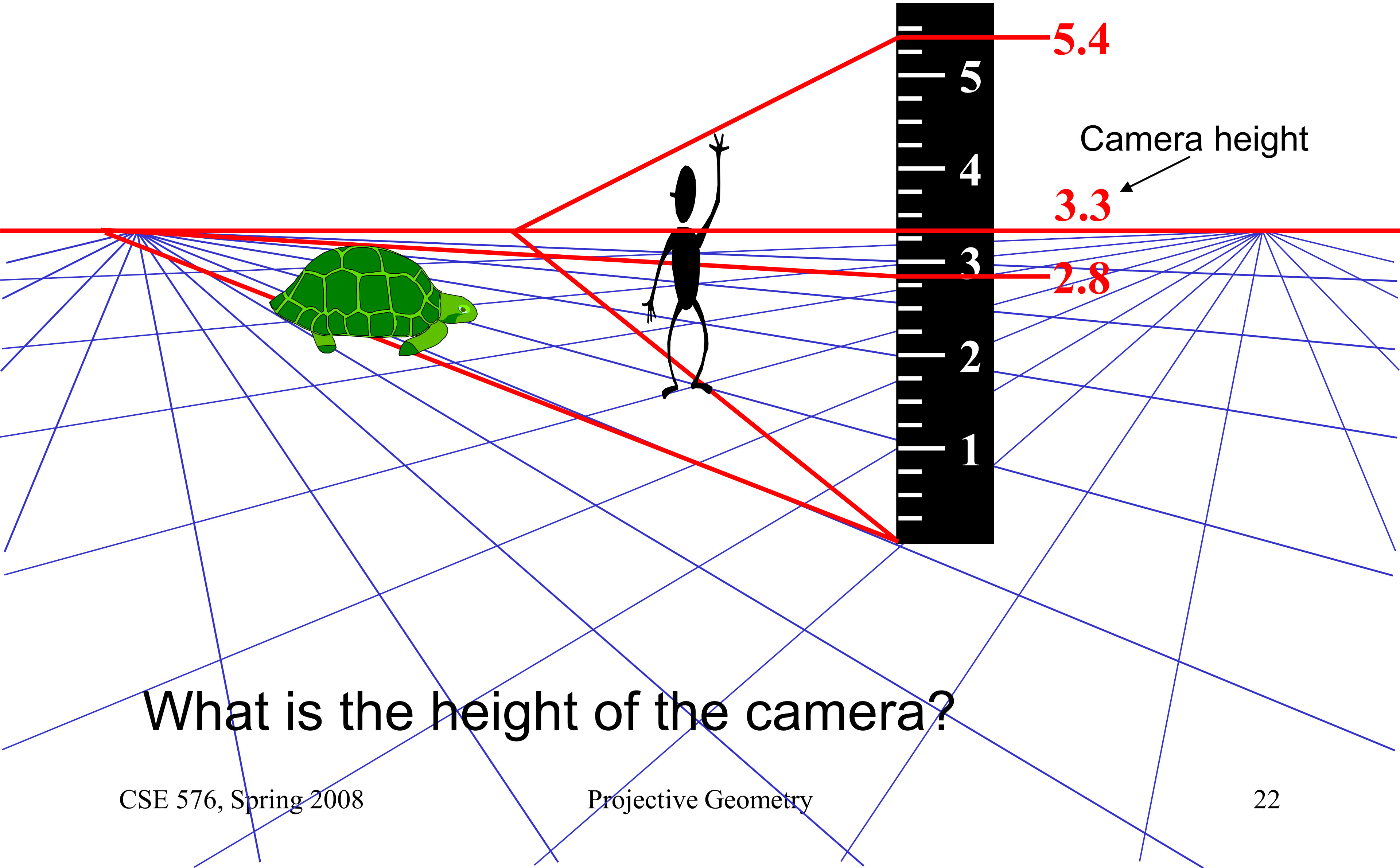
Perspective cues



Comparing heights

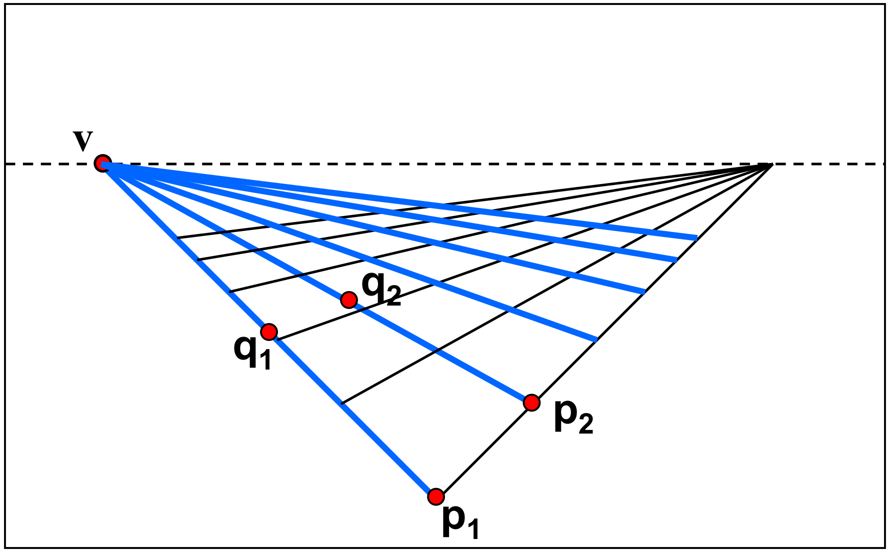


Measuring height



What is the height of the camera?

Computing vanishing points (from lines)



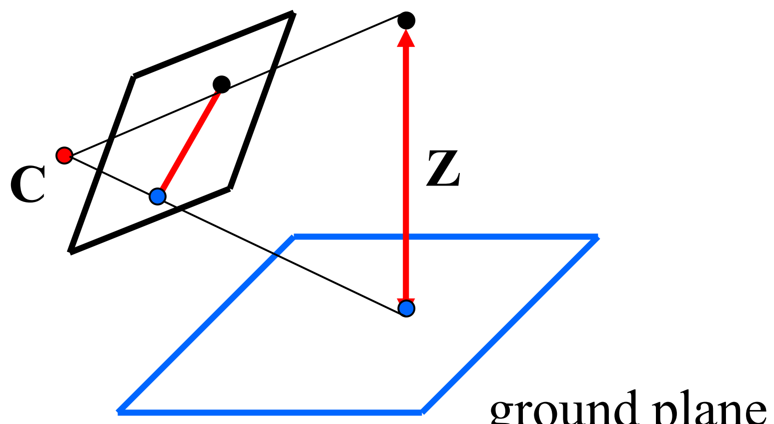
Intersect p_1q_1 with p_2q_2

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Least squares version

- Better to use more than two lines and compute the "closest" point of intersection
- See notes by Bob Collins for one good way of doing this:
  - http://www-2.cs.cmu.edu/~ph/869/www/notes/vanishing.txt

Measuring height without a ruler



Compute Z from image measurements

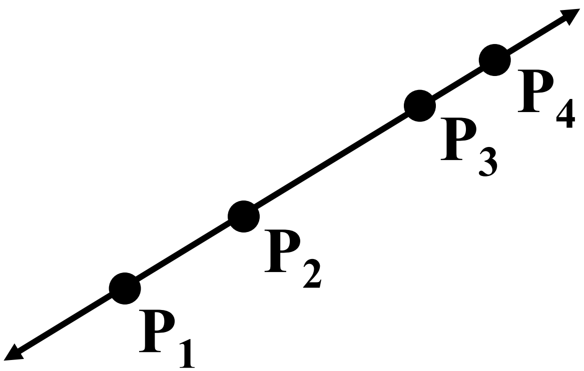
- Need more than vanishing points to do this

The cross ratio

A Projective Invariant

- Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|}$$

$$\mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

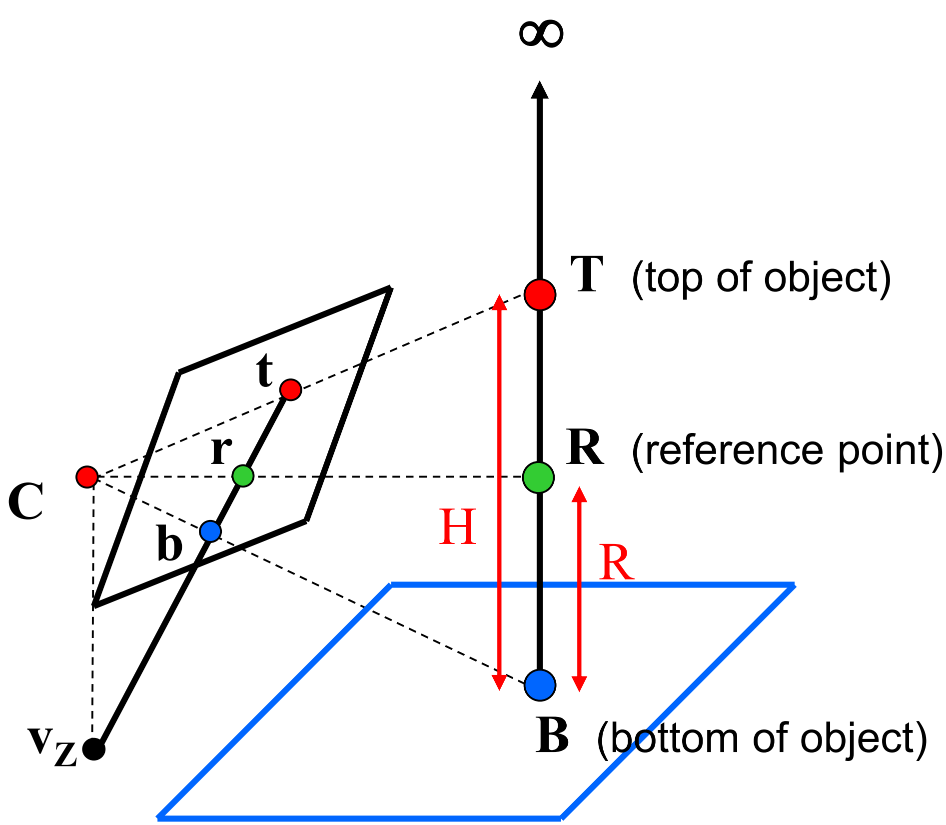
Can permute the point ordering

$$\frac{\|\mathbf{P}_1 - \mathbf{P}_3\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_1 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_3\|}$$

