NUS National University of Singapore | School of Computing

CS4277 / CS5477
3D Computer Vision

Lecture 5:
Single View Metrology

Assoc Prof. Lee Gim Hee

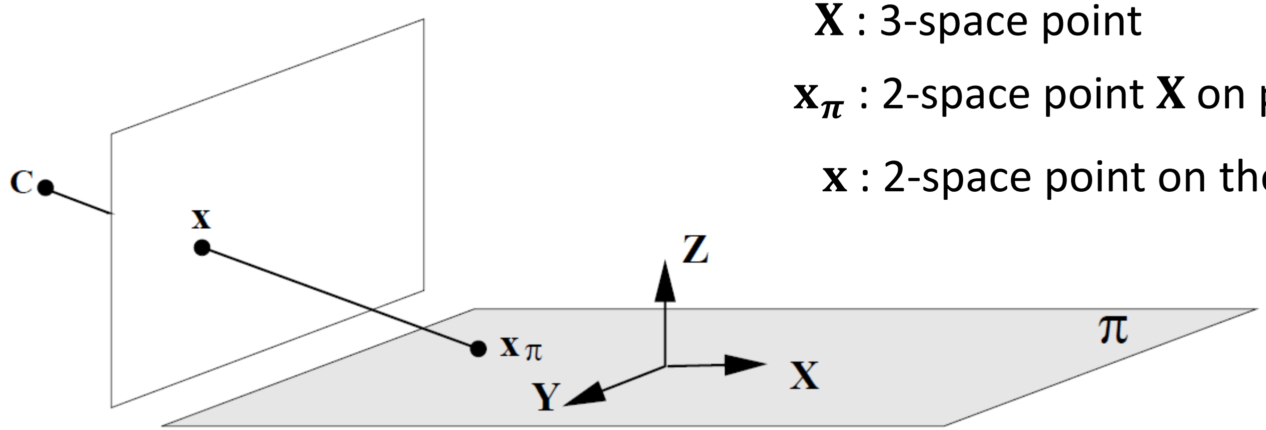
AY 2022/23

Semester 2

Action of Projective Camera on Planes

- Assuming we assign the XY-plane of the world coordinate frame to lie on the plane π, we get

$$\mathrm{x} = \mathrm{PX} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_4 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{pmatrix}.$$



$\mathbf{X}$: 3-space point

$\mathbf{x}_{\boldsymbol{\pi}}$: 2-space point $\mathbf{X}$ on plane $\boldsymbol{\pi}$

$\mathbf{x}$: 2-space point on the image

Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Action of Projective Camera on Planes

- So that the map between points $\mathbf{x}_\pi = (\mathrm{X}, \mathrm{Y}, 1)^\top$ on $\boldsymbol{\pi}$ and their image $\mathbf{x}$ is a general planar homography.
- That is a plane-to-plane projective transformation: $\mathbf{x} = \mathrm{H}\mathbf{x}_{\boldsymbol{\pi}}$, with H a 3×3 matrix of rank 3.

$$\mathbf{x} = \mathrm{PX} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 0 \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_4 \end{bmatrix}}_{\text{Homography H}} \overbrace{\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{pmatrix}}^{\mathbf{x}_\pi} .$$

Action of Projective Camera on Lines

- **Forward projection:** A line in 3-space projects to a line in the image.

- The line and camera centre define a plane, and the image is the intersection of this plane with the image plane.

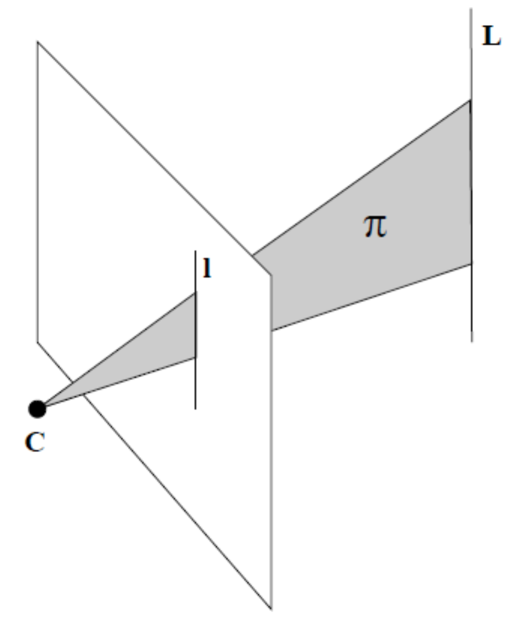
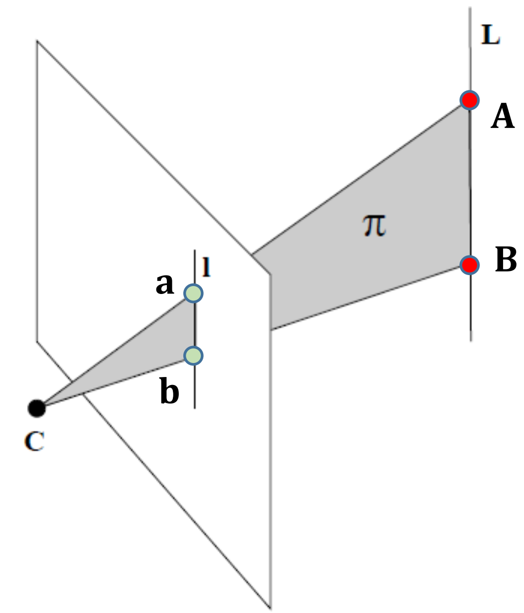


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Action of Projective Camera on Lines

- Given two 3-space points $\mathbf{A}, \mathbf{B}$, where **a**, **b** are their images under P, then a point $\mathbf{X}(\mu) = \mathbf{A} + \mu\mathbf{B}$ on the **L** projects to a point:

$$
\begin{aligned}
\mathbf{x}(\mu) &= \mathrm{P}(\mathbf{A} + \mu\mathbf{B}) = \mathrm{P}\mathbf{A} + \mu\mathrm{P}\mathbf{B} \\
&= \mathbf{a} + \mu\mathbf{b}
\end{aligned}
$$



which is on the line $\mathbf{l}$ joining **a** and **b**.

Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Action of Projective Camera on Lines

- **Back-projection of lines:** The set of points in space which map to a line in the image is a plane in space defined by the camera centre and image line.

- The set of points in space mapping to a line $\mathbf{l}$ via the camera matrix P is the plane $\boldsymbol{\pi} = \mathrm{P}^\top \mathbf{l}$.

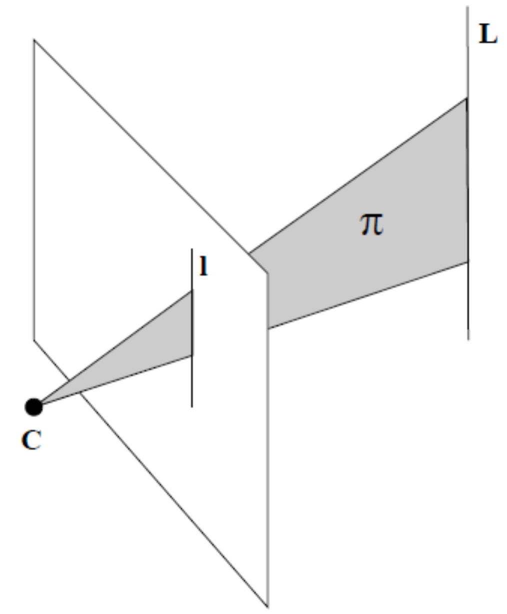


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Action of Projective Camera on Lines

Proof:

- A point $\mathbf{x}$ lies on $\mathbf{l}$ if and only if $\mathbf{x}^{\top}\mathbf{l} = 0$.
- A space point $\mathbf{X}$ maps to a point $\mathrm{P}\mathbf{X}$, which lies on the line if and only if $\mathbf{X}^{\top}\mathrm{P}^{\top}\mathbf{l} = 0$.

