Lecture 4
Single View Metrology

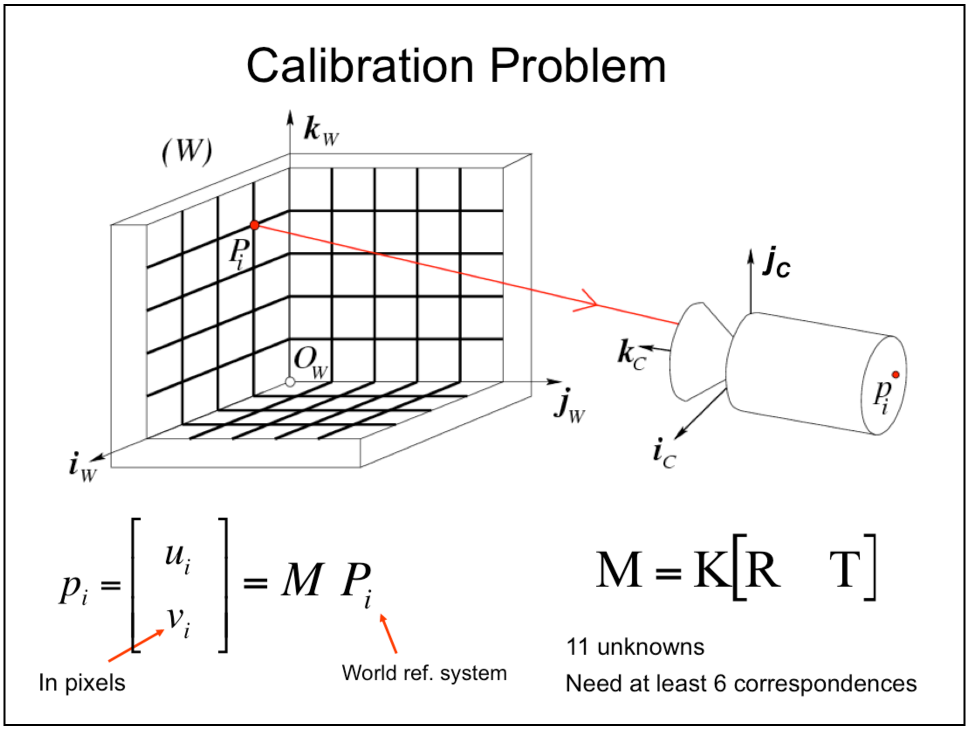
Professor Silvio Savarese
Computational Vision and Geometry Lab

Lecture 4

Single View Metrology

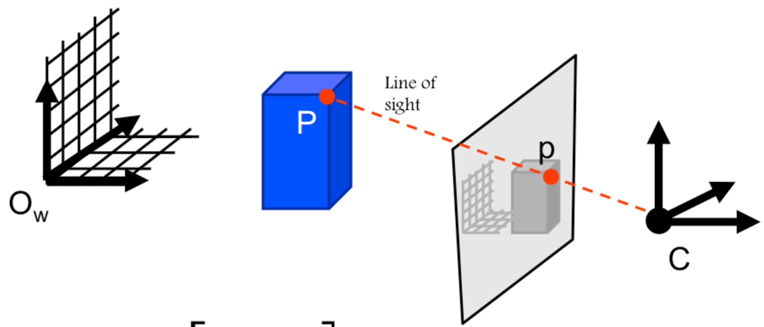
- Review calibration and 2D transformations
- Vanishing points and lines
- Estimating geometry from a single image
- Extensions

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



The calibration problem was discussed in details during lecture 3.

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)

Once the camera is calibrated (intrinsics are known) and the transformation from the world reference system to the camera reference system (which accounts for the extrinsics) is also known, can we estimate the location of a point P in 3D from its (known) observation p? The general answer to this question is no. This is because, given the observation p, even when the camera intrinsics and extrinsic are known, the only thing we can say is that the point P is located somewhere along the line defined by C and p. This line is called the **line of sight.** The actual location of P along this line is unknown and cannot be determined from the observation p alone.

Recovering structure from a single view

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

In the remainder of this lecture, we will introduce tools and techniques for inferring the geometry of the camera and the 3D environment from a just one image of such environment.

Transformation in 2D

-Isometries

-Similarities

-Affinity

-Projective

Before we go into the details of that, let me recap some of important concepts related to geometric transformations.

There are four important classes of transformations in 2D. All these will be described in homogenous coordinates.

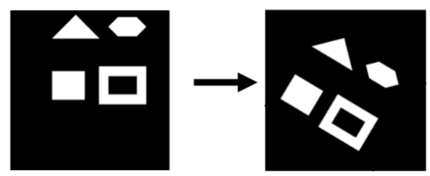
Transformation in 2D

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



The first one is called the **isometric transformation** which is in general the concatenation of a rotation and translation transformations and expressed by the matrix H_e and Eq. 4. This transformation preserves the distance between any pair of points and has 3 degrees of freedom (2 for translation, 1 for rotation). This transformation captures the motion of a rigid object.

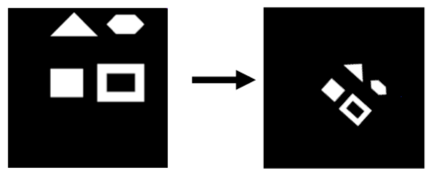
Transformation in 2D

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



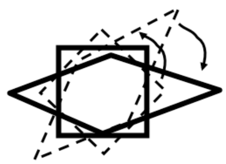
The second group of transformations is called **the similarity transformations**. The transformation is a concatenation of a translation, rotation and uniformscale transformation. It preserves the ratio of lengths between any two line segments before and after transformation. It also preserves the angle between any intersecting lines. The group has 4 degrees of freedom (2 translation, 1 rotation, 1 scale)

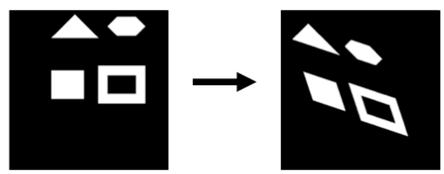
Transformation in 2D

Affinities:
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{[Eq. 6]}$$

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

[Eq. 7]





The third transformation group is the affine transformation. This group can be interpreted as a series of translation, rotation and anisotropic scaling transformations. The core of the affine transformation is characterized by the matrix A which includes a rotation by phi, an anisotropic scaling by [sx 0; 0 sy], an inverse rotation by –phi and an arbitrary rotation by theta (see Eq. 7). This transformation has also a mathematical interpretation: Any 2x2 matrix can be decomposed into two orthogonal matrices and a diagonal matrix by Singular Value Decomposition (this is by assuming positive scaling; for negative scaling, the transformation may not be unique.) Thus we can express A = $UDV^T = (UV^T)(VDV^T)$ = R(theta) R(-phi) D R(phi)

by replacing UV^T = R(theta), V = R(phi)^T, and where D is a diagonal matrix with the singular values. Note that transpose of a rotation matrix is its inverse. Thus, an arbitrary 2x2 matrix can be decomposed into R(theta), R(phi), D.

Transformation in 2D

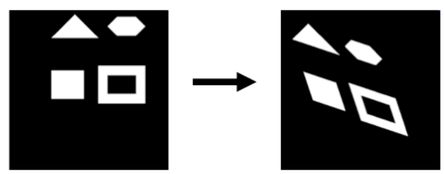
Affinities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{[Eq. 6]}$$

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

[Eq. 7]

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



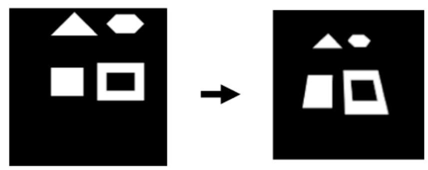
This transformation group preserves parallel lines (parallel lines are still parallel after transformation). The ratio of lengths on collinear lines is preserved after transformation and thus it follows that the ratio of areas within any arbitrary shape is preserved.

The transformation has 6 degrees of freedom (4 elements in the matrix A, 2 for translation).