- 4! = 24 different orders (but only 6 distinct values)

This is the fundamental invariant of projective geometry

Measuring height



$$\frac{\|\mathbf{T}-\mathbf{B}\| \; \|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\| \; \|\infty-\mathbf{T}\|} = \frac{H}{R}$$

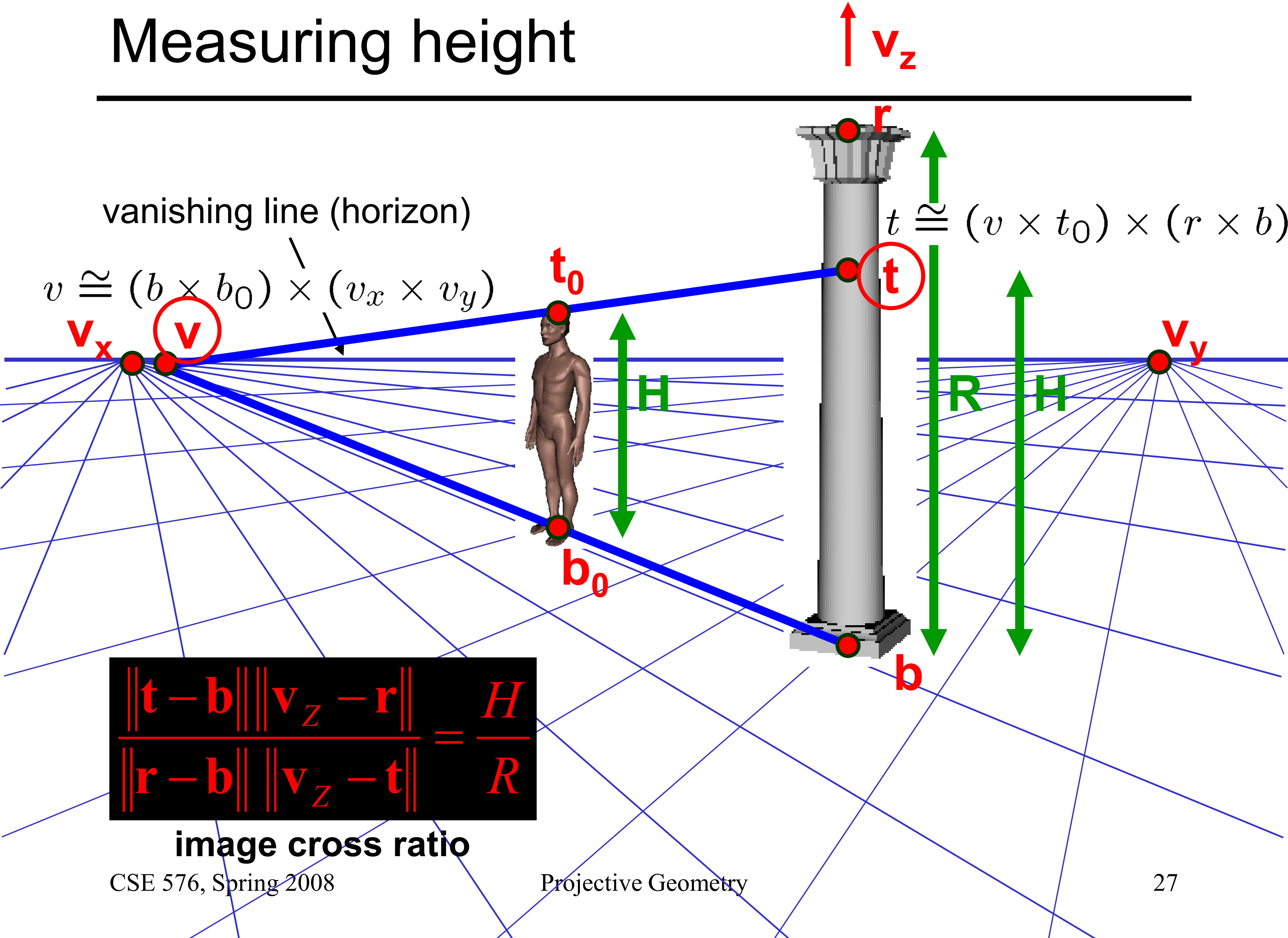
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\| \|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\| \; \|\mathbf{v}_Z-\mathbf{t}\|} = \frac{H}{R}$$

image cross ratio

scene points represented as $\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ image points as $\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

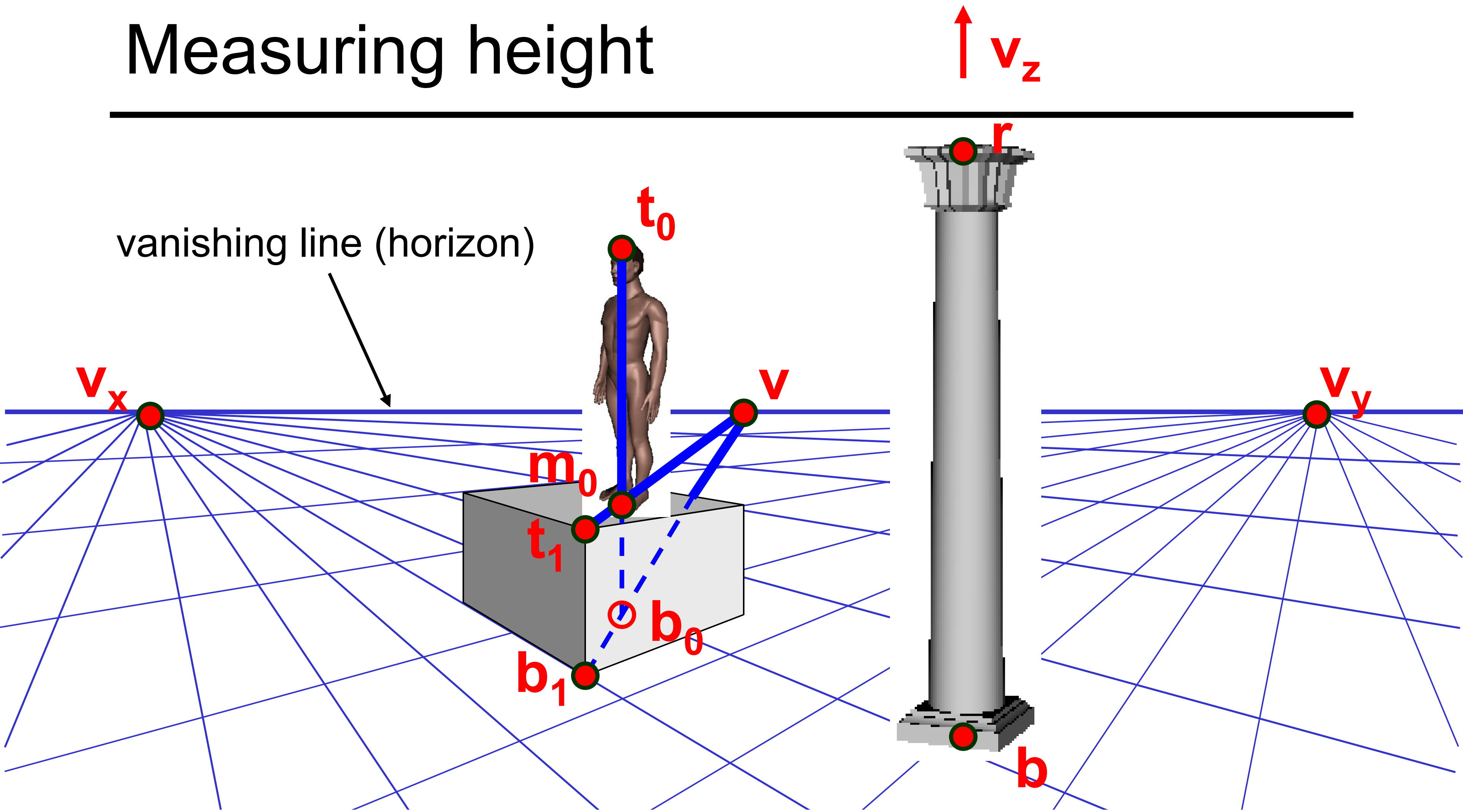
Measuring height



$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

image cross ratio

Measuring height



What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box, height of box is known
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

Lecture 4

Single View Metrology

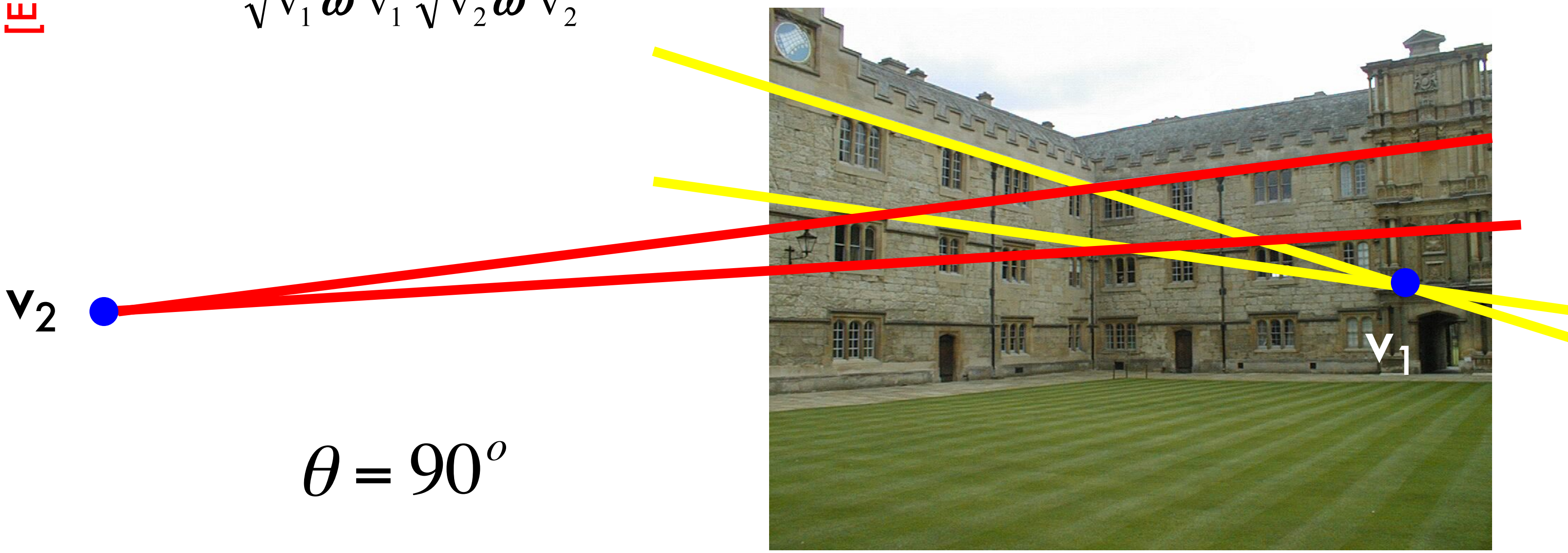
- Review calibration
- Vanishing points and line
- Estimating geometry from a single image
- Extensions