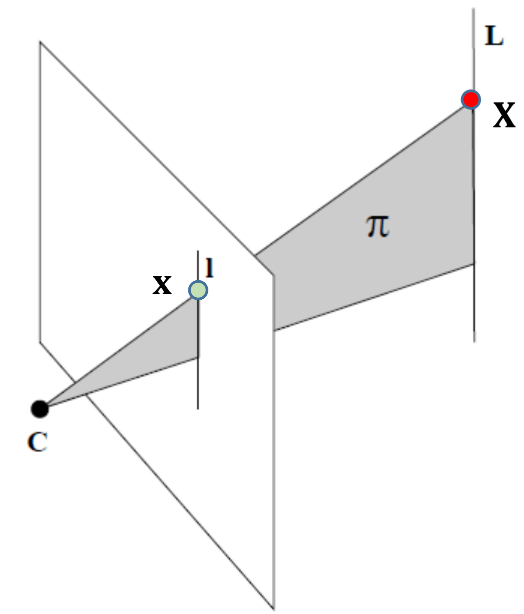


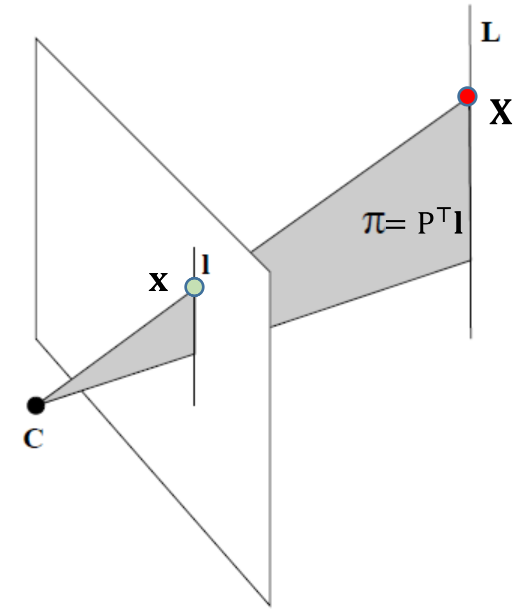
Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Action of Projective Camera on Lines

Proof:

- Thus, if $P^\top \mathbf{l}$ is taken to represent a plane, then $\mathbf{X}$ lies on this plane if and only if $\mathbf{X}$ maps to a point on the line $\mathbf{l}$.

- In other words, $P^\top \mathbf{l}$ is the back-projection of the line $\mathbf{l}$.



□

Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

The Importance of the Camera Centre

- An object in 3-space and camera centre define a set of rays, and an image is obtained by intersecting these rays with a plane.
- Often this set is referred to as a *cone* of rays, even though it is not a classical cone.

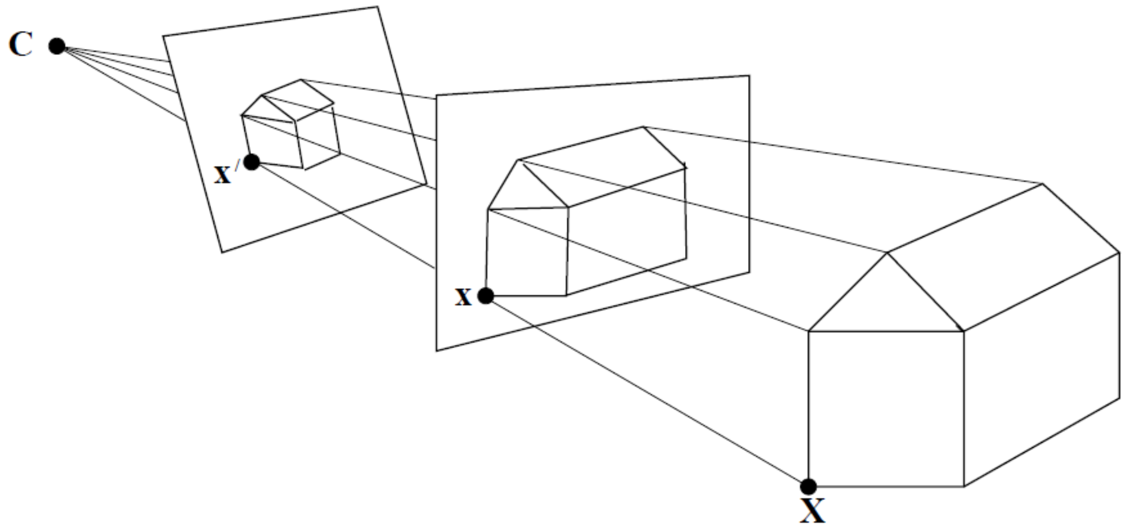


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

The Importance of the Camera Centre

- Images obtained with the same camera centre may be mapped to one another by a plane projective transformation, i.e. homography.

- In other words, they are projectively equivalent and so have the same projective properties.

- A camera can thus be thought of as a projective imaging device – measuring projective properties of the cone of rays with vertex the camera centre.

The Importance of the Camera Centre

- We now show that the two images, I and I', with the same camera centre are clearly related by a homography.
- Consider two cameras $\mathrm{P} = \mathrm{KR}[\mathrm{I} \mid -\widetilde{\mathbf{C}}], \quad \mathrm{P}' = \mathrm{K}'\mathrm{R}'[\mathrm{I} \mid -\widetilde{\mathbf{C}}]$ with the same centre, i.e. $\mathrm{P}' = (\mathrm{K}'\mathrm{R}')(\mathrm{KR})^{-1}\mathrm{P}$.
- It then follows that the images of a 3-space point $\mathbf{X}$ by the two cameras are related as

$$\mathbf{x}' = \mathrm{P}'\mathbf{X} = (\mathrm{K}'\mathrm{R}')(\mathrm{KR})^{-1}\mathrm{P}\mathbf{X} = (\mathrm{K}'\mathrm{R}')(\mathrm{KR})^{-1}\mathbf{x}.$$

- That is, the corresponding image points are related by a planar homography (a 3 × 3 matrix) as $\mathbf{x} = \mathrm{H}\mathbf{x}$, where $\mathrm{H} = (\mathrm{KR})(\mathrm{KR})^{-1}$.

The Importance of the Camera Centre

Moving the image plane (increase focal length):

- This corresponds to a displacement of the image plane along the principal axis, where the image effect is a simple magnification.
- If $\mathbf{x}$, $\mathbf{x}'$ are the images of a point $\mathbf{X}$ before and after zooming, then

$$\begin{aligned} \mathbf{x} &= \mathrm{K}[\mathrm{I} \mid \mathbf{0}]\mathbf{X} \\ \mathbf{x}' &= \mathrm{K}'[\mathrm{I} \mid \mathbf{0}]\mathbf{X} = \mathrm{K}'\mathrm{K}^{-1}\left(\mathrm{K}[\mathrm{I} \mid \mathbf{0}]\mathbf{X}\right) = \mathrm{K}'\mathrm{K}^{-1}\mathbf{x} \end{aligned}$$

so that $\mathbf{x}' = \mathrm{H}\mathbf{x}$ with $\mathrm{H} = \mathrm{K}'\mathrm{K}^{-1}$.

The Importance of the Camera Centre

Moving the image plane (increase focal length):

- If only the focal lengths differ between K and K' then

$$\mathrm{K}'\mathrm{K}^{-1} = \begin{bmatrix} k\mathrm{I} & (1-k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

- where $\tilde{\mathbf{x}}_0$ is the inhomogeneous principal point, and $k = f'/f$ is the magnification factor.

- Consequently, the effect of zooming by a factor k is to multiply the calibration matrix K on the right by $\mathrm{diag}(k, k, 1)$:

$$\begin{aligned} \mathrm{K}' &= \begin{bmatrix} k\mathrm{I} & (1-k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathrm{K} = \begin{bmatrix} k\mathrm{I} & (1-k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathrm{A} & \tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \\ &= \begin{bmatrix} k\mathrm{A} & \tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} = \mathrm{K} \begin{bmatrix} k\mathrm{I} & \\ & 1 \end{bmatrix}. \end{aligned}$$

The Importance of the Camera Centre

Camera rotation:

- Here we consider the camera is rotated about its centre with no change in the internal parameters.

- If $\mathbf{x}$, $\mathbf{x}'$ are the images of a point $\mathbf{X}$ before and after the pure rotation:

$$\begin{aligned} \mathbf{x} &= \mathrm{K}[\mathrm{I} \mid \mathbf{0}]\mathbf{X} \\ \mathbf{x}' &= \mathrm{K}[\mathrm{R} \mid \mathbf{0}]\mathbf{X} = \mathrm{KRK}^{-1}\mathrm{K}[\mathrm{I} \mid \mathbf{0}]\mathbf{X} = \mathrm{KRK}^{-1}\mathbf{x} \end{aligned}$$

so that $\mathbf{x}' = \mathrm{H}\mathbf{x}$ with $\mathrm{H} = \mathrm{KRK}^{-1}$.