Transformation in 2D

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

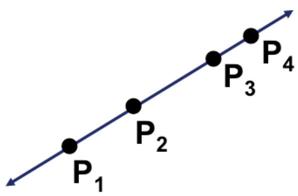
- 8 DOF
- Preserve:
    - cross ratio of 4 collinear points
    - collinearity
    - and a few others…



Finally, the fourth transformation is the projective transformation which is a generalization of the affine transformation where two additional elements (the 1x2 vector v) are non-zero. This transformation has 8 degrees of freedom and preserves the cross ratio of 4 collinear points. Note that the projective matrix is defined up to a scale (hence the degrees of freedom are 8 instead of 9).

The cross ratio

The cross-ratio of 4 collinear points is defined as



[Eq. 9]

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

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

An example of cross ratio is defined in Eq.9 for the points P_1,P_2,P_3,P_4.

Lecture 4

Single View Metrology

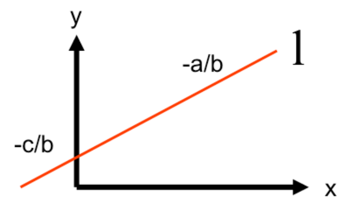
- Review calibration and 2D transformations
- Vanishing points and lines
- Estimating geometry from a single image
- Extensions

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

In the next slides we introduce a number of important definitions about lines and points in 2D and 3D and introduce the concepts of vanishing points and lines.

Lines in a 2D plane

$$ax + by + c = 0$$



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

If $x = [\, x_1, x_2]^T \in l$

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

[Eq. 10]

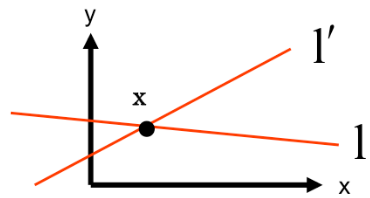
A line in 2D can be represented as the 3D vector $l=[a\ b\ c]^T$ in homogeneous coordinates; The ratio -a/b captures the slope of the line and the ratio -c/b defines the point of intersection of the line with the y axis.

If a point x belongs to a line l, than the dot product between x and l is equal to zero (Eq 10). This equations also define a line in 2D.

Lines in a 2D plane

Intersecting lines

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



Proof

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

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

→ x is the intersecting point

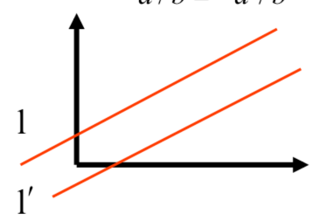
In general, two lines l and l' intersect at a point x. This point is defined as the the cross product between l and l' [Eq. 11]. This is easy to verify as the slide shows.

Proof: Given two intersecting lines l and l', the intersection point x should lie on both lines l and line l'; thus the point x is the intersection if and only if $x^T l = 0$ [Eq.11] and $x^T l' = 0$ [Eq.12]. Let x be l x l'. Then, the vector x is perpendicular to the vector l and the vector l' and, thus, it satisfies the above constraints. Since the intersection is unique (set arbitrary x' =\= x and show that x' is x), l x l' is the point of intersection of the two lines.

2D Points at infinity (ideal points)

$$\mathrm{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \mathrm{x}_3 \end{bmatrix}, \mathrm{x}_3 \neq 0 \qquad \mathrm{l} = \begin{bmatrix} \mathrm{a} \\ \mathrm{b} \\ \mathrm{c} \end{bmatrix}$$

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



$$x_\infty = \begin{bmatrix} x'_1 \\ x'_2 \\ 0 \end{bmatrix} \qquad l' = \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}$$

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

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

Let now us compute the point of intersection of two parallel lines.

We start by observing that a point at infinity in Euclidean coordinates corresponds to a point x_{inf} in homogenous coordinates whose third coordinate is equal to zero.

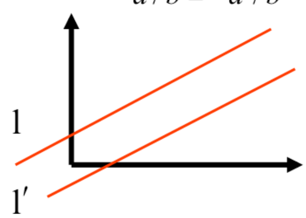
Let us now consider two parallel lines l and l'. When two lines are parallel, their slope is equal and thus –a/b=-a'/b'. Let's now compute the point of intersection of these two lines. How do we do that? By computing the cross product of l and l

we obtain Eq. 13 (you can do this as an exercise) which is exactly the expression of a point at infinity (in homogenous coordinates). This confirms the intuition that two parallel lines intersect at infinity.

The point of intersection of two parallel lines returns a point at infinity which is also called ***ideal point***.

2D Points at infinity (ideal points)

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



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

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

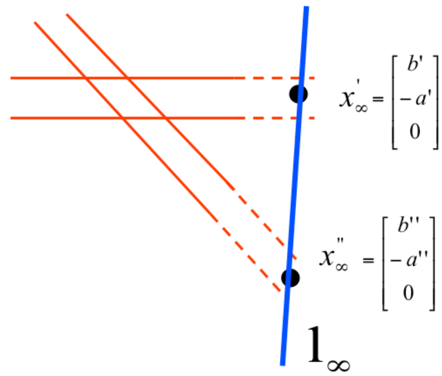
One interesting property of a point at infinity is that all the parallel lines with the same slope -a/b passes through the ideal point [b -a 0] [Eq.15].

Lines infinity $\mathbf{l}_\infty$

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

$$\mathbf{l}_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

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

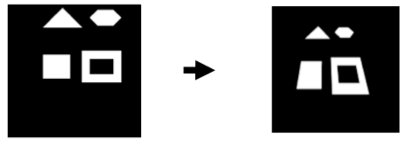
We can further extend this concept and define the **lines at infinity**.

Consider 2 or more pairs of parallel lines (right side of the slide). Each pair of parallel lines intersect into a point at infinity. Let us call these set of points x_{inf}’, x_{inf}’’, …. The line l that passes through all these points at infinity must satisfy $l^T x_{inf}' = 0$, $l^T x_{inf}'' = 0$, etc… and is simply l_inf = $[0\ 0\ c]^T$. Since c in an arbitrary value, we can simply write l_inf=$[0,0,1]^T$.

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

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?

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

…no!

[Eq. 17]

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

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

[Eq. 18]

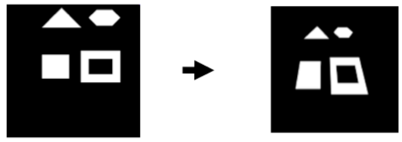
Now, let's see what happens if we apply a generic projective transformation H to a point at infinity p_{inf} (Eq. 17).

Notice that the last element H p_{inf} becomes non-zero which suggests that a projective transformation in general maps points at infinity to points that are no longer at infinity.

This is not true for affine transformations [Eq.18]. If we apply an affine transformation H_A to p_{inf} will still obtain a point at infinity.

Projective transformation of a line (in 2D)

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



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

[Eq. 19]

is it a line at infinity?

$$H^{-T}\, l_\infty = ? = \begin{bmatrix} A & t \\ v & b \end{bmatrix}^{-T} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ b \end{bmatrix}$$

[Eq. 20]

…no!

$$H_A^{-T}\, l_\infty = ? \quad = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}^{-T} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} A^{-T} & 0 \\ -t^T A^{-T} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

[Eq. 21]

This time, let's apply a projective transformation H to a line. The projective transformation of a line l is $l' = H^{-T} l$ (Eq 19).

Let's derive this equation. All points that pass through a line l must satisfy the line equation: $x^T l = 0$; —> $x^T H^T H^{-T} l = 0$

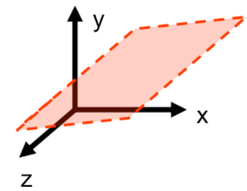
Since $x' = H x$ (and $x'^T = x^T H^T$), then $x'^T H^{-T} l = 0$; Because, after the transformation, a projected point x' must still belong to the projected line l' ($x'^T l' = 0$), it implies that $l' = H^{-T} l$.

Let's now apply the projective transformation H to a line at infinity l_{inf}. Is the projected line still at infinity? No.