Reading:
[HZ] Chapter 2 “Projective Geometry and Transformation in 3D”
[HZ] Chapter 3 “Projective Geometry and Transformation in 3D”
[HZ] Chapter 8 “More Single View Geometry”
[Hoeim & Savarese] Chapter 2

Single view calibration - example

[Eq. 28]

$$\cos\theta = \frac{v_1^T \omega\, v_2}{\sqrt{v_1^T \omega\, v_1}\sqrt{v_2^T \omega\, v_2}}$$



$$\theta = 90^o$$

[Eq. 29]

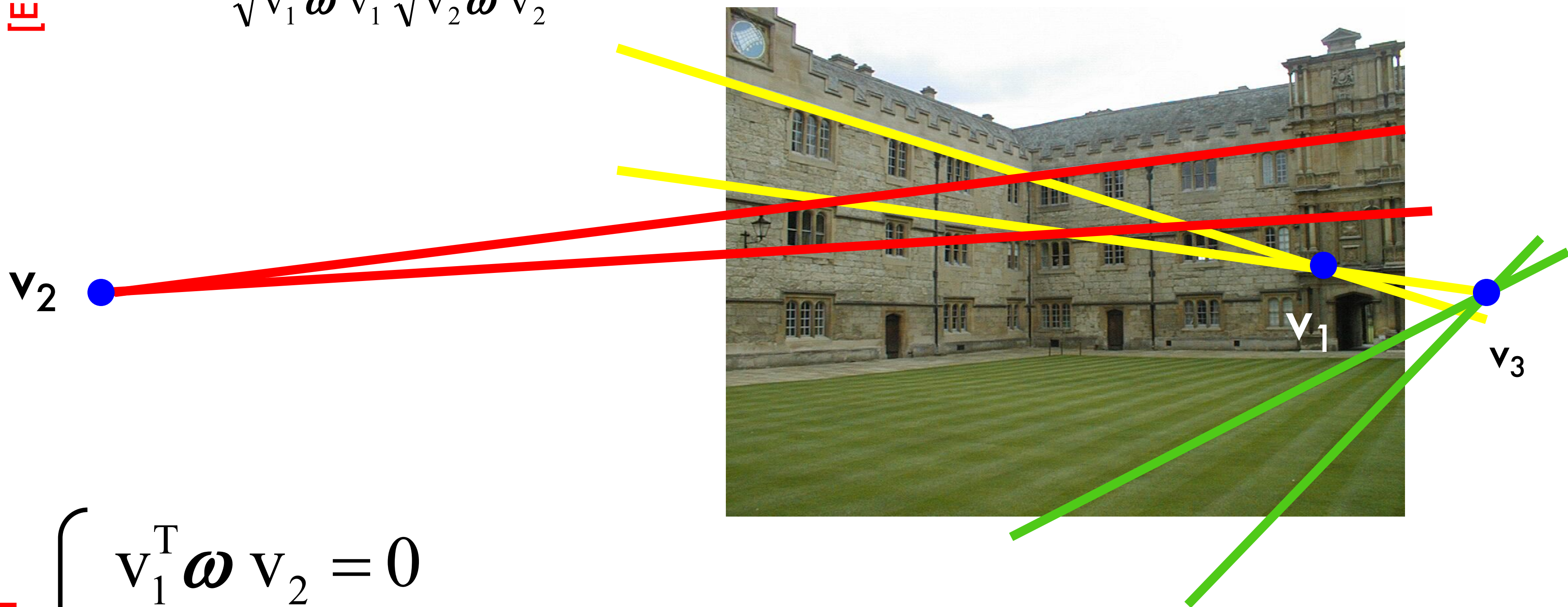
$$\begin{cases} v_1^T \omega\, v_2 = 0 \\ \omega = (K\, K^T)^{-1} \end{cases}$$

→ Do we have enough constraints to estimate K?
K has 5 degrees of freedom and Eq.29 is a scalar equation ☹

Single view calibration - example

[Eq. 28]

$$\cos\theta = \frac{v_1^T \boldsymbol{\omega}\, v_2}{\sqrt{v_1^T \boldsymbol{\omega}\, v_1}\sqrt{v_2^T \boldsymbol{\omega}\, v_2}}$$

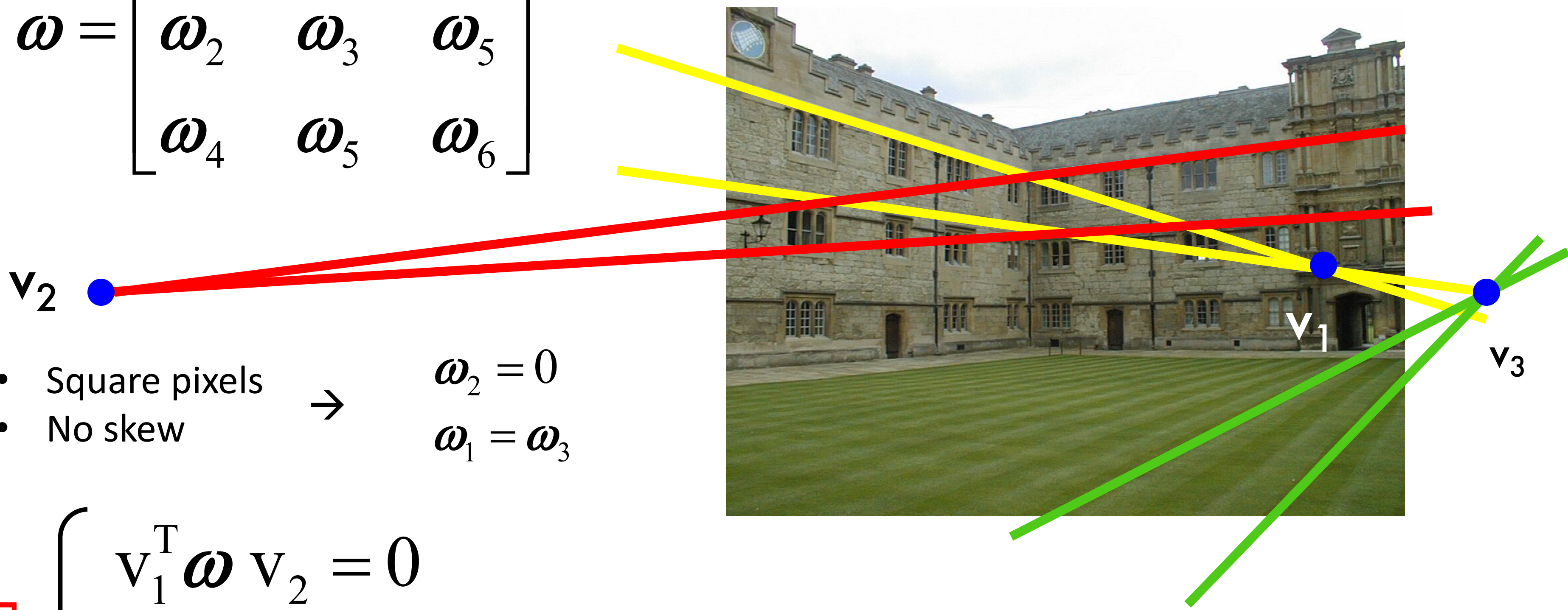


[Eqs. 31]

$$\begin{cases} v_1^T \boldsymbol{\omega}\, v_2 = 0 \\ v_1^T \boldsymbol{\omega}\, v_3 = 0 \\ v_2^T \boldsymbol{\omega}\, v_3 = 0 \end{cases}$$

Single view calibration - example

$$
\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_4 \\ \omega_2 & \omega_3 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}
$$