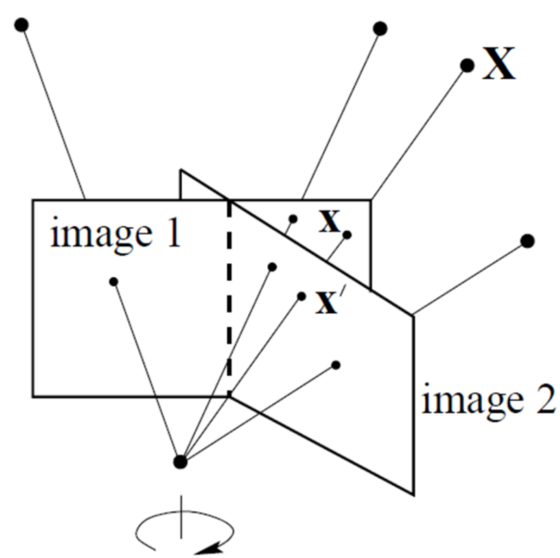


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

The Importance of the Camera Centre

Properties of a conjugate rotation:

- This homography $\mathrm{H} = \mathrm{KRK}^{-1}$ is a conjugate rotation.
- It has the same eigenvalues (up to scale) as the rotation matrix, i.e. $\{\mu, \mu e^{i\theta}, \mu e^{-i\theta}\}$.
- μ is an unknown scale factor (if H is scaled such that $\det \mathrm{H} = 1$, then $\mu = 1$).
- The angle of rotation between views may be computed directly from the phase of the complex eigenvalues of H.

The Importance of the Camera Centre

Moving the camera centre (Motion parallax):

- No information on 3-space structure can be obtained by zooming and pure rotation, i.e. with fixed camera centres.
- Corresponding image points *does* depend on the 3-space structure if the camera centre is moved.
- May often be used to (partially) determine the structure.
- More details in subsequent lectures.

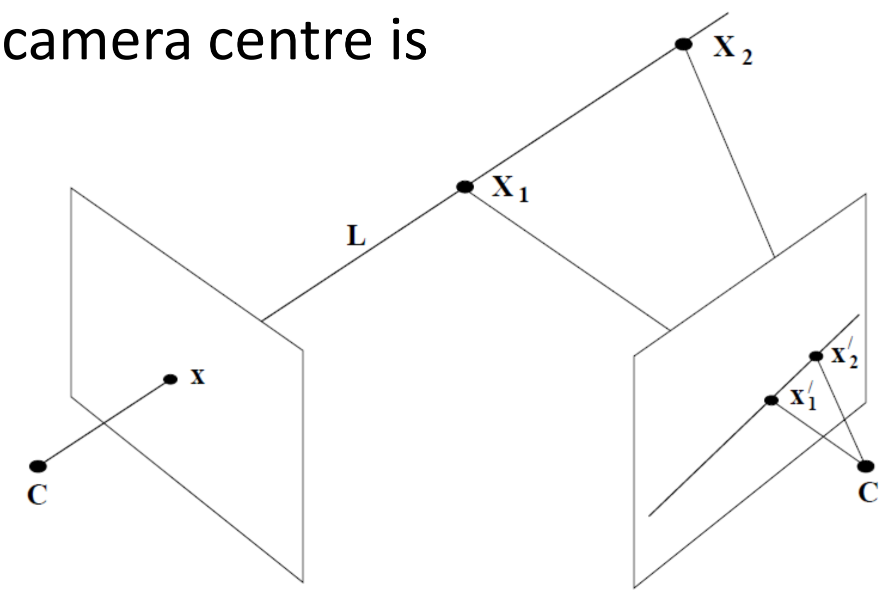


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Applications and Examples

Example: Synthetic Views

- New images corresponding to different camera orientations (same camera centre) can be generated from an existing image by warping with planar homographies.

Source image → Fronto-parallel views of floor and wall

homographies

Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Applications and Examples

Example: Synthetic Views

The algorithm is:

i. Compute the homography H which maps the image quadrilateral to a rectangle with the correct aspect ratio.

ii. Projectively warp the source image with this homography.

Applications and Examples

Example: Planar Panoramic Mosaicing

- Images acquired by a camera rotating about its centre are related to each other by a planar homography.

- A set of such images may be registered with the plane of one of the images by projectively warping the other images.

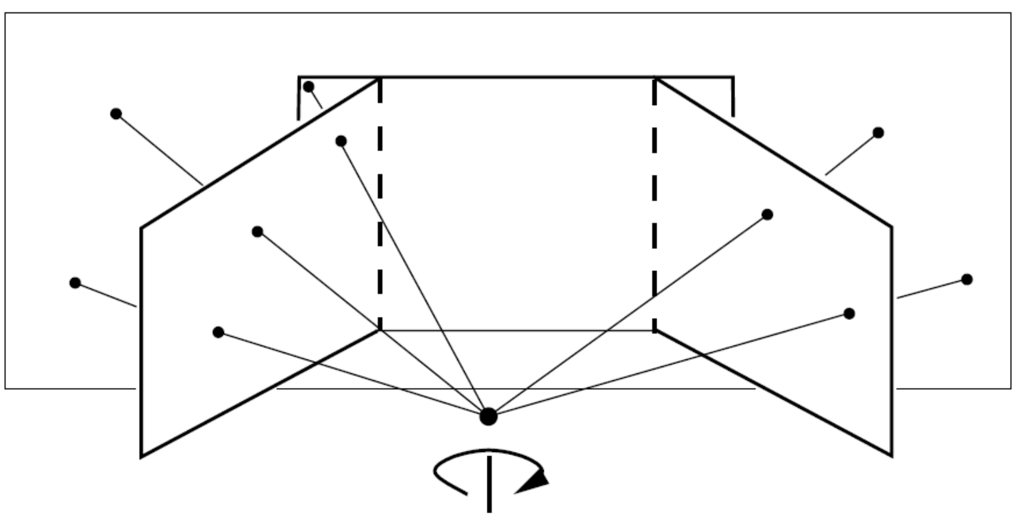


Image source: “Multiple View Geometry in Computer Vision”, Richard Hartley and Andrew Zisserman

Applications and Examples

Image source: “Multiple View Geometry in Computer Vision”, Richard Hartley and Andrew Zisserman

Applications and Examples

Example: Planar Panoramic Mosaicing

In outline the algorithm is:

i. Choose one image of the set as a reference.

ii. Compute the homography H (4-point) which maps one of the other images of the set to this reference image.

iii. Projectively warp the image with this homography, and augment the reference image with the non-overlapping part of the warped image.

iv. Repeat the last two steps for the remaining images of the set.

What does Calibration Give?

- Suppose points on the ray are written as $\widetilde{\mathbf{X}} = \lambda \mathbf{d}$ in the camera Euclidean coordinate frame.

- Then, these points map to the point

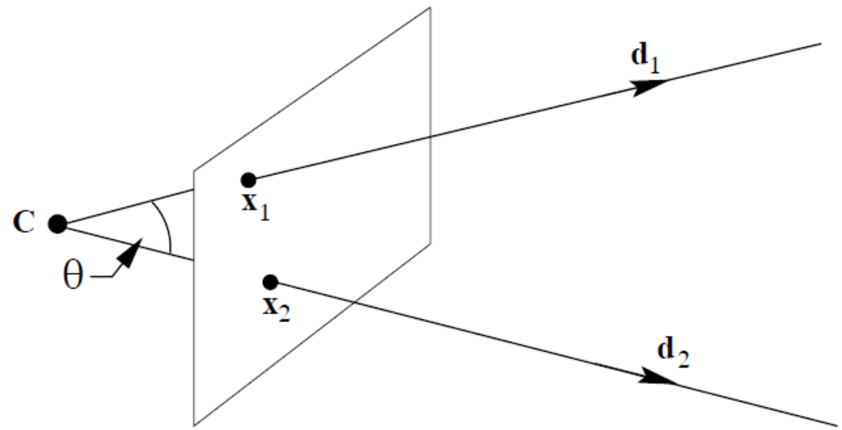
$$\mathbf{x} = \mathrm{K}[\mathrm{I} \mid \mathbf{0}](\lambda \mathbf{d}^\top, 1)^\top = \mathrm{K}\mathbf{d}$$

  up to scale.