Let's now apply the affine transformation H_A to a line at infinity l_{inf}. Is the projected line still at infinity? Yes, as the derivation next to Eq 21 shows.

Points and planes in 3D

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \qquad \Pi = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$



$$x \in \Pi \leftrightarrow x^T \Pi = 0 \quad \text{[Eq. 22]}$$

$$ax + by + cz + d = 0 \quad \text{[Eq. 23]}$$

How about lines in 3D?

- Lines have 4 degrees of freedom - hard to represent in 3D-space
- Can be defined as intersection of 2 planes

So far we have introduced the concepts of lines and points at infinity in 2D. Let's introduce the equivalent concepts in 3D (and in the corresponding homogenous coordinates). Points in 3D (homogenous coordinates) are denoted as x.

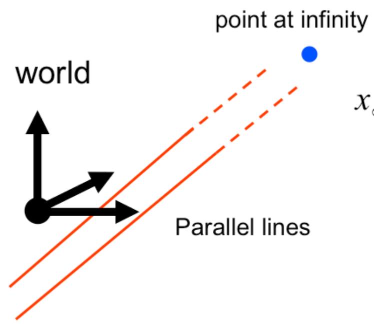
Following the 2D case, we can represent a plane as a normal vector (a,b,c) and a distance from the origin (d) which is [Eq.23]. Thus, a plane is formally defined as the vector PI = [a b c d] in the slide. A point is on a plane if and only if [Eq.22] holds.

How about lines in 3D?

- Lines have 4 degrees of freedom - hard to represent in 3D-space
- Can be defined as intersection of 2 planes. See [HZ] Ch.3.2.2 for detail.

Points at infinity in 3D

Points where parallel lines intersect in 3D

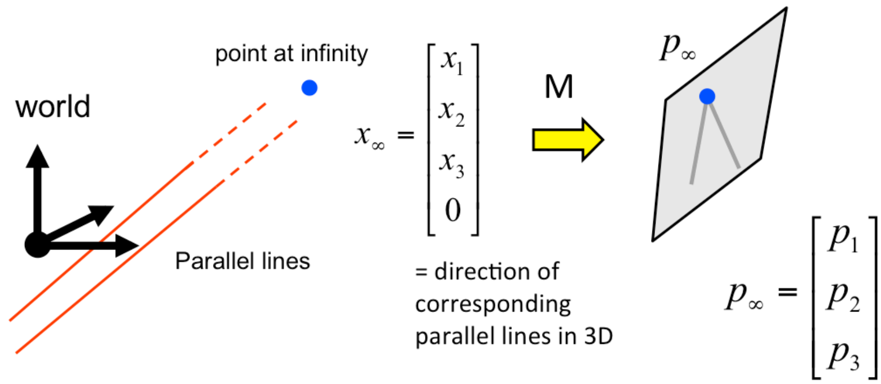


$$x_\infty = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \end{bmatrix}$$

In 3D, similarly to ideals points in 2D, points at infinity are defined as the point of intersection of parallel lines in 3D.

Vanishing points

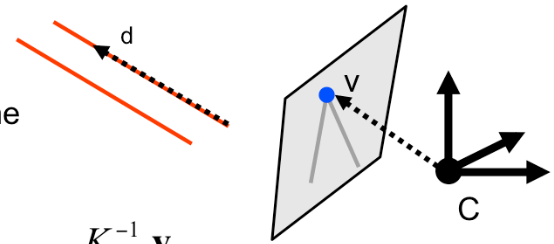
The projective projection of a point at infinity into the image plane defines a vanishing point.



Similarly to the 2D case, by applying the projective transformation M to a point at infinity x_{inf} we obtain a point p_{inf} in the image plane which is no longer at infinity. This is called a **vanishing point.**

Interestingly, similarly to the 2D case, the direction of the corresponding (parallel) lines in 3D associated to x_{inf} is given by the coordinates x_1, x_2 and x_3 of x_{inf}

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:

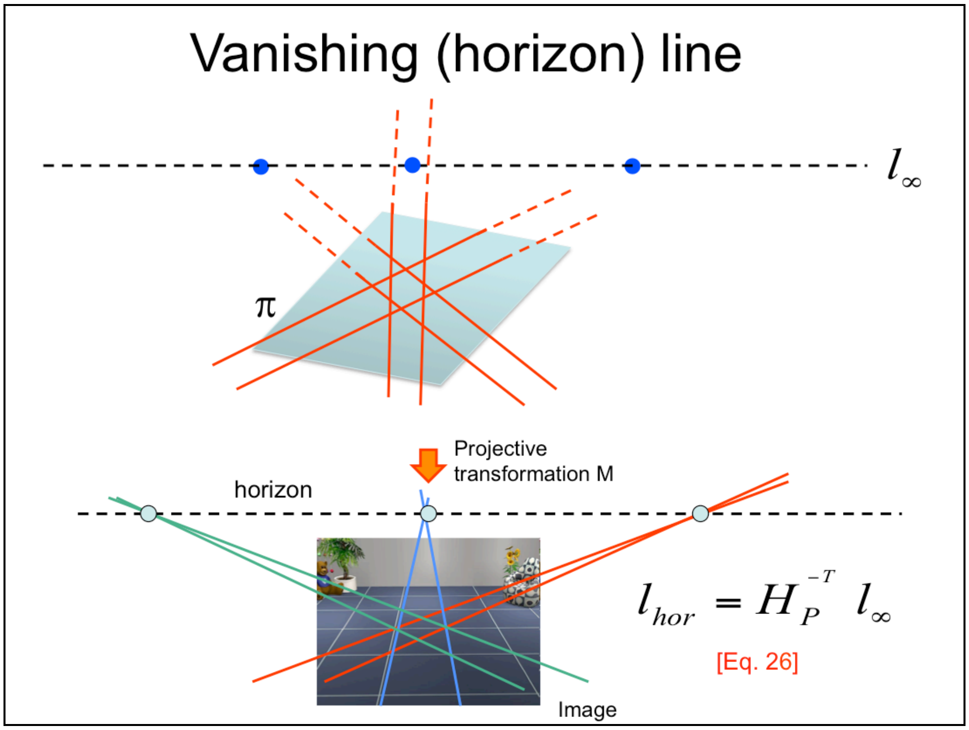
$$X_\infty = \begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix} \xrightarrow{\mathrm{M}} \mathbf{v} = \mathbf{M}\,\mathbf{X}_\infty = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix} = \mathbf{K}\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Next we derive a useful relationship between parallel lines in 3D, the corresponding vanishing point in the image and the camera parameters.

Let's define as **d**=[a,b,c] the direction of a set of parallel lines in 3D in the camera reference system. These lines intersect to a point at infinity and the projection of such point in the image returns the vanishing point **v**.

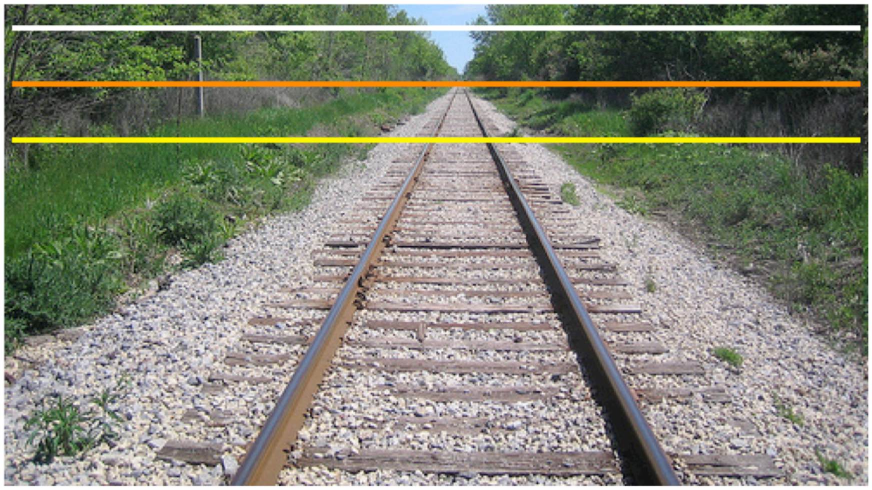
It's easy to prove that **v** is related to **d** via Eq. 24, where K is the camera matrix. Equivalently, **d** can be expressed as function of **v** by Eq. 25, where the division by || K^{-1} **v** || guarantees that **d** has unit norm (d is a direction).