- Square pixels
- No skew

→ $\omega_2 = 0$, $\omega_1 = \omega_3$

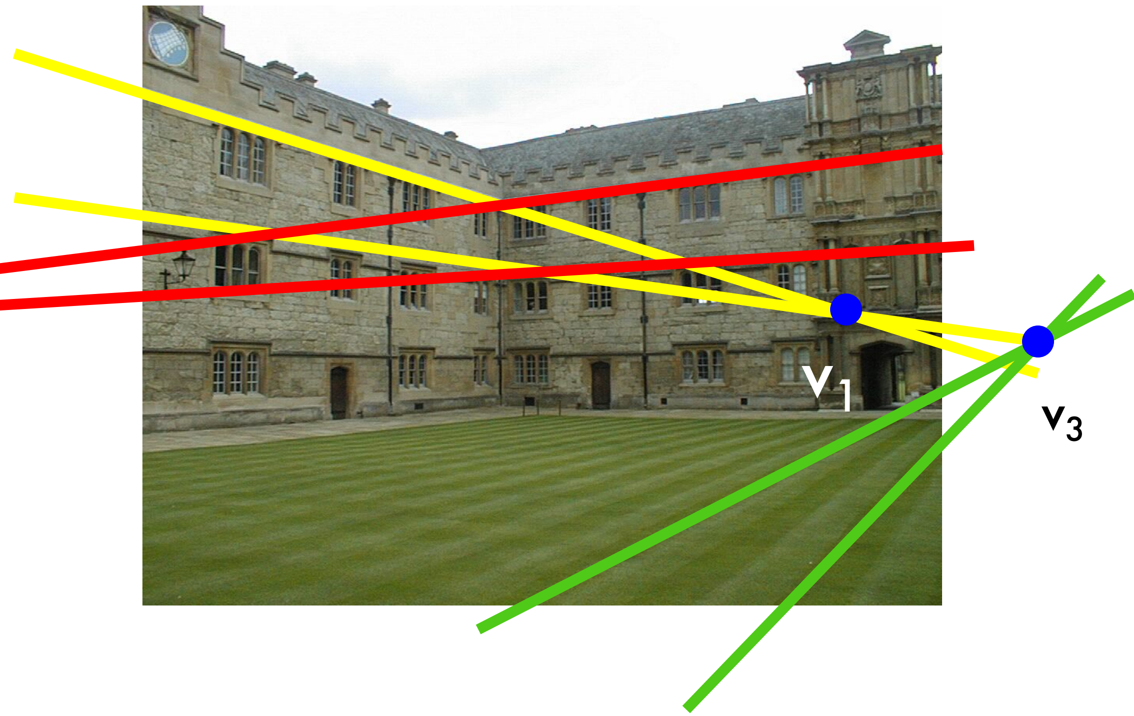
[Eqs. 31]

$$
\begin{cases} v_1^T \omega \, v_2 = 0 \\ v_1^T \omega \, v_3 = 0 \\ v_2^T \omega \, v_3 = 0 \end{cases}
$$

Single view calibration - example

$$\omega = \begin{bmatrix} \omega_1 & 0 & \omega_4 \\ 0 & \omega_1 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}$$

known up to scale



- Square pixels
- No skew

→ $\omega_2 = 0$, $\omega_1 = \omega_3$

[Eqs. 31]

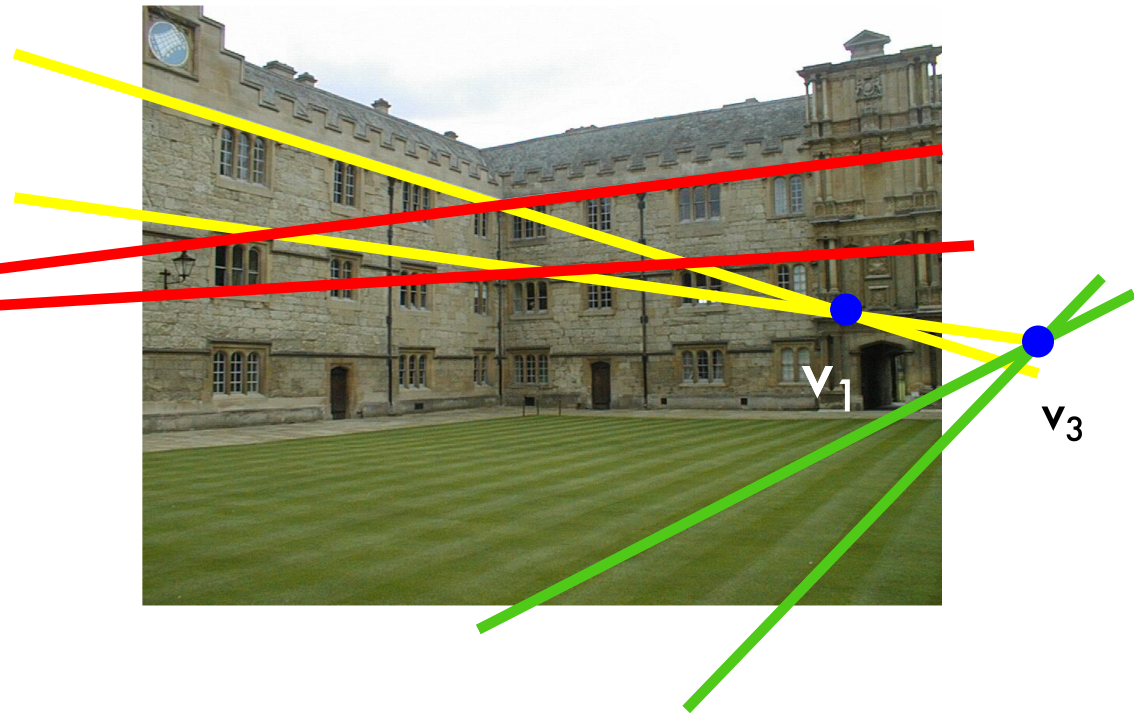
$$\begin{cases} \mathrm{v}_1^T \boldsymbol{\omega}\, \mathrm{v}_2 = 0 \\ \mathrm{v}_1^T \boldsymbol{\omega}\, \mathrm{v}_3 = 0 \\ \mathrm{v}_2^T \boldsymbol{\omega}\, \mathrm{v}_3 = 0 \end{cases}$$

→ Compute ω !

Single view calibration - example

$$\omega = \begin{bmatrix} \omega_1 & 0 & \omega_4 \\ 0 & \omega_1 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}$$

v_2



- Square pixels
- No skew

$\rightarrow$ $\omega_2 = 0$, $\omega_1 = \omega_3$

[Eqs. 31]

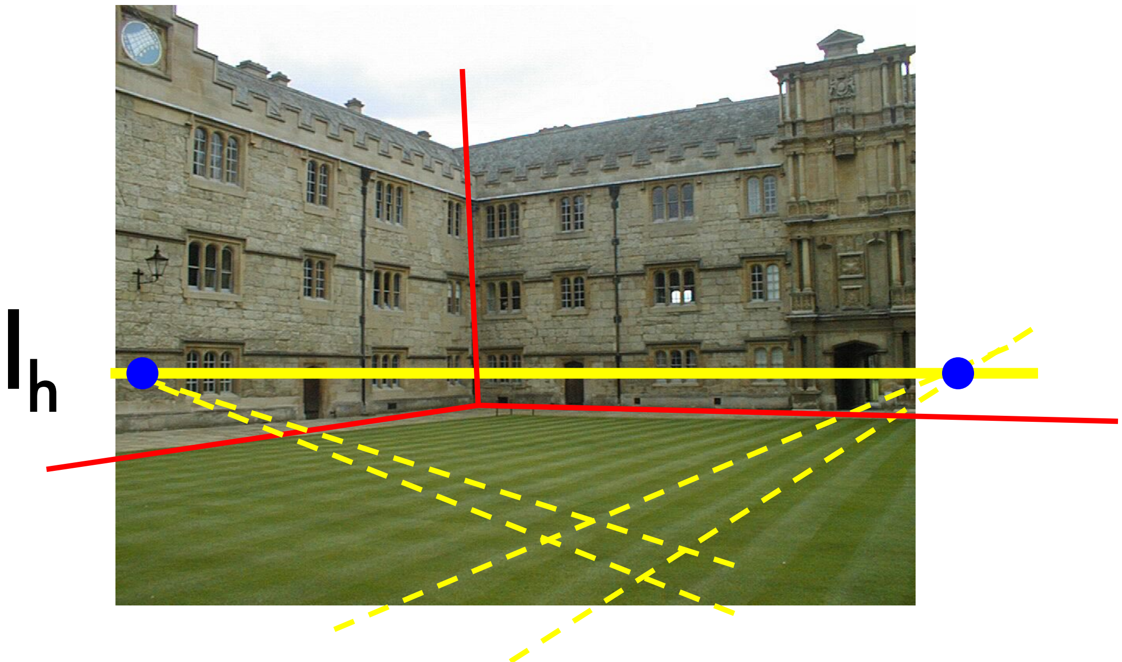
$$\begin{cases} v_1^T \omega \, v_2 = 0 \\ v_1^T \omega \, v_3 = 0 \\ v_2^T \omega \, v_3 = 0 \end{cases}$$

Once ω is calculated, we get K:

$$\omega = (K\,K^T)^{-1} \rightarrow K$$

(Cholesky factorization; HZ pag 582)

Single view reconstruction - example



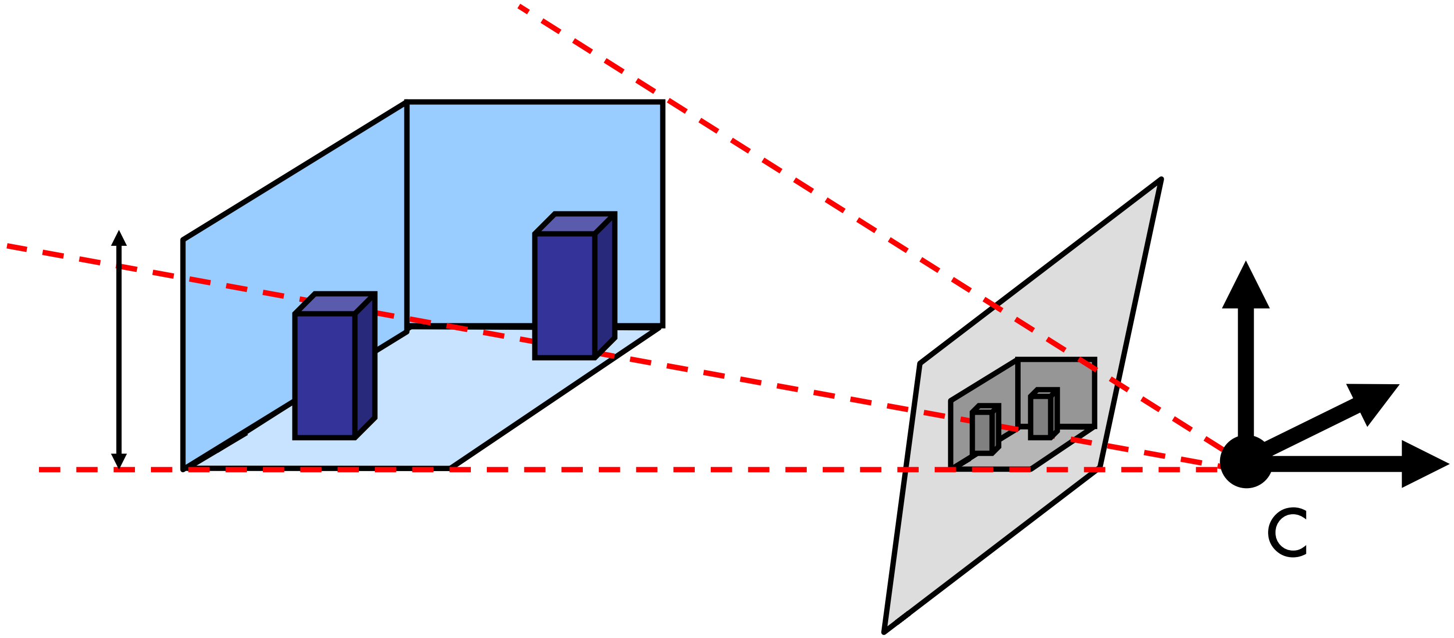
[Eq. 27]