- Conversely, the direction $\mathbf{d}$ is obtained from the image point $\mathbf{x}$ as $\mathbf{d} = \mathrm{K}^{-1}\mathbf{x}$.

- Note, $\mathbf{d} = \mathrm{K}^{-1}\mathbf{x}$ is in general *not* a unit vector.

What does Calibration Give?



- The angle between two rays, with directions $\mathbf{d}_1$, $\mathbf{d}_2$ corresponding to image points $\mathbf{x}_1$, $\mathbf{x}_2$ respectively, may be obtained:

$$
\begin{aligned}
\cos\theta &= \frac{\mathbf{d}_1^{\mathsf{T}}\mathbf{d}_2}{\sqrt{\mathbf{d}_1^{\mathsf{T}}\mathbf{d}_1}\sqrt{\mathbf{d}_2^{\mathsf{T}}\mathbf{d}_2}} = \frac{(\mathtt{K}^{-1}\mathbf{x}_1)^{\mathsf{T}}(\mathtt{K}^{-1}\mathbf{x}_2)}{\sqrt{(\mathtt{K}^{-1}\mathbf{x}_1)^{\mathsf{T}}(\mathtt{K}^{-1}\mathbf{x}_1)}\sqrt{(\mathtt{K}^{-1}\mathbf{x}_2)^{\mathsf{T}}(\mathtt{K}^{-1}\mathbf{x}_2)}} \\
&= \frac{\mathbf{x}_1^{\mathsf{T}}(\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1})\mathbf{x}_2}{\sqrt{\mathbf{x}_1^{\mathsf{T}}(\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1})\mathbf{x}_1}\sqrt{\mathbf{x}_2^{\mathsf{T}}(\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1})\mathbf{x}_2}} \; .
\end{aligned}
$$

Image source: “Multiple View Geometry in Computer Vision”, Richard Hartley and Andrew Zisserman

What does Calibration Give?

- A camera for which K is known is termed calibrated, and thus the matrix $\mathrm{K}^{-\top}\mathrm{K}^{-1}$ is known.
- Then the angle between rays can be measured from their corresponding image points.
- A calibrated camera is a direction sensor, able to measure the direction of rays – like a 2D protractor.

What does Calibration Give?

- An image line $\mathbf{l}$ defines a plane through the camera centre with normal direction $\mathbf{n} = \mathrm{K}^\top \mathbf{l}$ measured in the camera's Euclidean coordinate frame.

Proof:

- Points $\mathbf{x}$ on the line $\mathbf{l}$ back-project to directions $\mathbf{d} = \mathrm{K}^{-1}\mathbf{x}$.
- Which are orthogonal to the plane normal $\mathbf{n}$, and thus satisfy $\mathbf{d}^\top \mathbf{n} = \mathbf{x}^\top \mathrm{K}^{-\top} \mathbf{n} = 0$.
- Since points on $\mathbf{l}$ satisfy $\mathbf{x}^\top \mathbf{l} = 0$, it follows that $\mathbf{l} = \mathrm{K}^{-\top}\mathbf{n}$, and hence $\mathbf{n} = \mathrm{K}^\top \mathbf{l}$. □

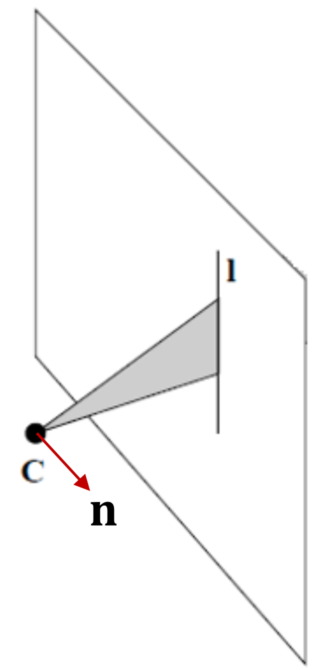


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

The Image of the Absolute Conic

- Points on $\boldsymbol{\pi}_\infty$ may be written as $\mathbf{X}_\infty = (\mathbf{d}^\top, 0)^\top$, and are imaged by a general camera $\mathrm{P} = \mathrm{KR}[\mathrm{I} \mid -\tilde{\mathbf{C}}]$ as:

$$\mathbf{x} = \mathrm{P}\mathbf{X}_\infty = \mathrm{KR}[\mathrm{I} \mid -\widetilde{\mathrm{C}}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = \mathrm{KR}\mathbf{d}.$$

- This shows that the mapping between $\boldsymbol{\pi}_\infty$ and an image is given by the planar homography $\mathbf{x} = \mathrm{H}\mathbf{d}$ with:

$$\mathrm{H} = \mathrm{KR}.$$

- This map is independent of the position of camera **C**, and depends only on the camera internal calibration and orientation w.r.t the world frame.

The Image of the Absolute Conic

- Now, since the absolute conic Ω_∞ is on $\boldsymbol{\pi}_\infty$ we can compute its image under H.

- And find that the image of the absolute conic (the IAC) is the conic $\boldsymbol{\omega} = (\mathrm{K}\mathrm{K}^\top)^{-1} = \mathrm{K}^{-\top}\mathrm{K}^{-1}$.

- Like Ω_∞ the conic $\boldsymbol{\omega}$ is an imaginary point conic with no real points.

- Nonetheless, we will see some of its practical uses later.

The Image of the Absolute Conic

Proof:

- Under a point homography $\mathbf{x} \mapsto \mathrm{H}\mathbf{x}$ a conic C maps as $\mathrm{C} \mapsto \mathrm{H}^{-\top}\mathrm{C}\mathrm{H}^{-1}$.

- It follows that Ω_∞, which is the conic $\mathrm{C} = \Omega_\infty = \mathrm{I}$ on $\boldsymbol{\pi}_\infty$, maps to $\omega = (\mathrm{KR})^{-\top}\mathrm{I}(\mathrm{KR})^{-1} = \mathrm{K}^{-\top}\mathrm{R}\mathrm{R}^{-1}\mathrm{K}^{-1} = (\mathrm{K}\mathrm{K}^\top)^{-1}$.

- So, the IAC is $\boldsymbol{\omega} = (\mathrm{K}\mathrm{K}^\top)^{-1}$.

□

The Image of the Absolute Conic

- A few remarks here:

i. The image of the absolute conic, $\boldsymbol{\omega}$, depends only on the internal parameters K of the matrix P; it does not depend on the camera orientation or position.

ii. The angle between two rays we seen earlier can now be expressed with $\boldsymbol{\omega}$, i.e.

$$\cos\theta = \frac{\mathbf{x}_1^\top \boldsymbol{\omega} \mathbf{x}_2}{\sqrt{\mathbf{x}_1^\top \boldsymbol{\omega} \mathbf{x}_1}\sqrt{\mathbf{x}_2^\top \boldsymbol{\omega} \mathbf{x}_2}} .$$

The Image of the Absolute Conic

- This expression is unchanged under projective transformation of the image.

Proof:
Let's consider the numerator $\mathbf{x}_1^\top \boldsymbol{\omega} \mathbf{x}_2$. Under any projective transformation $\mathbf{x}' = \mathrm{H}\mathbf{x}$, the numerator becomes:

$$(\mathbf{x}_1^\top \mathrm{H}^\top)(\mathrm{H}^{-\top} \omega \mathrm{H}^{-1})(\mathrm{H}\mathbf{x}_2) = \mathbf{x}_1^\top \boldsymbol{\omega} \mathbf{x}_2$$

It can also be easily shown that H is also canceled out in the demoninator.