The proof of Eq 24 is reported in the bottom part of the slide.



If we consider a plane π as a superset of set of parallel lines, each set of parallel lines intersects at a point at infinity. The line that passes through such set of points at infinity is the line at infinity l_{inf} associated to π. A line at infinity is also defined as the line where two parallel planes intersect (in general, the intersection of two planes in 3D is a line). The projective transformation of l_{inf} to the image plane is no longer a line at infinity and is called the vanishing line or **horizon** line l_{horiz} (see Eq. 26). The horizon line is a line that passes through the corresponding vanishing points in the image.

Example of horizon line



The orange line is the horizon!

Example of horizon line. Question! What’s the projective transformation of the line at infinity associated to the ground plane – that is the horizon? White, orange or yellow? The orange one!

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

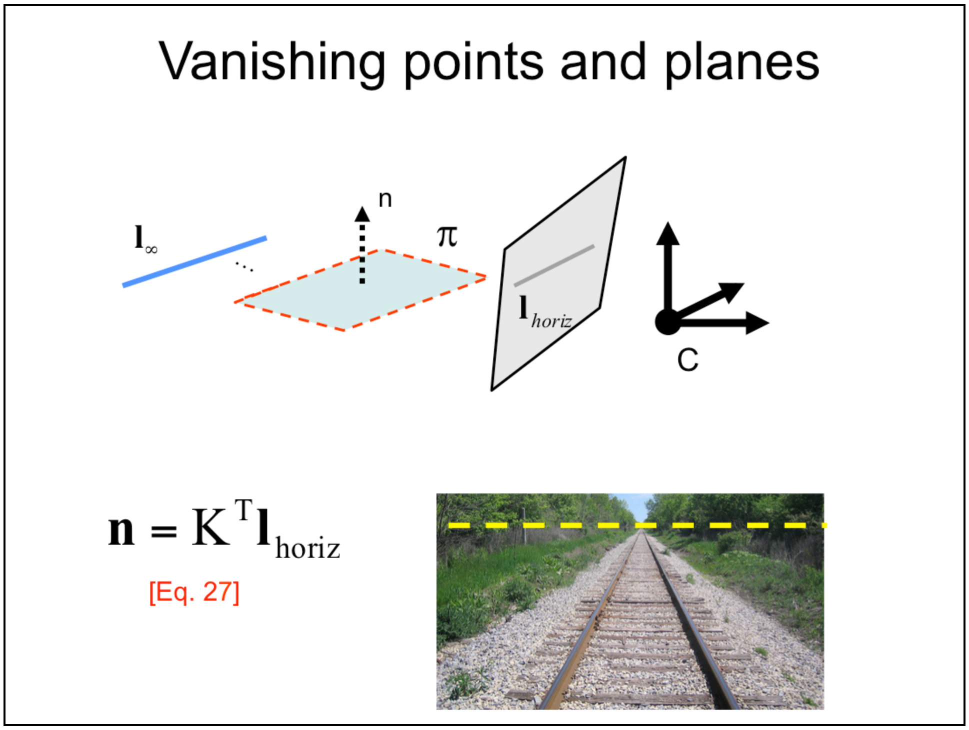
The concept of horizon line allows to answer interesting questions about images. One is the following:

Are the two dashed lines in the image parallel or not?

In order to answer this question we can assume that an oracle tells us that the horizon line is the orange one.

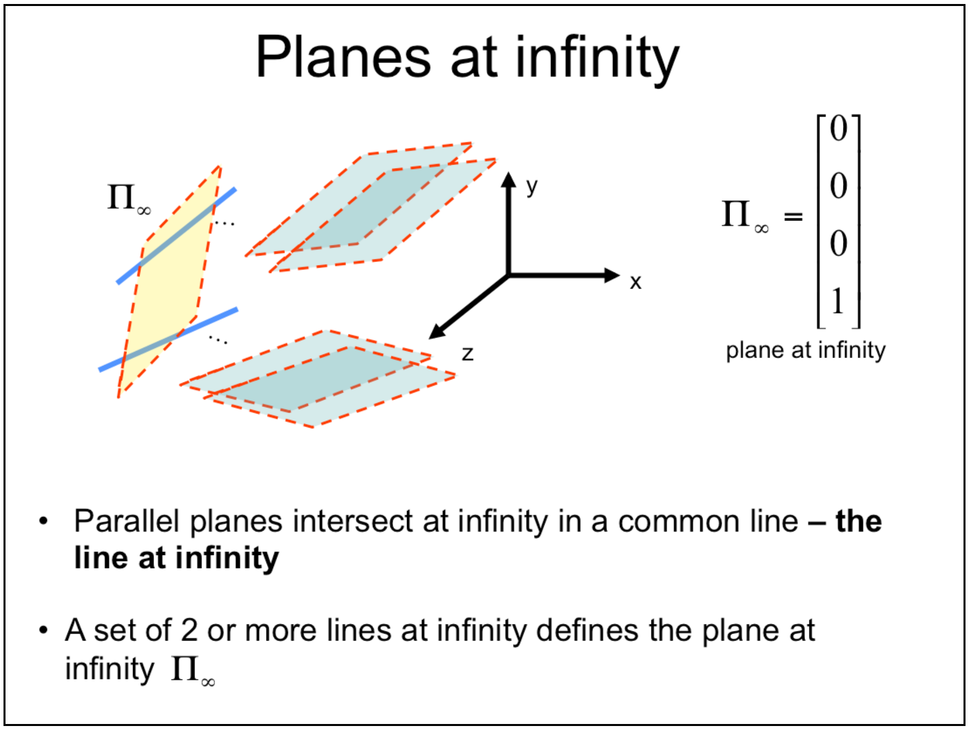
Then, we can verify whether the 2 dashed lines meet at the horizon or not. If yes, these 2 lines are parallel in 3D.

This is an a nice example that shows that recognition can help solve a reconstruction problem. If we recognize the horizon line, we can infer properties about the world (verify the property that two lines are parallel or not). Humans have learnt many of these properties from our daily experience and we use those for solving important estimation problems.



Now we are showing an interesting relationship that connects the normal of a plane in 3D with the corresponding horizon line in the image. This relationship is expressed by Eq. 27 and allows to relate the normal **n** of π and l_{horiz} (see sec. 8.6.2 [HZ] for details), where K is a camera matrix.

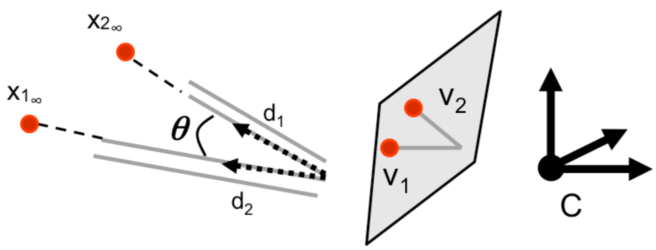
Again, Eq. 27 can be useful for estimating properties of the world. If we recognize the horizon and our camera is calibrated (K is known), we can estimate the orientation of the ground plane.



Before introducing the last property that relates vanishing points and lines, we define the plane at infinity π_∞.