$$K \text{ known} \rightarrow \quad \mathbf{n} = K^T \mathbf{l}_{horiz}$$

= Scene plane orientation in the camera reference system

Select orientation discontinuities

Single view reconstruction - example



Recover the structure within the camera reference system

Notice: the actual scale of the scene is NOT recovered

- Recognition helps reconstruction!
- Humans have learnt this

Lecture 4

Single View Metrology

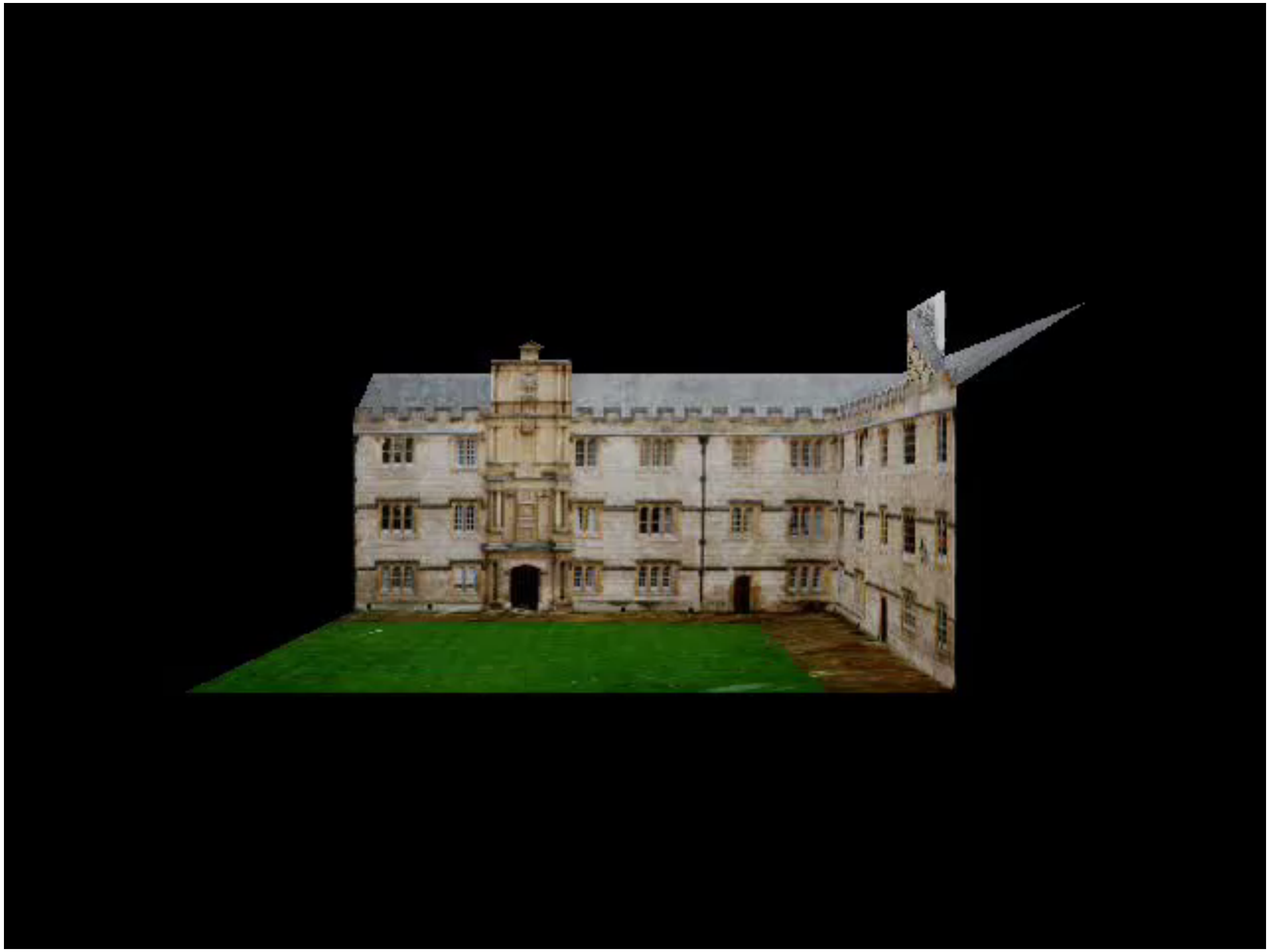
- Review calibration
- Vanishing points and lines
- Estimating geometry from a single image
- Extensions

Reading:
[HZ] Chapter 2 "Projective Geometry and Transformation in 3D"
[HZ] Chapter 3 "Projective Geometry and Transformation in 3D"
[HZ] Chapter 8 "More Single View Geometry"
[Hoeim & Savarese] Chapter 2