$\square$

The Image of the Absolute Conic

iii. A direct result of (ii) is: if two image points $\mathbf{x}_1$ and $\mathbf{x}_2$ correspond to orthogonal directions then

$$\mathbf{x}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_2 = 0.$$

iv. We may also define the dual image of the absolute conic (the DIAC) as

$$\boldsymbol{\omega}^* = \boldsymbol{\omega}^{-1} = \mathrm{K}\mathrm{K}^\mathsf{T}.$$

- This is a dual (line) conic, whereas $\boldsymbol{\omega}$ is a point conic (though it contains no real points).
- The conic $\boldsymbol{\omega}^*$ is the image of Q_∞^* and is given by $\boldsymbol{\omega}^* = \mathrm{P}\mathrm{Q}_\infty^*\mathrm{P}^\mathsf{T}$.

The Image of the Absolute Conic

v. Once $\boldsymbol{\omega}$ (or equivalently $\boldsymbol{\omega}^*$) is identified in an image, K can be identified uniquely via Cholesky factorization, i.e. $\boldsymbol{\omega}^* = \mathrm{KK}^\top$.

vi. The imaged circular points lie on $\boldsymbol{\omega}$ at the points at which the vanishing line of the plane $\boldsymbol{\pi}$ intersects $\boldsymbol{\omega}$.

- We saw in Lecture 2 that a plane $\boldsymbol{\pi}$ intersects $\boldsymbol{\pi}_\infty$ in a line, and this line intersects Ω_∞ in two points which are the circular points of $\boldsymbol{\pi}$.

Example: A Simple Calibration Device

- The image of three squares (on planes which are not parallel, but which need not be orthogonal) provides sufficiently many constraints to compute K.

Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Example: A Simple Calibration Device

Outline the calibration algorithm:

1. For each square, compute the homography H that maps its corner points, $(0,0)^\top$, $(1,0)^\top$, $(0,1)^\top$, $(1,1)^\top$, to their imaged points.

   Remarks:

   The alignment of the plane coordinate system with the square is a similarity transformation and does not affect the position of the circular points on the plane.

Example: A Simple Calibration Device

2. Compute the imaged circular points for the plane of that square as $\mathrm{H}(1, \pm i, 0)^\top$; and writing $\mathrm{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$, the imaged circular points are $\mathbf{h}_1 \pm i\mathbf{h}_2$.

3. Fit a conic $\boldsymbol{\omega}$ to the six imaged circular points.

   If $\mathbf{h}_1 \pm i\mathbf{h}_2$ lies on $\boldsymbol{\omega}$ then $(\mathbf{h}_1 \pm i\mathbf{h}_2)^\top \boldsymbol{\omega} (\mathbf{h}_1 \pm i\mathbf{h}_2) = 0$, and the imaginary and real parts give respectively:

$$\mathbf{h}_1^\top \omega \mathbf{h}_2 = 0 \quad \text{and} \quad \mathbf{h}_1^\top \omega \mathbf{h}_1 = \mathbf{h}_2^\top \omega \mathbf{h}_2$$

   which are linear in $\boldsymbol{\omega}$, then the conic $\boldsymbol{\omega}$ is determined up to scale from five or more such equations.

Example: A Simple Calibration Device

4. Compute the calibration K from $\boldsymbol{\omega} = (\mathrm{K}\mathrm{K}^{\top})^{-1}$ using the Cholesky factorization.

$$\mathrm{K} = \begin{bmatrix} 1108.3 & -9.8 & 525.8 \\ 0 & 1097.8 & 395.9 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Three squares provide a simple calibration object. The planes need not be orthogonal. (b) The computed calibration matrix using the algorithm mentioned earlier. The image size is 1024×768 pixels.

Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Vanishing Points

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

- Geometrically, the vanishing point of a line is obtained by intersecting the image plane with a ray parallel to the world line and passing through the camera centre.

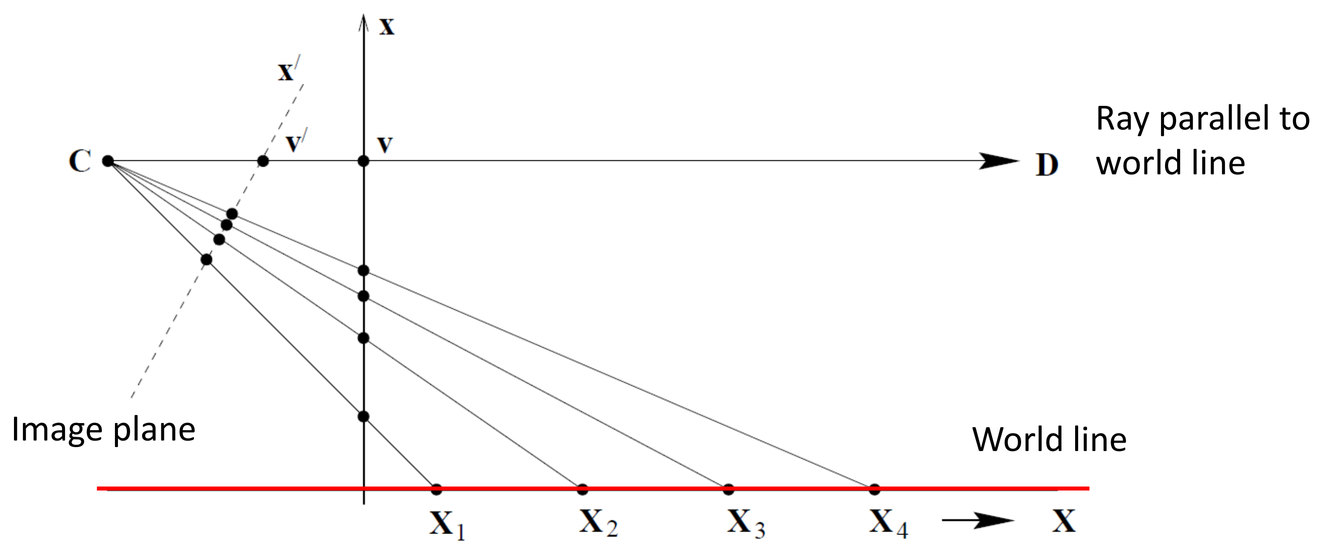


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Vanishing Points

- Thus, a vanishing point depends only on the direction of a line, not on its position.
- Consequently, a set of parallel world lines have a common vanishing point.

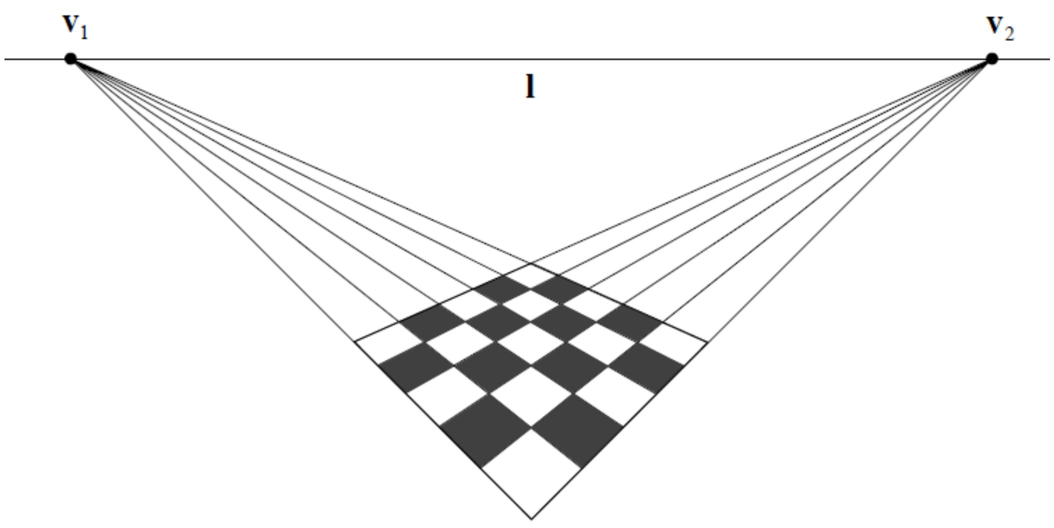


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Vanishing Points

- Algebraically, the vanishing point may be obtained as a limiting point as follows:

1. Points on a line in 3-space through the point $\mathbf{A}$ and with direction $\mathbf{D} = (\mathbf{d}^{\top}, 0)^{\top}$ are written as $\mathbf{X}(\lambda) = \mathbf{A} + \lambda\mathbf{D}$.