A set of 2 or more vanishing lines (blue lines in the figure) defines the plane at infinity π_∞ (yellow plane in the figure). The plane at infinity is described by $[0\ 0\ 0\ 1]^T$ in homogenous coordinates

Angle between 2 vanishing points



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

[Eq. 28]

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

If $\boldsymbol{\theta} = 90 \rightarrow \boxed{v_1^T \boldsymbol{\omega}\, v_2 = 0}$ [Eq. 29]

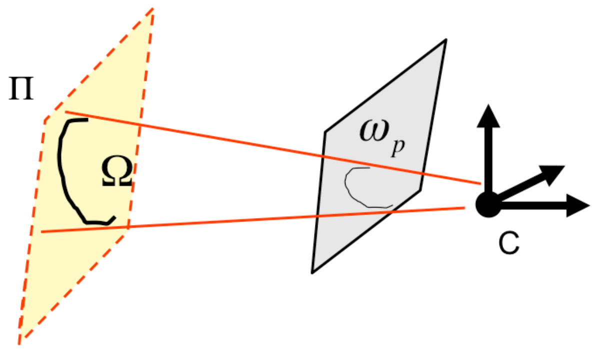
Scalar equation

The last property we introduce allows to relate the direction of lines in 3D with the corresponding vanishing points in the image. Suppose that two pairs of parallel lines have directions $\mathbf{d}_1$ and $\mathbf{d}_2$, and are associated to the points at infinity $\mathbf{x}_{1\infty}$ and $\mathbf{x}_{2\infty}$, respectively. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the corresponding vanishing points. It is possible to prove (but we won't this in this lecture) that the angle between $\mathbf{d}_1$ and $\mathbf{d}_2$ and the vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$ are related by the Eq. 28., where $\boldsymbol{\omega}$ is defined as the matrix $(K\, K^T)^{-1}$.

One special case which is useful in practice is when the two set of parallel lines are orthogonal to each other. In this case, $\mathbf{d}_1$ and $\mathbf{d}_2$ are orthogonal, which gives us [Eq. 29]. Notice that this is a scalar equation.

NOTE: The matrix $\boldsymbol{\omega}$ has a special geometrical meaning in that it is the projective transformation in the image plane of an absolute conic Ω_∞ in 3D.

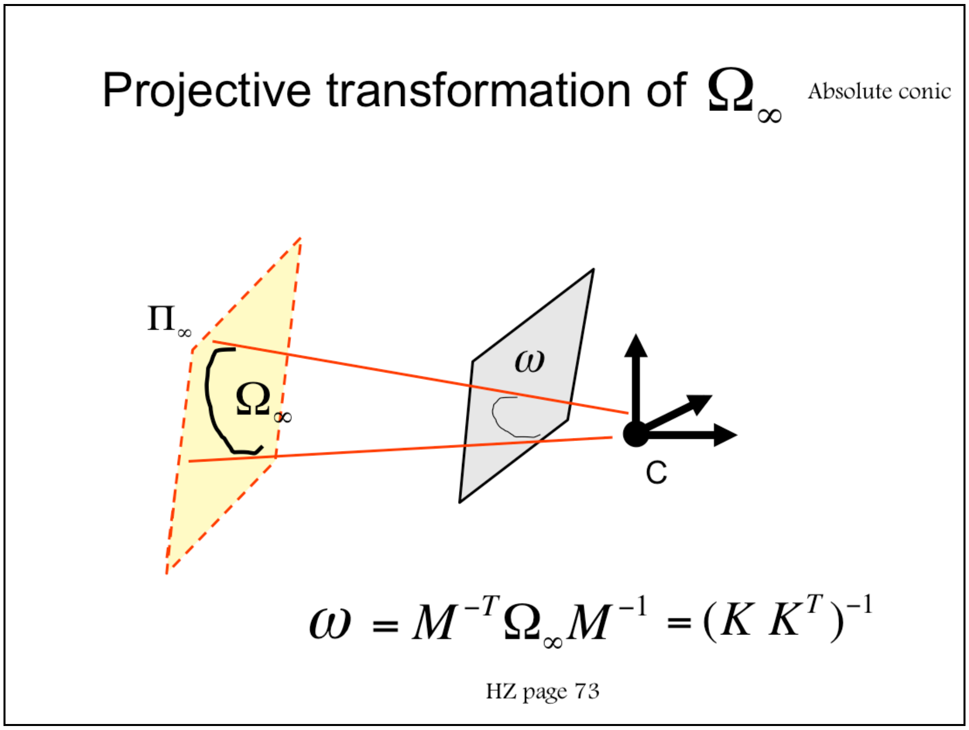
Projective transformation of a conic Ω



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

HZ page 73, eq. 3.16

Ok, let's step back! Let's consider a generic conic Ω in 3D – like a parabola. It can be proven that the projective transformation to image plane of the conic Ω in 3D is given by M^{-T} Ω M^{-1} . For instance, if you have your parabola in 3D you can project it into the image using this expression. The derivation of this is given in HZ page 73, eq. 3.16).



So now, what's the absolute conic Ω_∞? It's a conic that lies in the plane at infinity $[0\ 0\ 0\ 1]^T$.

Thus, the projective transformation in the image plane of an absolute conic Ω_∞ in 3D is $M^{-T}\ \Omega_\infty\ M^{-1}$. It turns out that Ω_∞ is exactly equal to $\boldsymbol{\omega} =$ $(K\ K^T)^{-1}$. For more details please consult HZ.

Projective transformation of Ω_∞

Absolute conic

$$\boxed{\omega = M^{-T}\Omega_\infty M^{-1} = (K\,K^T)^{-1}} \quad \text{[Eq. 30]}$$

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

1. It is not function of R, T

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

3. $\omega_2 = 0$ zero-skew

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

I've summarized these results here. Why this is useful? First, we are going to use this result again when we derive the equations for solving the 3D reconstruction problem from multiple views. Second, because the matrix **ω** satisfies other interesting properties:

- Even if **ω** the result of a projective transformation**,** it depends only on the internal matrix and not on the extrinsic parameters of the camera R and T.

- It is symmetric and known up to scale

- If w_2 is 0, the camera has no skew.

- If $w_1 = w_3$, the camera has square pixels.

Summary

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

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

$$\cos\theta = \frac{v_1^T \omega\, v_2}{\sqrt{v_1^T \omega\, v_1}\sqrt{v_2^T \omega\, v_2}}$$ [Eq. 28]

$$\xrightarrow{\theta = 90} \boxed{v_1^T \omega\, v_2 = 0}$$ [Eq. 29]

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

Useful to:

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

So what we have derived so far?

1. A relationship between parallel lines in 3D and the corresponding vanishing point in the image
2. A relationship that connects the normal of a plane in 3D with the corresponding horizon line in the image.
3. A relationship between the direction of lines in 3D and the corresponding vanishing points in the image.

Notice that in each of these relationships we have K that captures the intrinsic camera parameters.

These properties are useful for two reasons:

- To calibrate the camera: By using Eq 28 or Eq 29, we can set up a system of equations that allows us to calibrate our camera – that is, to estimate internal parameters of the camera
- To estimate the geometry of the 3D world: Once K is estimated or K is known, we can use equation 27 to estimate the orientation of planes in 3D w.r.t. to the camera reference system

Lecture 4

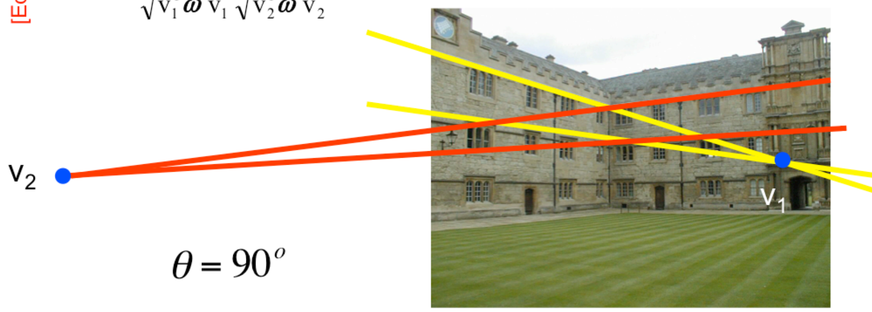
Single View Metrology

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

Reading:
[HZ] Chapter 2 "Projective Geometry and Transformation in 2D"
[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]