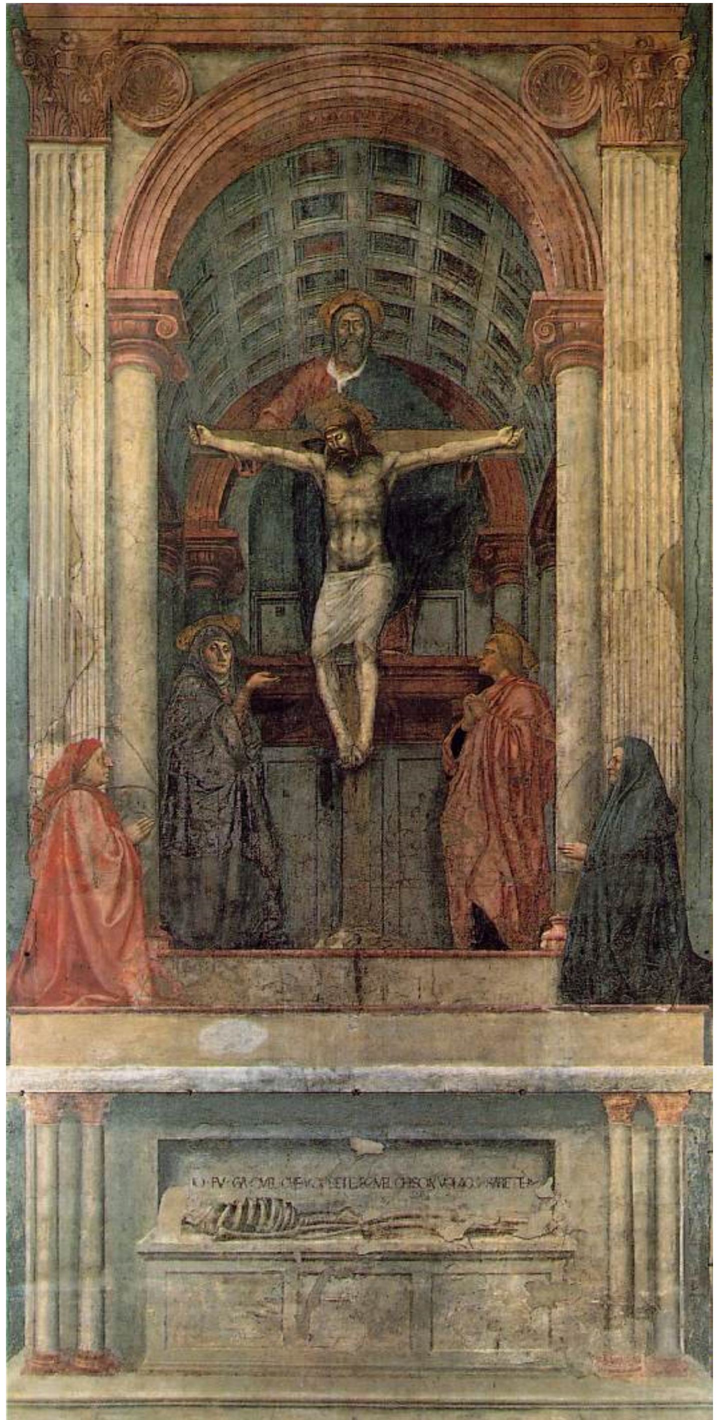
Criminisi & Zisserman, 99

http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/merton/merton.wrl

Criminisi & Zisserman, 99



http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/merton/merton.wrl

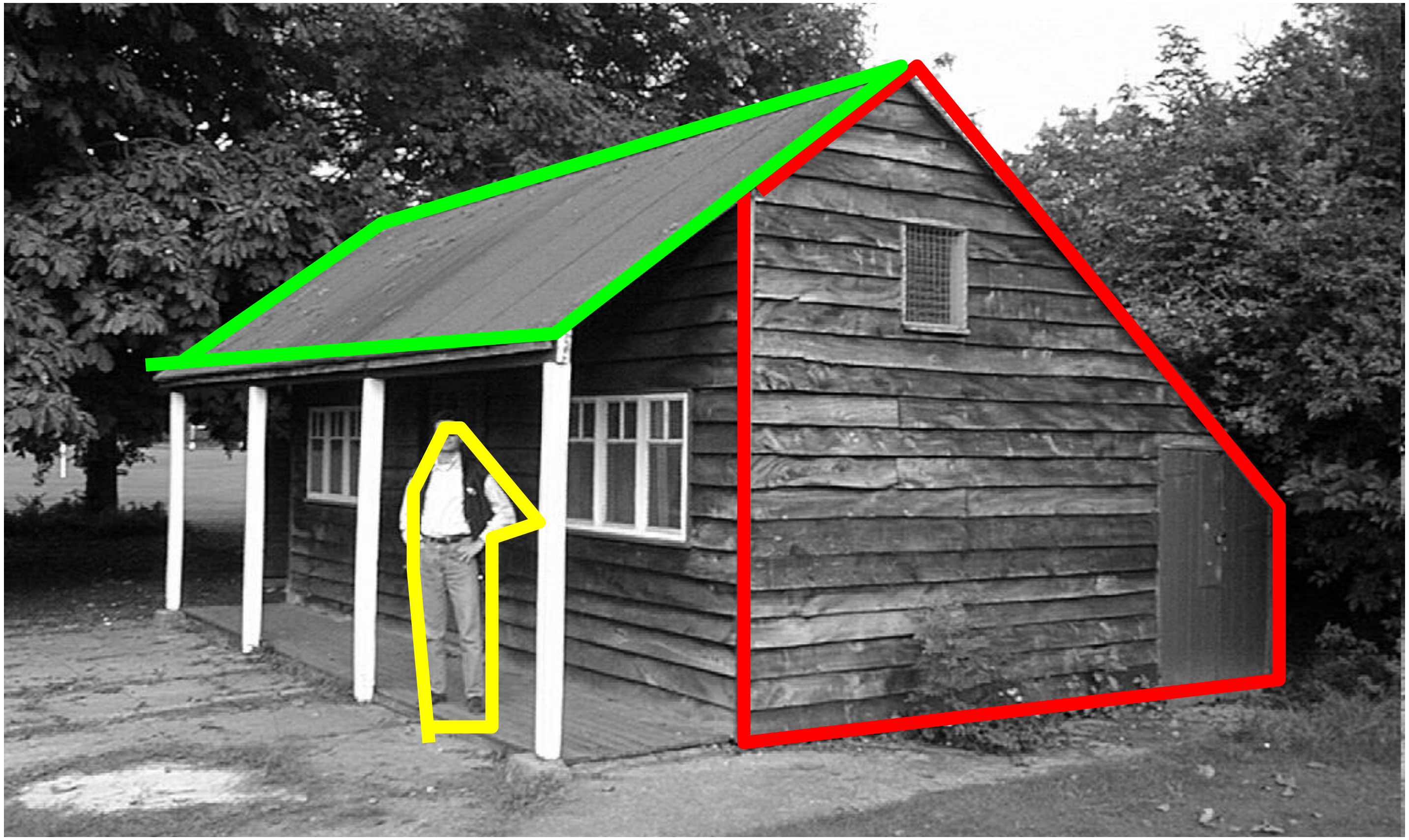


La Trinita' (1426)
Firenze, Santa Maria
Novella; by Masaccio
(1401-1428)

La Trinita' (1426)
Firenze, Santa Maria
Novella; by Masaccio
(1401-1428)

http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/hut/hutme.wrl

Single view reconstruction - drawbacks



Manually select:

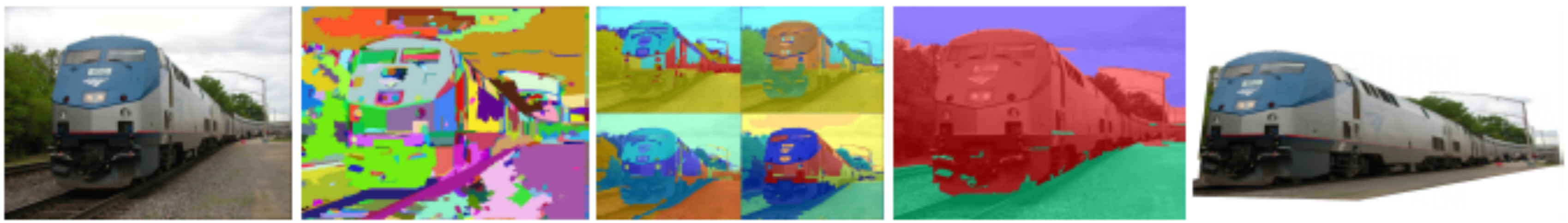
- Vanishing points and lines;
- Planar surfaces;
- Occluding boundaries;
- Etc..

Automatic Photo Pop-up

Hoiem et al, 05

Automatic Photo Pop-up

Hoiem et al, 05…



Automatic Photo Pop-up

Hoiem et al, 05…

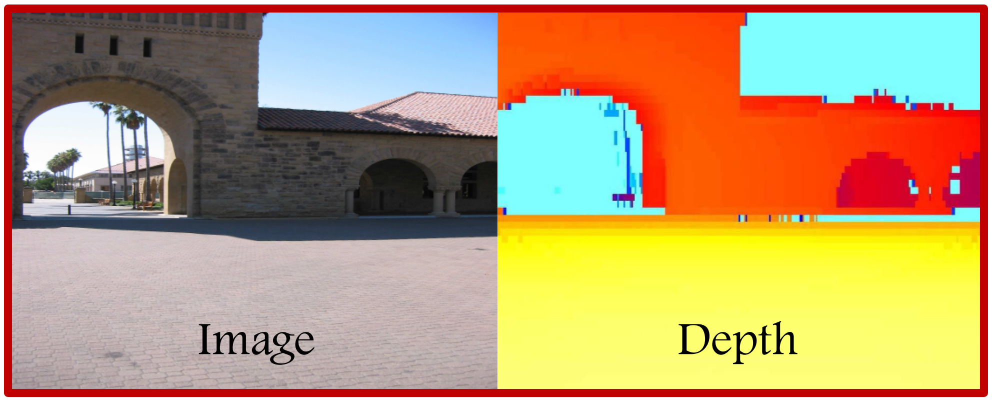
Software:

http://www.cs.uiuc.edu/homes/dhoiem/projects/software.html

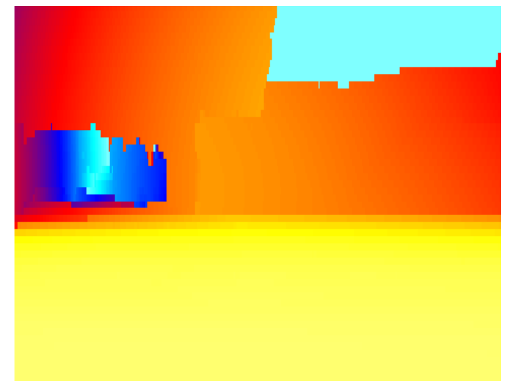
Make3D

Saxena, Sun, Ng, 05…

Training

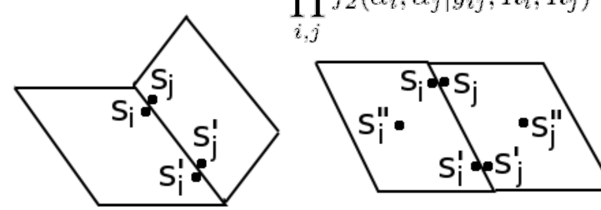


Prediction



Plane Parameter MRF

$$P(\alpha|X,\nu,y,R;\theta) = \frac{1}{Z}\prod_i f_1(\alpha_i|X_i,\nu_i,R_i;\theta) \prod_{i,j} f_2(\alpha_i,\alpha_j|y_{ij},R_i,R_j)$$



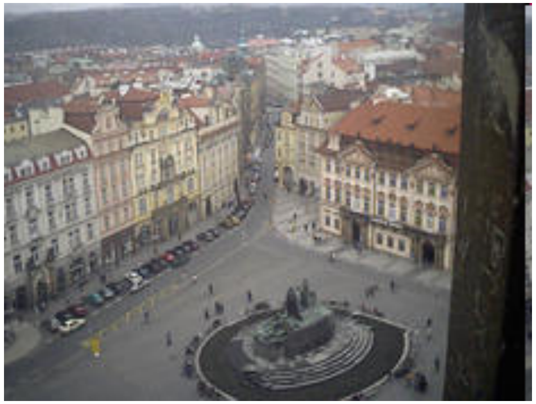
(a) Connectivity

(b) Co-Planarity

youtube

Make3D

Saxena, Sun, Ng, 05...