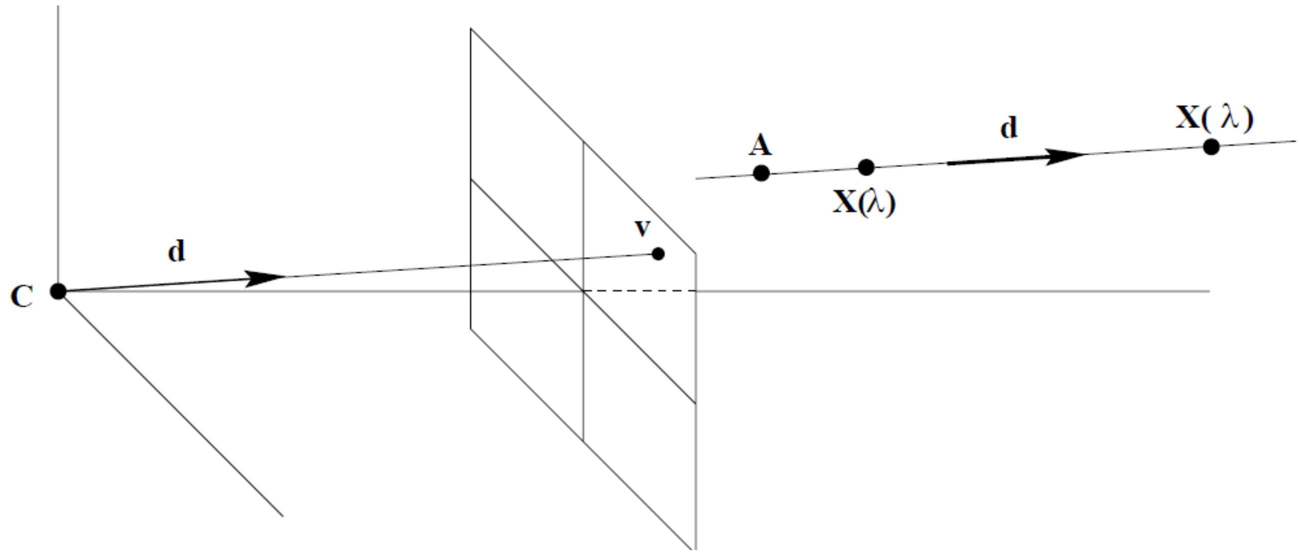


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Vanishing Points

2. Under a projective camera $\mathrm{P} = \mathrm{K}[\mathrm{I} \mid \mathbf{0}]$, a point $\mathbf{X}(\lambda)$ is imaged at:

$$\mathbf{x}(\lambda) = \mathrm{P}\mathbf{X}(\lambda) = \mathrm{P}\mathbf{A} + \lambda\mathrm{P}\mathbf{D} = \mathbf{a} + \lambda\mathrm{K}\mathbf{d} \,,$$

where $\mathbf{a}$ is the image of $\mathbf{A}$.

3. Then, the vanishing point $\mathbf{v}$ of the line is obtained as the limit:

$$\mathbf{v} = \lim_{\lambda\to\infty} \mathbf{x}(\lambda) = \lim_{\lambda\to\infty} (\mathbf{a} + \lambda\mathrm{K}\mathbf{d}) = \mathrm{K}\mathbf{d}.$$

Note that $\mathbf{v}$ depends only on the direction $\mathbf{d}$ of the line, not on its position specified by $\mathbf{A}$.

Vanishing Points

- In projective 3-space, the vanishing point is simply the image of the intersection of the plane at infinity $\boldsymbol{\pi}_\infty$ and a set of lines with the same direction $\mathbf{d}$, i.e.

$$\mathbf{v} = \mathtt{P}\mathbf{X}_\infty = \mathtt{K}[\mathtt{I} \mid \mathbf{0}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = \mathtt{K}\mathbf{d}.$$

- Note, lines parallel to the image plane are imaged as parallel lines since $\mathbf{v}$ is at infinity in the image.

- However, parallel image lines might not be the image of parallel scene lines since lines which intersect on the principal plane are imaged as parallel lines.

Vanishing Points

Example: rotation estimation from vanishing points.

- Suppose two cameras have the same calibration matrix K, and the camera rotates by R between views.
- Let a scene line have vanishing point $\mathbf{v}_i$ in the first view, and $\mathbf{v}'_i$ in the second, where the directions are given by:

$$\mathbf{d}_i = \mathrm{K}^{-1}\mathbf{v}_i / \| \mathrm{K}^{-1}\mathbf{v}_i \| \ , \qquad \text{(a unit vector).}$$

- Two independent constraints on R are given by $\mathbf{d}'_i = \mathrm{R}\mathbf{d}_i$, thus R can be computed from two such corresponding directions.

Vanishing Points

Example: angle between two scene lines.

- Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the vanishing points of two lines in an image, and let $\boldsymbol{\omega}$ be the image of the absolute conic in the image.

- If θ is the angle between the two line directions, then

$$\cos\theta = \frac{\mathbf{v}_1^{\mathsf{T}}\boldsymbol{\omega}\mathbf{v}_2}{\sqrt{\mathbf{v}_1^{\mathsf{T}}\boldsymbol{\omega}\mathbf{v}_1}\sqrt{\mathbf{v}_2^{\mathsf{T}}\boldsymbol{\omega}\mathbf{v}_2}} .$$

Computing Vanishing Points

Chicken-and-egg problem:

1. Under known vanishing points, we can compute the corresponding set of imaged parallel scene lines.
2. Under known set of imaged parallel scene lines, we can compute the vanishing points.

Problem: Both are unknown!

Image source: “Multiple View Geometry in Computer Vision”, Richard Hartley and Andrew Zisserman

Computing Vanishing Points

- We'll skip the details of computing vanishing points by just giving several references:

1. Grant Schindler, Frank Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments", CVPR 2004.

2. Jean-Philippe Tardif, "Non-Iterative Approach for Fast and Accurate Vanishing Point Detection", ICCV 2009.

3. Gim Hee Lee, " Line Association and Vanishing Point Estimation with Binary Quadratic Programming", 3DV 2017.

Vanishing Lines

- Parallel planes in 3-space intersect $\boldsymbol{\pi}_\infty$ in a common line, and the image of this line is the vanishing line of the plane.

- Geometrically the vanishing line is constructed by intersecting the image with a plane parallel to the scene plane through the camera centre.

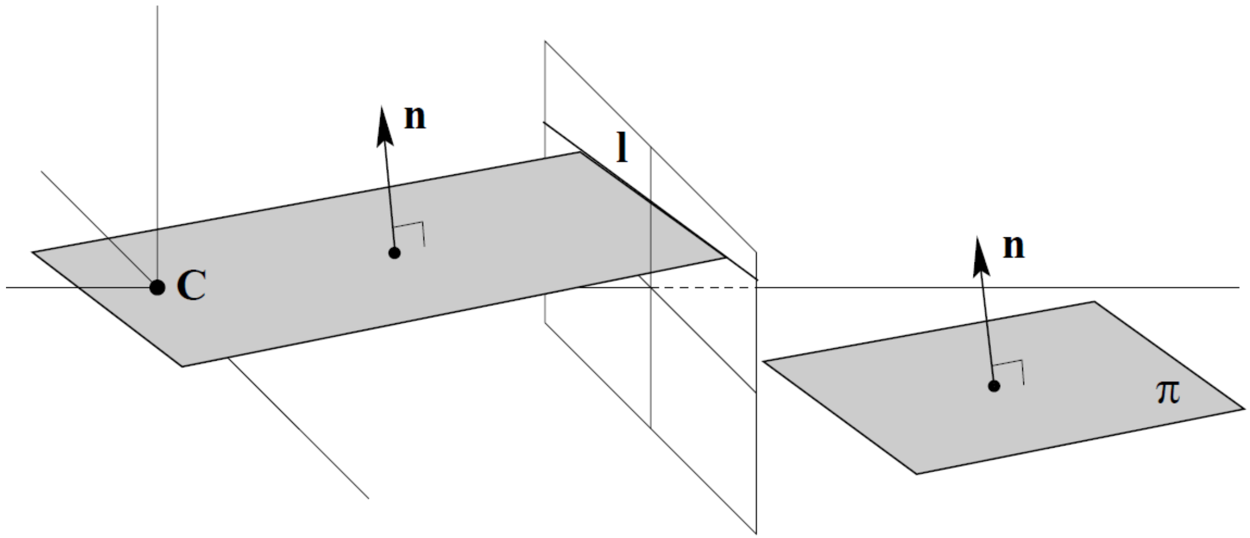


Image source: “Multiple View Geometry in Computer Vision”, Richard Hartley and Andrew Zisserman

Vanishing Lines

- Vanishing line depends only on the orientation of the scene plane; it does not depend on its position.