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

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

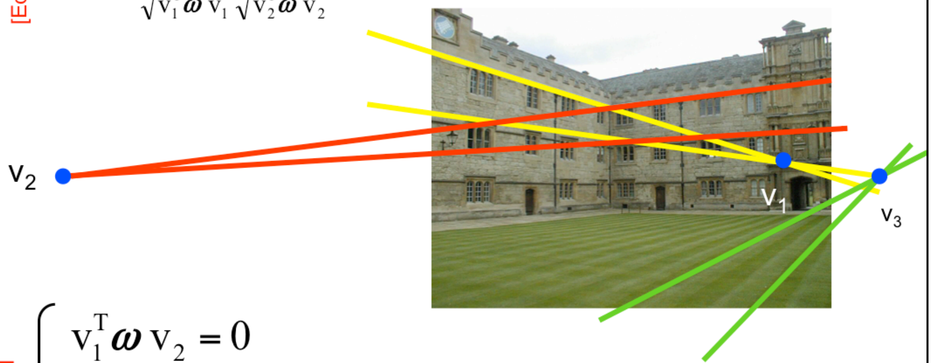
Suppose we can identify two planes in an image of the 3D world (e.g., the two building facades) and suppose we can identify a pair of parallel lines on each of these planes. This allows to estimate two vanishing points in the image v_1 and v_2. Suppose we know that these planes are perpendicular in 3D (the two building facades are perpendicular in 3D). In this case Eq 28 becomes Eq 29. Than we can set up the system of equations above using Eq 29 and the definition of **ω.**

Is this sufficient to estimate the camera parameters? K has in general 5 degrees of freedom and Eq.29 is a scalar equation; so clearly we don't have enough constraints.

Single view calibration - example

[Eq. 28]

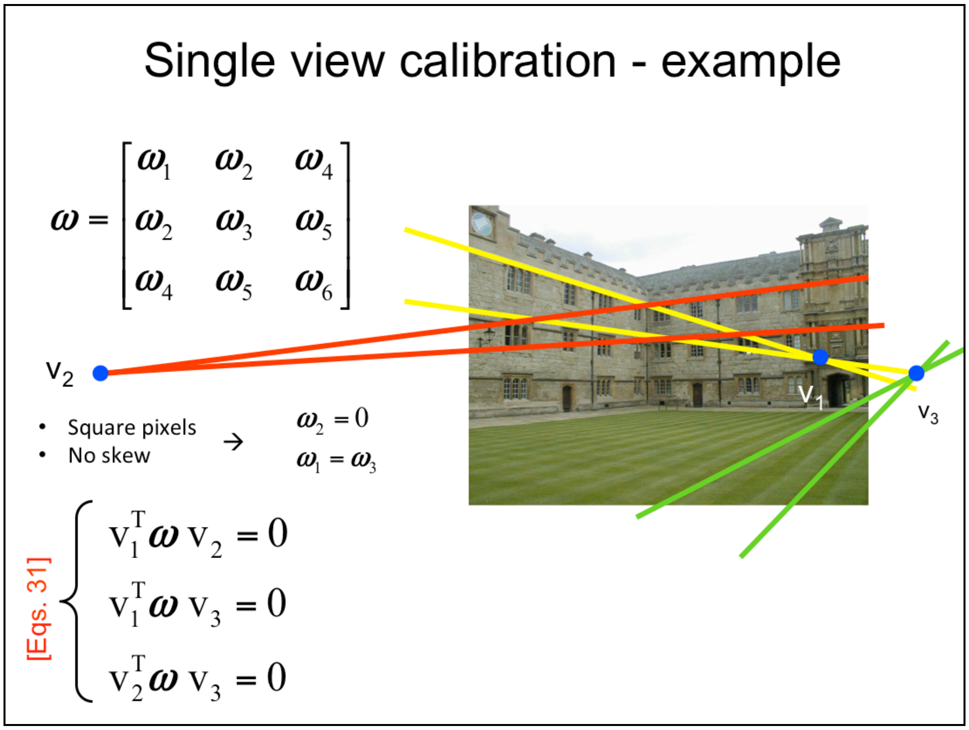
$$\cos\theta = \frac{v_1^T \omega\, v_2}{\sqrt{v_1^T \omega\, v_1}\sqrt{v_2^T \omega\, v_2}}$$



[Eqs. 31]

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

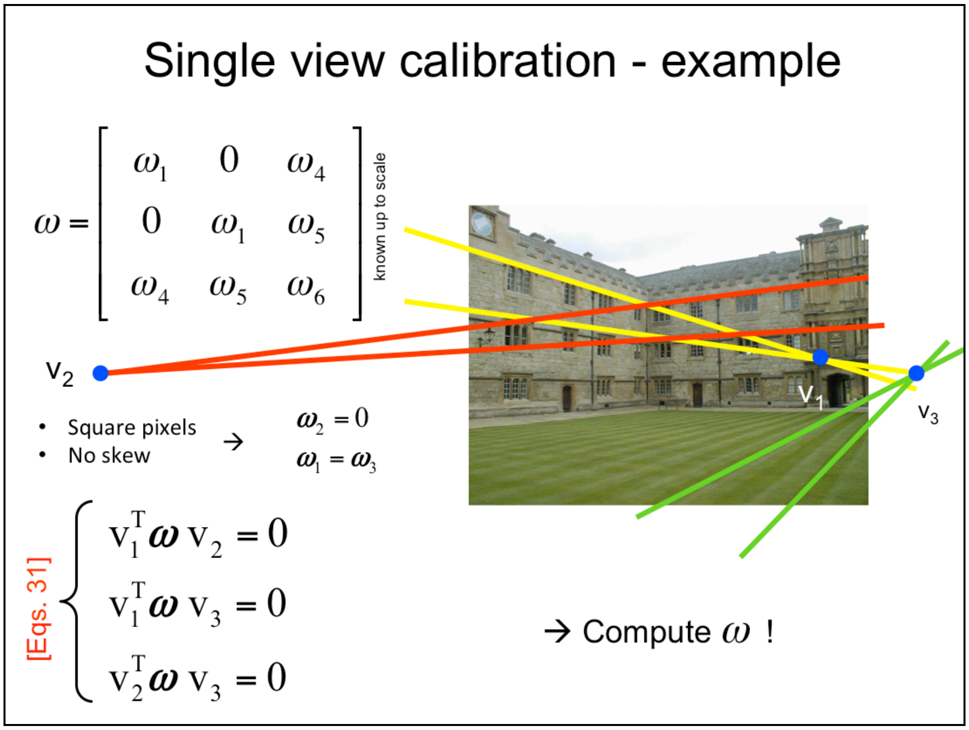
Let’s now assume we can identify another pair of parallel lines and its corresponding vanishing point v_3. With a third vanishing point and by assuming that the 3 set of pairs of parallel lines are pairwise orthogonal in 3D (which is true in this example), we can use Eq. 29 to set up a system of 3 equations (constraints) (Eqs. 31). Do we have enough constraints? Not yet… (3 vs 5 unknown…)



Let’s not take a look again at omega and let’s make some assumptions about the camera. For instance, we can assume that the camera has zero-skew and with square pixels. In this case we have left with only 3 unknowns!

In fact, using the properties of **ω:**

- **ω** is symmetric which means we have 6 unknowns
- **ω** is known up a scale which reduces to 5 unknowns
- $\omega_2=0$ and $\omega_1=\omega_3$ which reduces to 3 unknowns

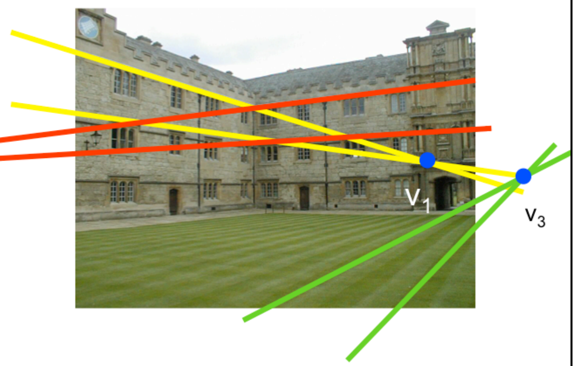


So now we have now enough constraints to solve for the unknowns and can calculate **ω**.