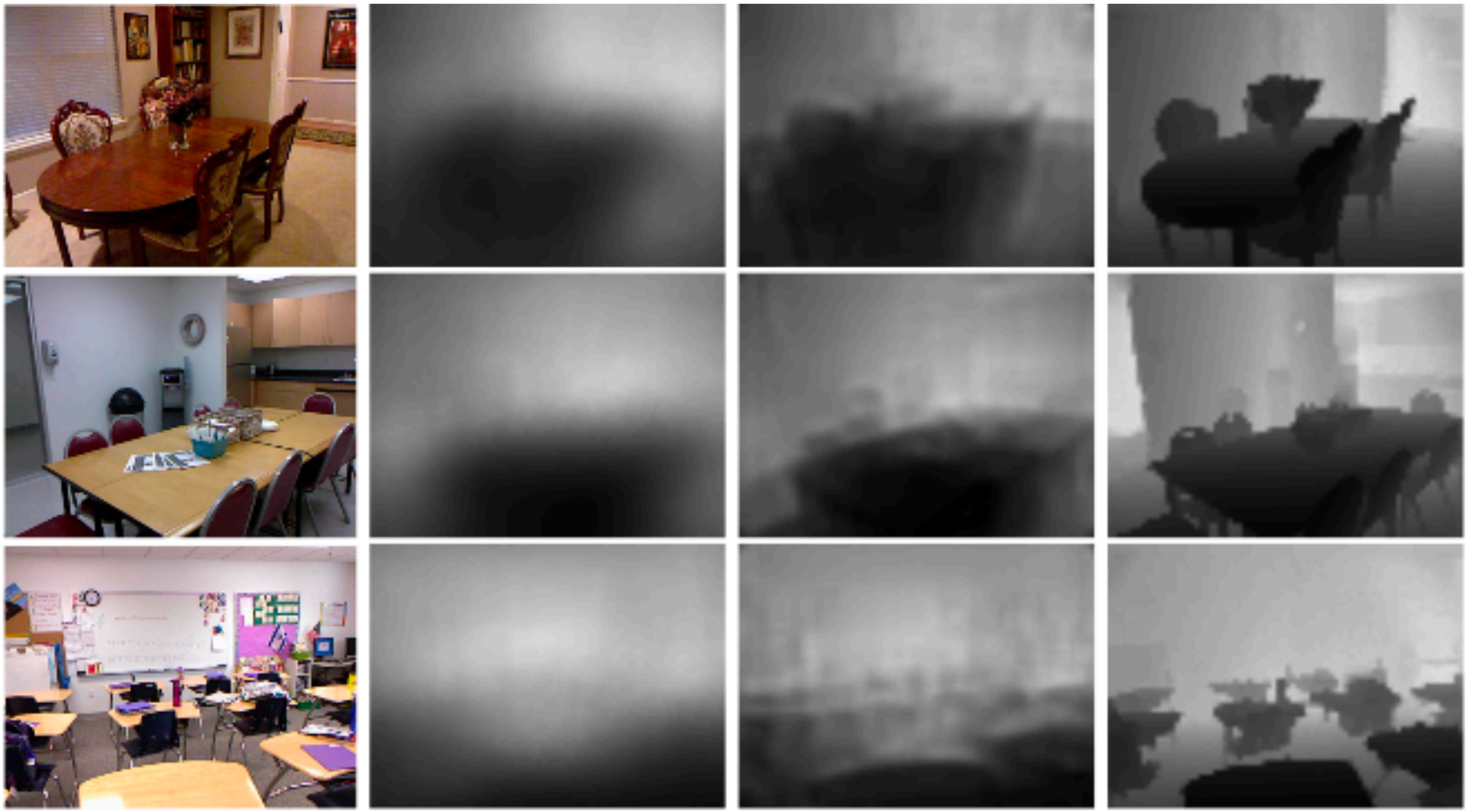
A software: **Make3D**
"Convert your image into 3d model"

http://make3d.stanford.edu/

http://make3d.cs.cornell.edu/

Depth map reconstruction using deep learning

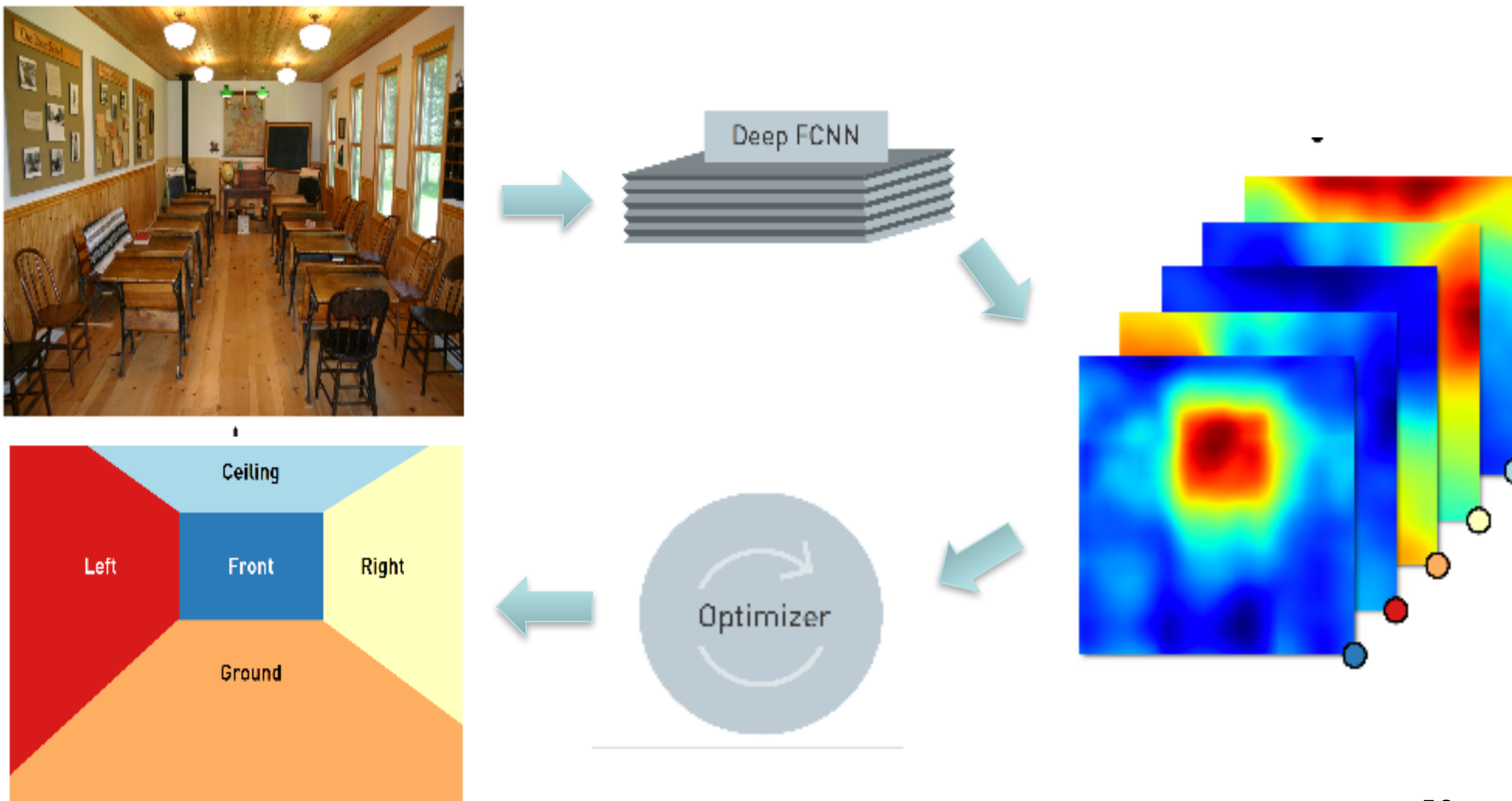
Eigen et al., 2014



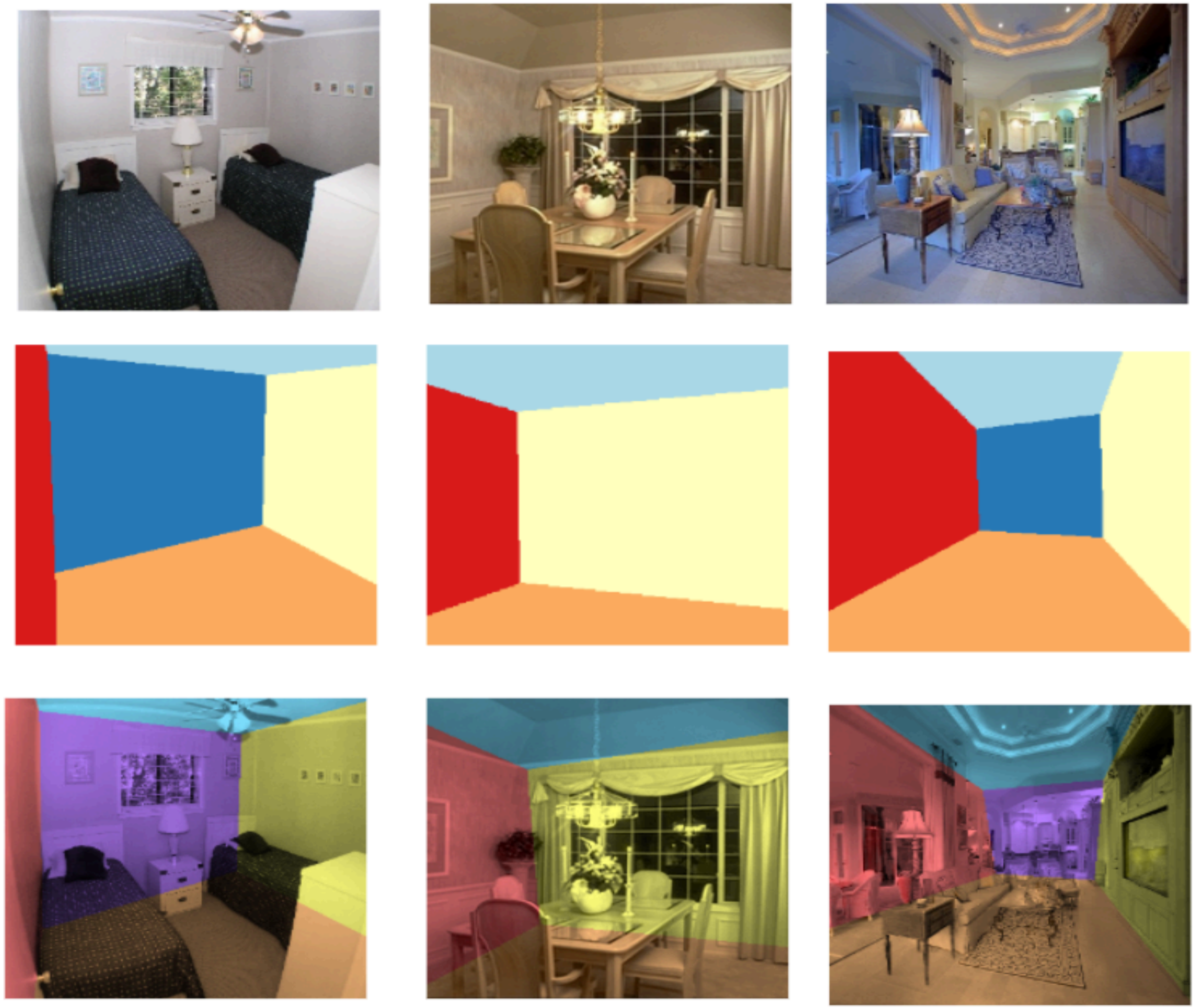
Depth Map Prediction from a Single Image using a Multi-Scale Deep Network,
Eigen, D., Puhrsch, C. and Fergus, R. Proc. Neural Information Processing Systems 2014,