- Since lines parallel to a plane intersect the plane at $\boldsymbol{\pi}_{\infty}$, the vanishing point of a line parallel to a plane lies on the vanishing line of the plane.

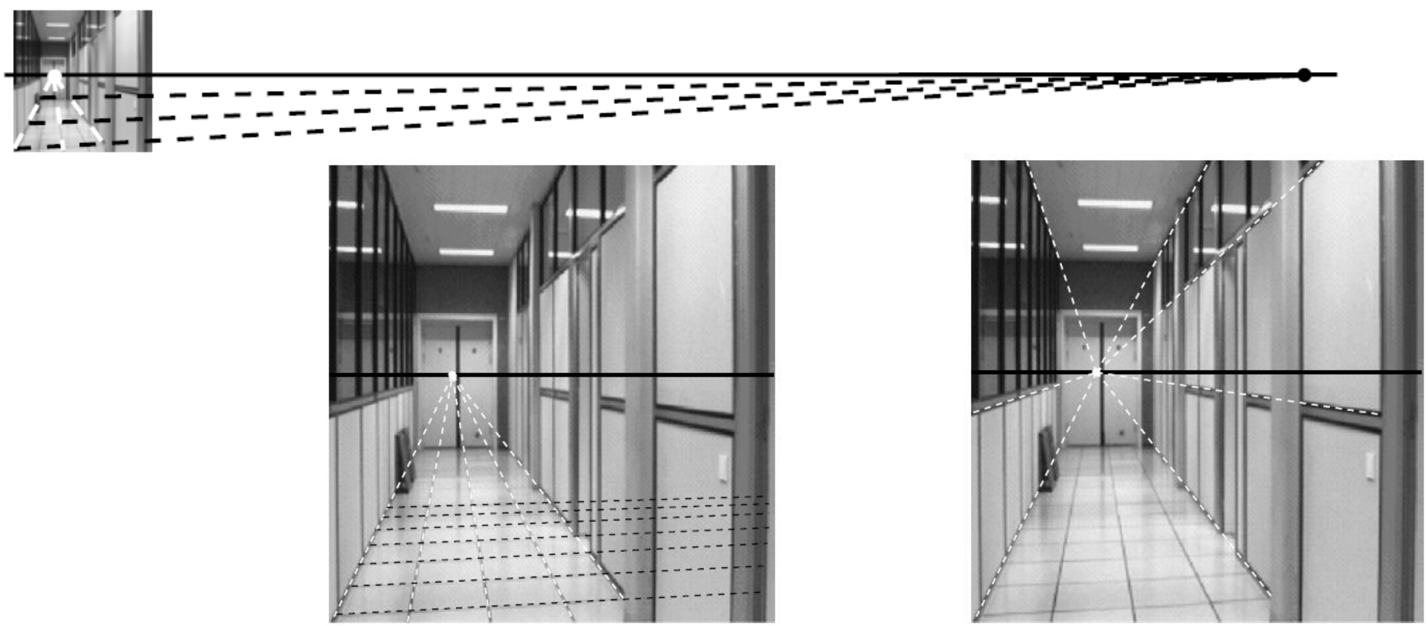


Image source: “Multiple View Geometry in Computer Vision”, Richard Hartley and Andrew Zisserman

Vanishing Lines

- If the camera calibration K is known, then a scene plane's vanishing line may be used to determine information about the plane.
- We will look at three examples.

Vanishing Lines

Case 1:

- The plane's orientation relative to the camera may be determined from its vanishing line.
- A plane through the camera centre with normal direction $\mathbf{n}$ intersects the image plane in the line $\mathbf{l} = \mathrm{K}^{-\top}\mathbf{n}$.
- Consequently, $\mathbf{l}$ is the vanishing line of planes perpendicular to $\mathbf{n}$.
- Thus, a plane with vanishing line $\mathbf{l}$ has orientation $\mathbf{n} = \mathrm{K}^{\top}\mathbf{l}$ in the camera's Euclidean coordinate frame.

Vanishing Lines

Case 2:

- The plane may be metrically rectified given only its vanishing line.
- Since the plane normal is known from the vanishing line, the camera can be synthetically rotated by a homography so that the plane is fronto-parallel (i.e. parallel to the image plane).

Vanishing Lines

Case 3:

- The angle between two scene planes can be determined from their vanishing lines.

- Suppose the vanishing lines are $\mathbf{l}_1$ and $\mathbf{l}_2$, then the angle θ between the planes is given by

$$\cos\theta = \frac{\mathbf{l}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{l}_2}{\sqrt{\mathbf{l}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{l}_1}\sqrt{\mathbf{l}_2^\mathsf{T}\boldsymbol{\omega}^*\mathbf{l}_2}}.$$

Exercise: Prove it!

Computing Vanishing Lines

- A common way to determine a vanishing line of a scene plane is:

1. Determine vanishing points for two sets of lines parallel to the plane, and then

2. Construct the line through the two vanishing points.

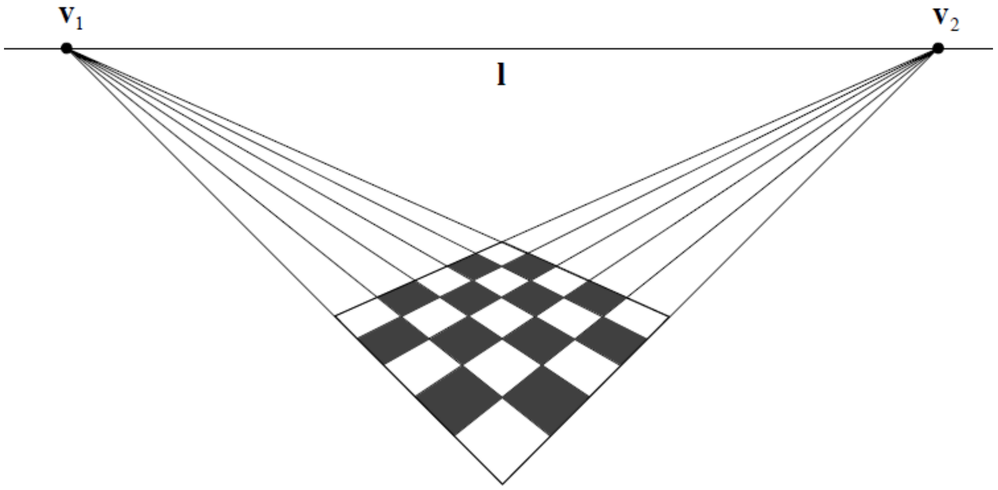


Image source: “Multiple View Geometry in Computer Vision”, Richard Hartley and Andrew Zisserman

Orthogonality Relationships: Vanishing Points and Lines

The orthogonality relationships among vanishing points and lines can be used to determine $\boldsymbol{\omega}$:

i. The vanishing points of lines with perpendicular directions satisfy

$$\mathbf{v}_1^{\mathsf{T}} \boldsymbol{\omega} \mathbf{v}_2 = 0.$$

ii. If a line is perpendicular to a plane, then their respective vanishing point **v** and vanishing line **l** are related by

$$\mathbf{l} = \boldsymbol{\omega} \mathbf{v} \quad \text{and inversely} \quad \mathbf{v} = \boldsymbol{\omega}^* \mathbf{l}.$$

iii. The vanishing lines of two perpendicular planes satisfy $\mathbf{l}_1^{\mathsf{T}} \boldsymbol{\omega}^* \mathbf{l}_2 = 0.$

Affine 3D Measurements

- Given the vanishing line of the ground plane $\mathbf{l}$ and the vertical vanishing point **v.**
- Then the relative length of vertical line segments can be measured provided their end point lies on the ground plane.