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

Once ω is calculated, we get K:

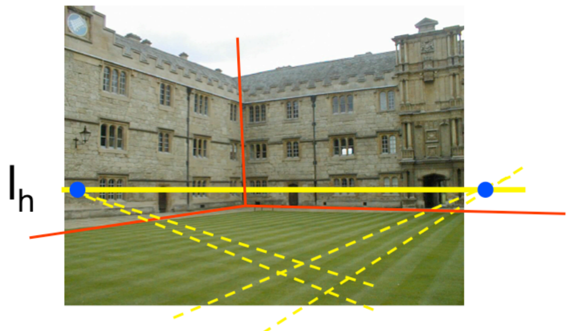
$$\omega = (K\,K^T)^{-1} \longrightarrow K$$

(Cholesky factorization; HZ pag 582)

The actual parameters of K can be computed from **ω** using the Cholesky factorization. We don't proof this results; for more details please refer to HZ pag 582.

Thus, at the end of this procedure we have managed to calibrate the camera from just one single image!

Single view reconstruction - example



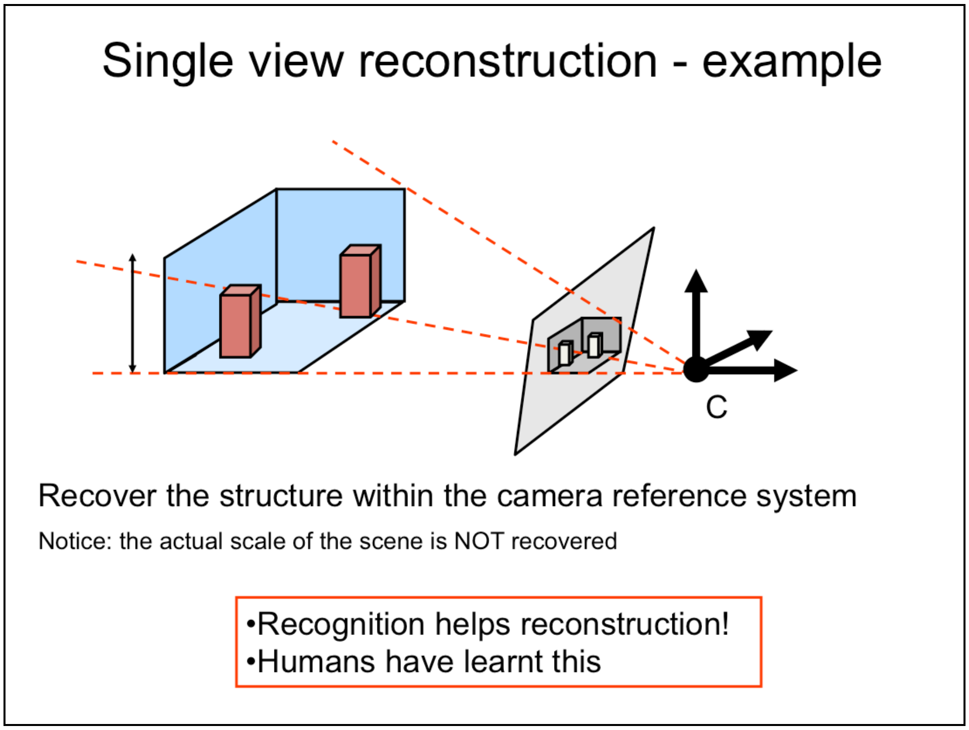
[Eq. 27]

K known $\longrightarrow$ $\mathbf{n} = \mathrm{K}^T \mathbf{l}_{\mathrm{horiz}}$ = Scene plane orientation in the camera reference system

Select orientation discontinuities

Once K is known we can “reconstruct” the geometry of the scene; for instance, we can compute the orientation of all the planes in 3D using Eq. 27.

In order to do so, we need to identify the corresponding lines at infinity and select the orientation discontinuities (that is, where planes fold).



Again, the assortment of tools introduced in this lecture allows us to estimate properties of the camera from observations and/or estimate properties of the world by assuming we have some knowledge about the world (e.g., where the horizon lines are; planes discontinuities; etc…).

Notice that that actual scale of the scene cannot be recovered unless we assume we have access to some measurements in 3D (e.g., a window size); this is similar to what we did when we calibrated the camera.

Lecture 4

Single View Metrology

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

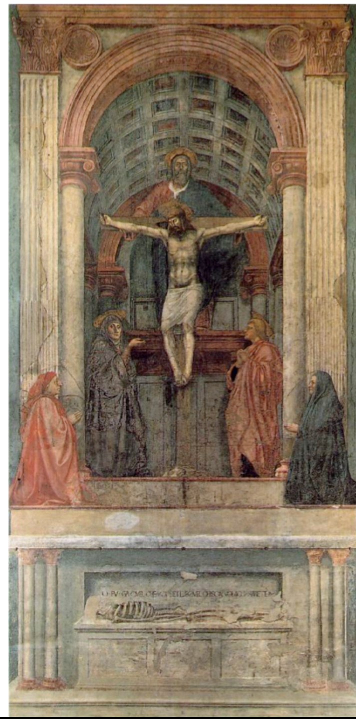
Reading:
[HZ] Chapter 2 “Projective Geometry and Transformation in 3D”
[HZ] Chapter 3 “Projective Geometry and Transformation in 3D”
[HZ] Chapter 8 “More Single View Geometry”
[Hoeim & Savarese] Chapter 2

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

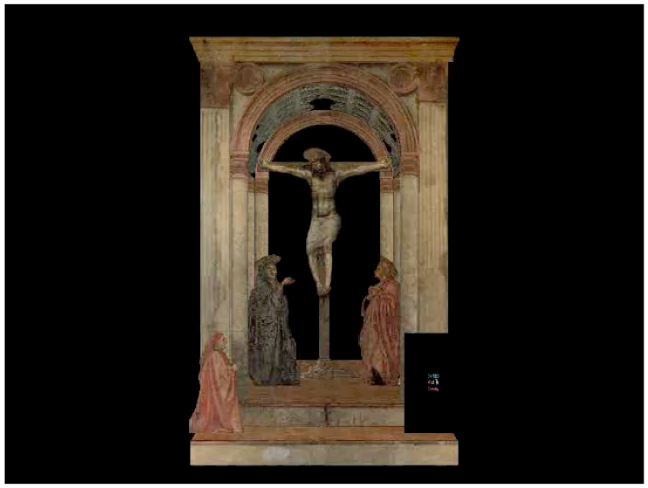
Here we see some results obtained using the approach proposed by A. Criminisi and Zisserman (1999). This approach uses many of the results introduced in this lecture.

Criminisi & Zisserman, 99

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



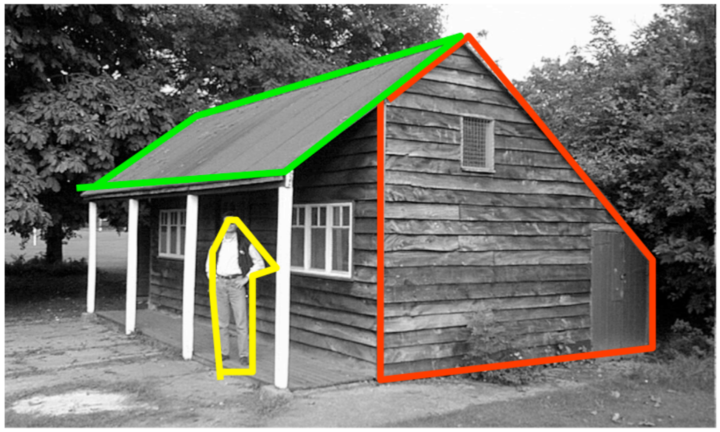
La Trinita' (1426)
Firenze, Santa Maria
Novella; by Masaccio
(1401-1428)



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..