3D Layout estimation

Dasgupta, et al. CVPR 2016

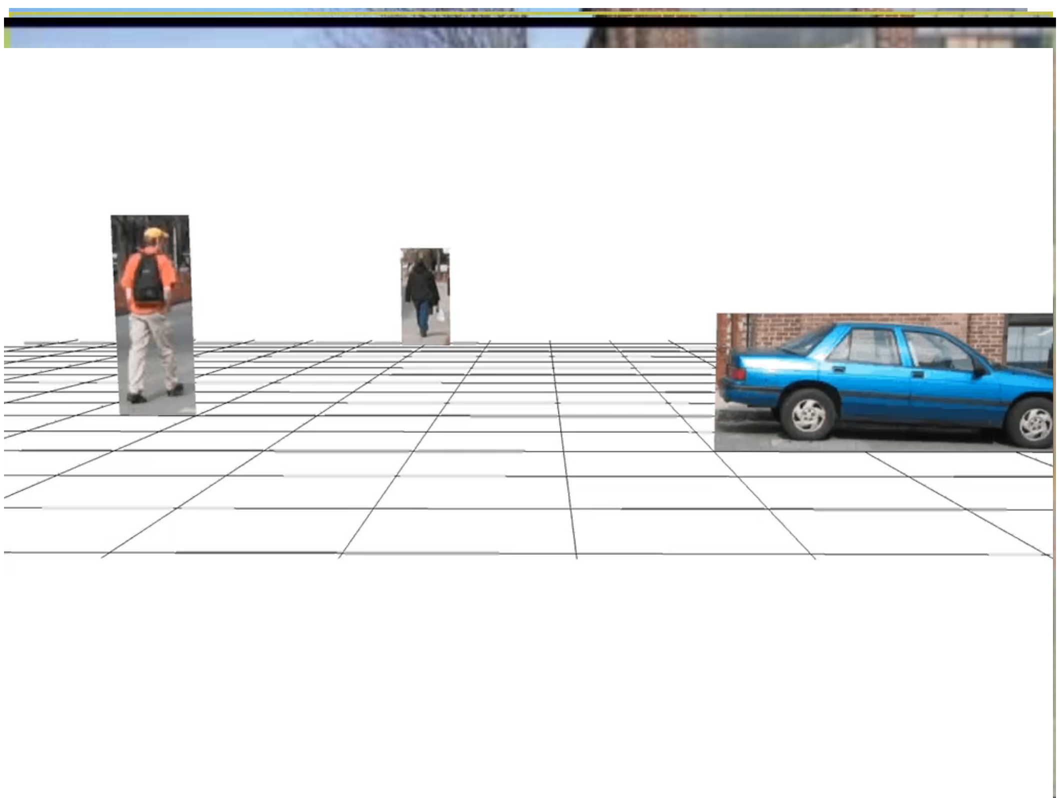


3D Layout estimation



Coherent object detection and scene layout estimation from a single image

Y. Bao, M. Sun, S. Savarese, CVPR 2010, BMVC 2010



Next lecture:

Multi-view geometry (epipolar geometry)

Appendix

Vanishing points - example

v1, v2: measurements
K = known and constant

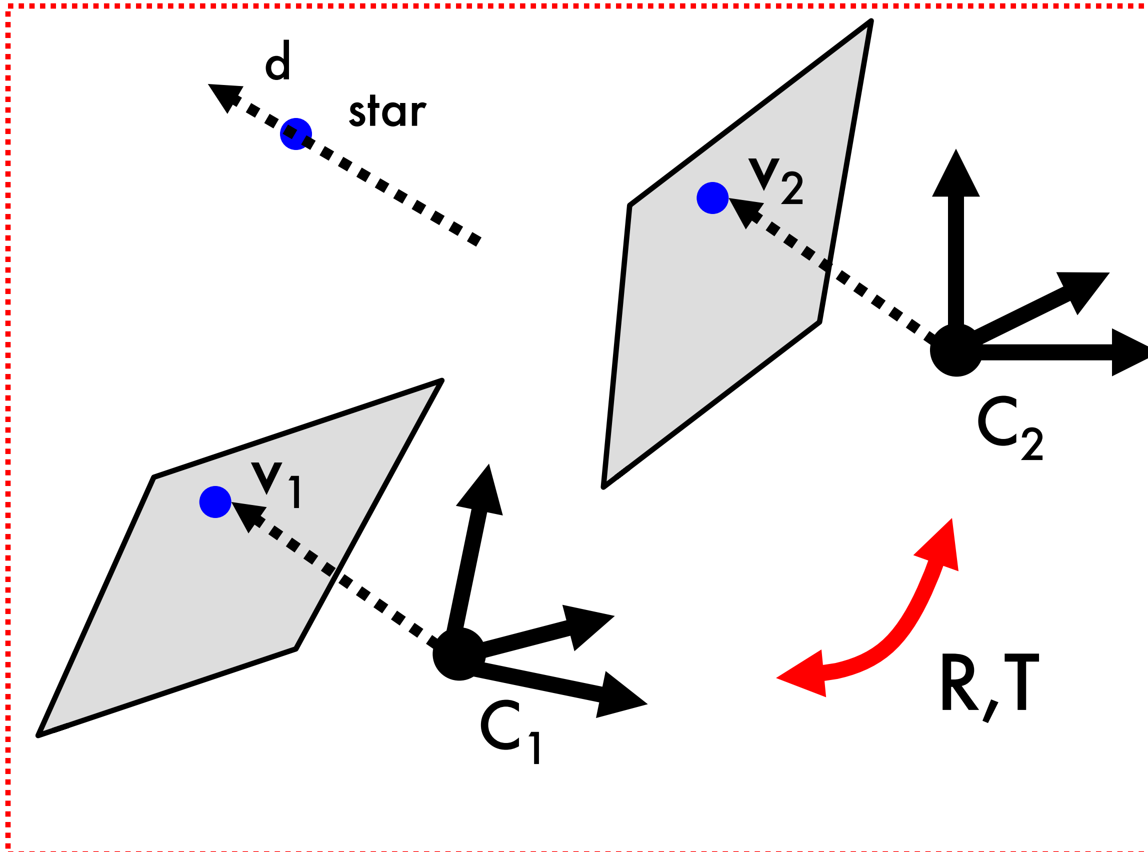
Can I compute R?

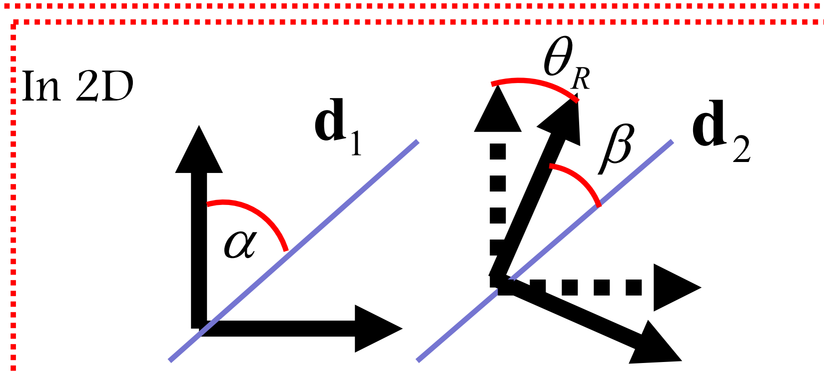
No rotation around z

$$\mathbf{d}_1 = \frac{K^{-1}\,\mathbf{v}_1}{\left\| K^{-1}\,\mathbf{v}_1 \right\|}$$

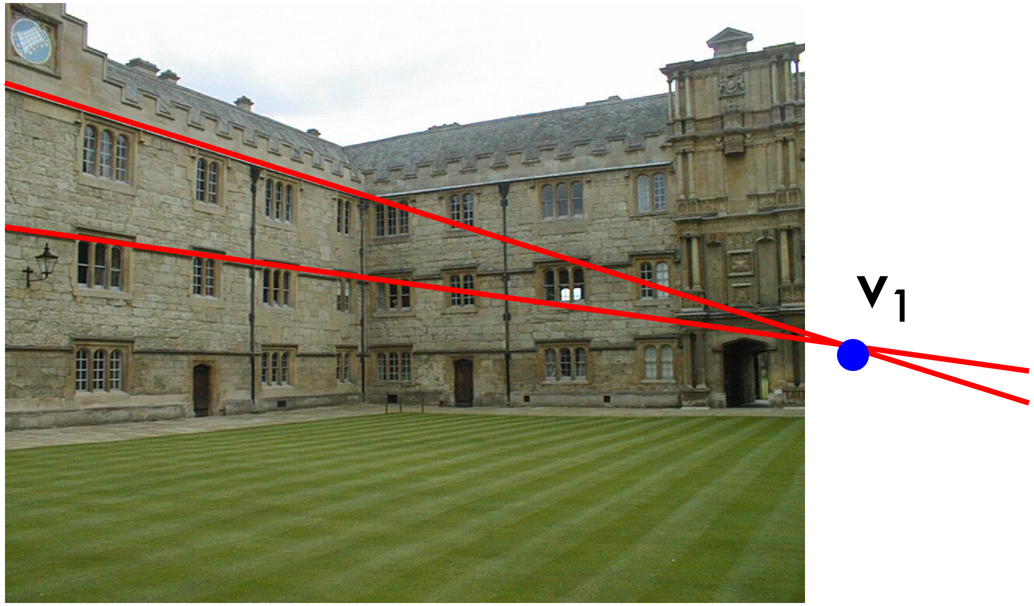
$$\mathbf{d}_2 = \frac{K^{-1}\,\mathbf{v}_2}{\left\| K^{-1}\,\mathbf{v}_2 \right\|}$$

$$R\,\mathbf{d}_1 = \mathbf{d}_2 \longrightarrow R$$





$$\theta_R = \alpha - \beta$$



$$\mathbf{d}_1 = \frac{K^{-1}\,\mathbf{v}_1}{\left\| K^{-1}\,\mathbf{v}_1 \right\|}$$

$$\mathbf{d}_2 = \frac{K^{-1}\,\mathbf{v}_2}{\left\| K^{-1}\,\mathbf{v}_2 \right\|}$$

$$\longrightarrow \quad R$$