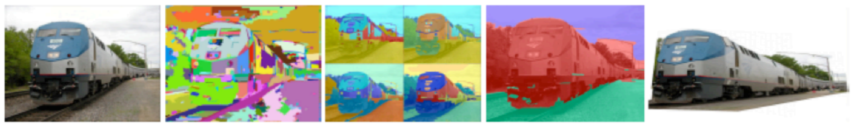
Automatic Photo Pop-up

Hoiem et al, 05

A few years later, D Hoiem proposed an approach where the process of recovering the geometry from a single image is mostly automatic. This approach leverages recognition and segmentation results;

Automatic Photo Pop-up

Hoiem et al, 05…

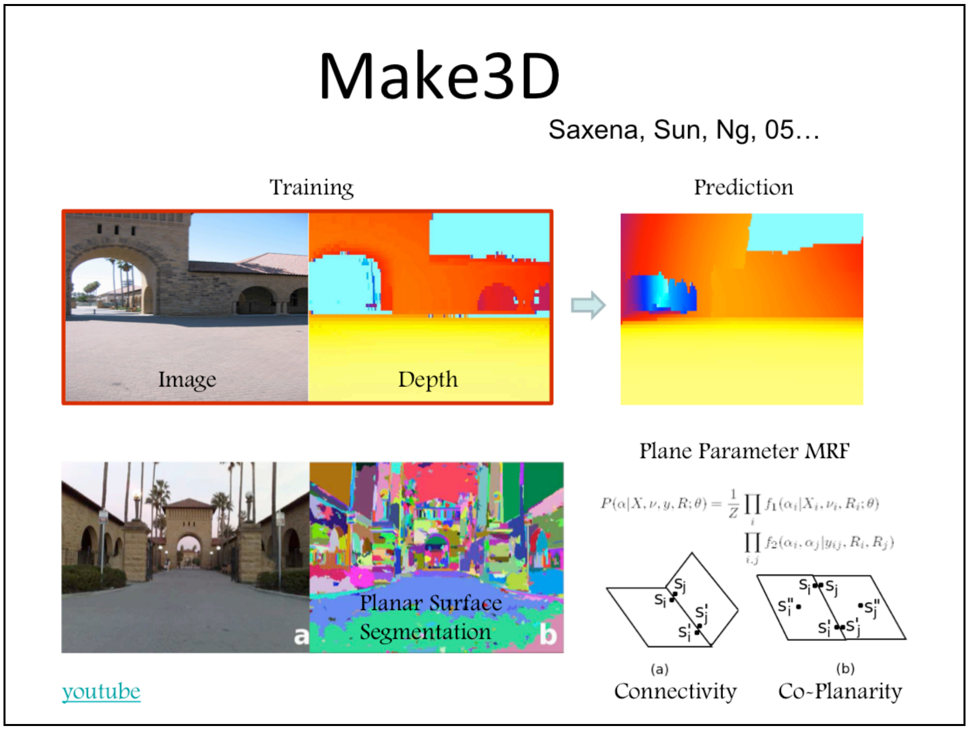


Automatic Photo Pop-up

Hoiem et al, 05…

Software:

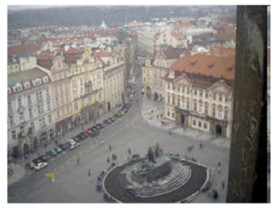
http://www.cs.uiuc.edu/homes/dhoiem/projects/software.html



During the same period, A. Ng and his student A. Saxena at Stanford also demonstrated that it is possible to recover the geometry of the scene from a single image using recognition results and probabilistic inference.

Make3D

Saxena, Sun, Ng, 05…

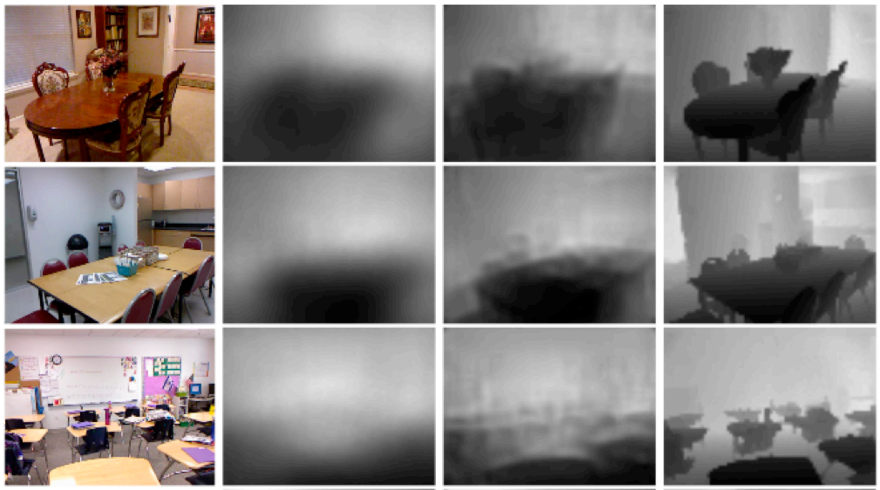

A software: **Make3D**
“Convert your image into 3d model”

http://make3d.stanford.edu/

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

Depth map reconstruction using deep learning

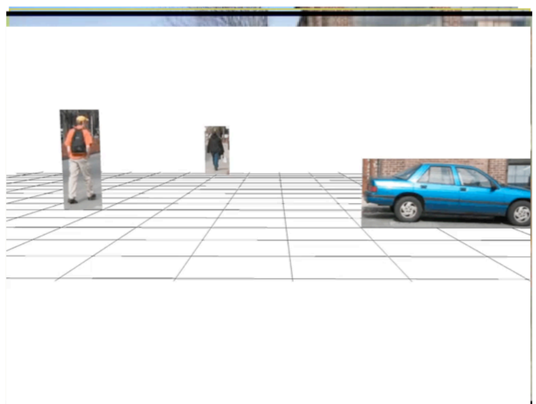
Eigen et al., 2014



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,

Coherent object detection and scene layout estimation from a single image

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



In my own group, we have shown it is possible to combine recognition and reconstruction from a single image in a coherent formulation.

Next lecture:

Multi-view geometry (epipolar geometry)

Appendix

Vanishing points - example

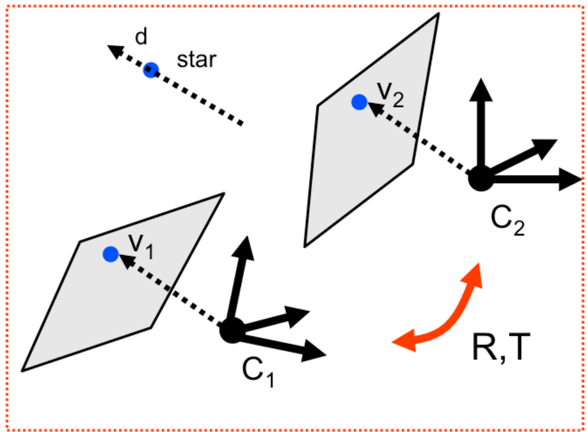
v1, v2: measurements
K = known and constant

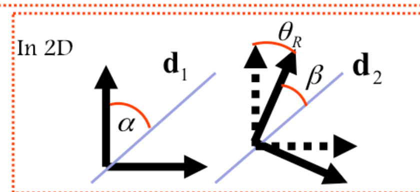
Can I compute R?
No rotation around z

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

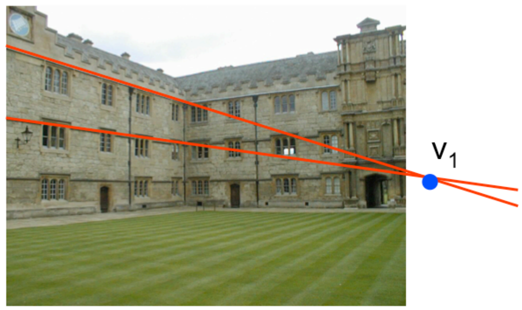
$$\mathbf{d}_2 = \frac{K^{-1}\,\mathbf{v}_2}{\left\| K^{-1}\,\mathbf{v}_2 \right\|}$$

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





$\theta_R = \alpha - \beta$



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

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

$$\longrightarrow \; R$$