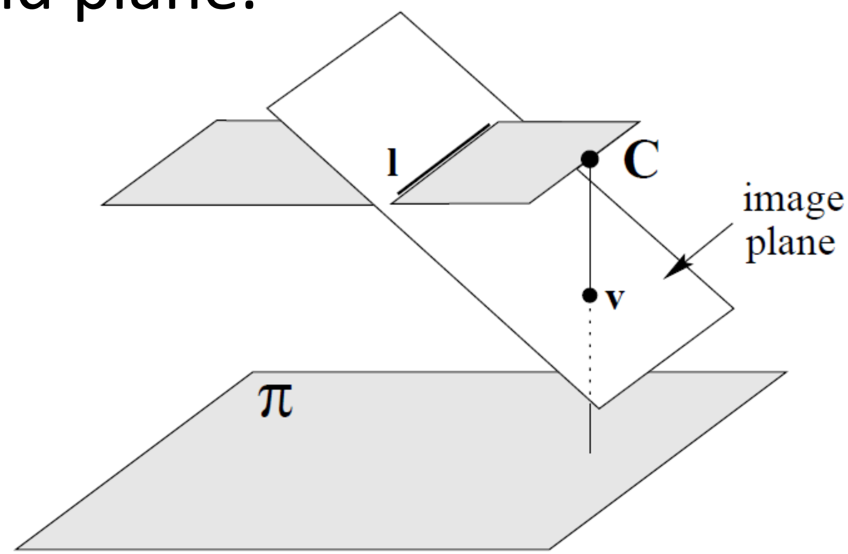
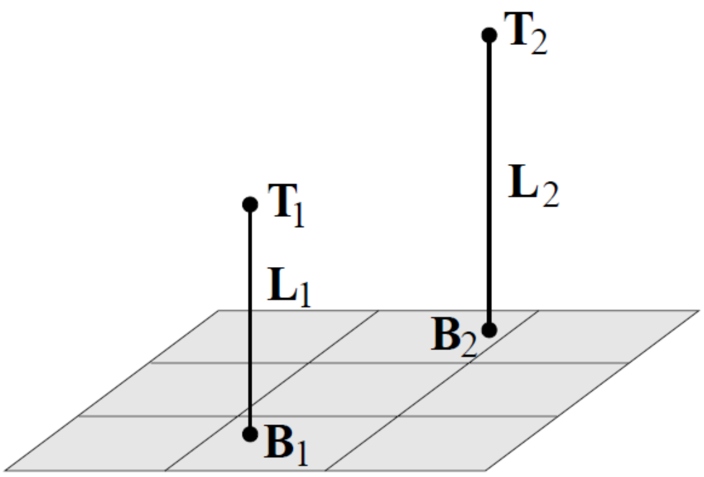


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Affine 3D Measurements

- **Given:** The vanishing line of the ground plane $\mathbf{l}$ and the vertical vanishing point $\mathbf{v}$ and the top $(\mathbf{t}_1, \mathbf{t}_2)$ and base $(\mathbf{b}_1, \mathbf{b}_2)$ points of two line segments.
- **Compute:** The ratio of lengths of the line segments in the scene.



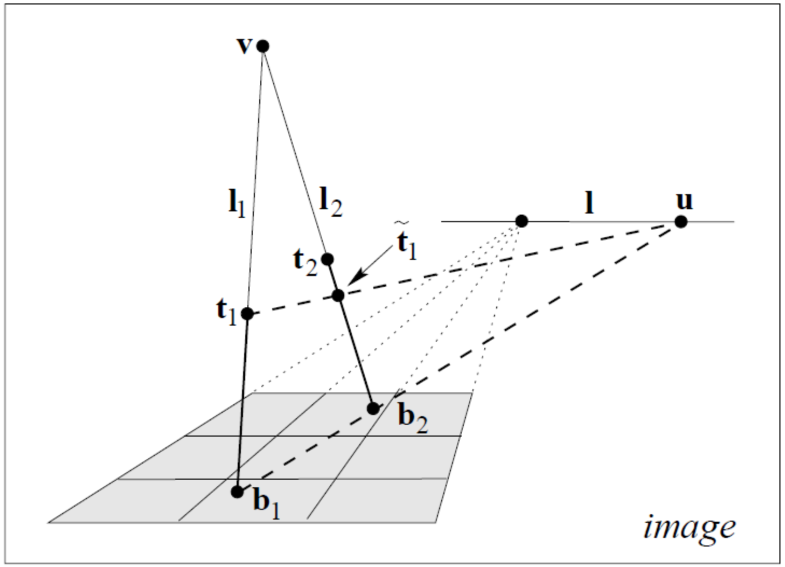


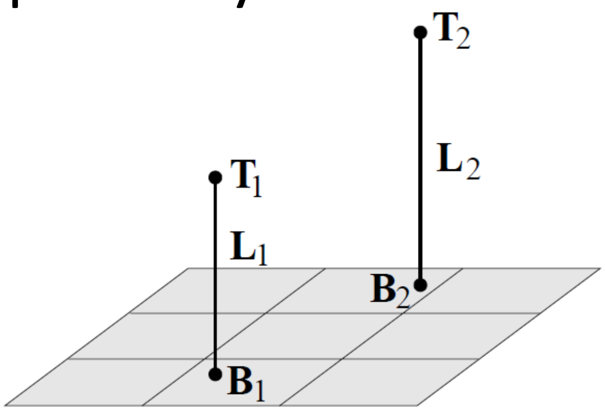
Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Affine 3D Measurements

1. Compute the vanishing point $\mathbf{u} = (\mathbf{b}_1 \times \mathbf{b}_2) \times \mathbf{l}$.

2. Compute the transferred point $\tilde{\mathbf{t}}_1 = (\mathbf{t}_1 \times \mathbf{u}) \times \mathbf{l}_2$, where $\mathbf{l}_2 = \mathbf{v} \times \mathbf{b}_2$.

3. Represent the four points $\mathbf{b}_2$, $\tilde{\mathbf{t}}_1$, $\mathbf{t}_2$ and $\mathbf{v}$ on the image line $\mathbf{l}_2$ by their distance from $\mathbf{b}_2$, as 0, $\tilde{t}_1$, t_2 and v, respectively.



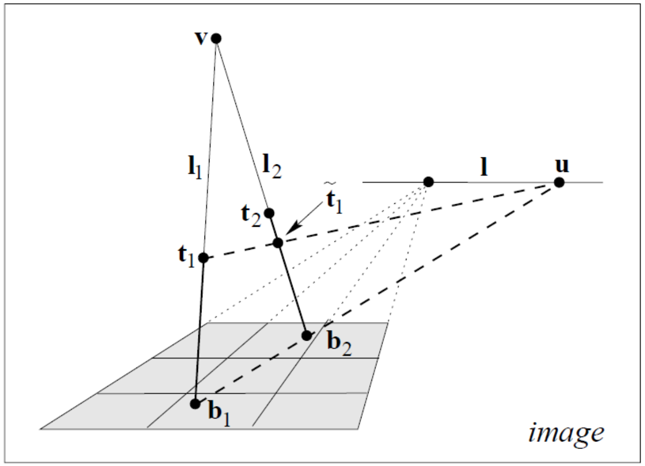


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Affine 3D Measurements

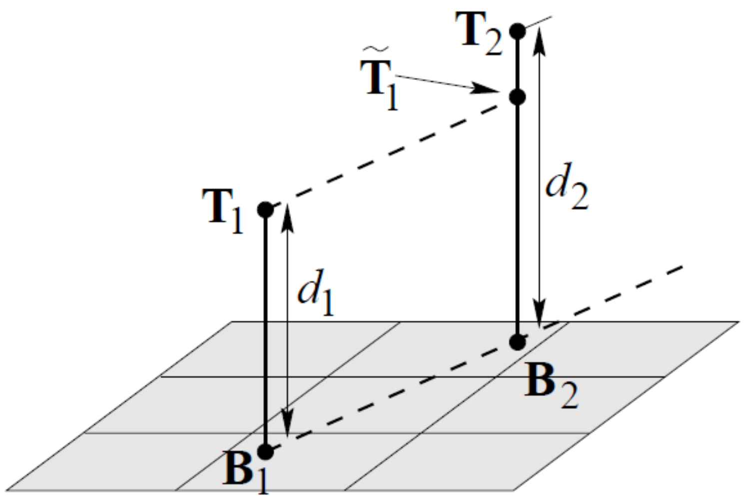
4. Compute a 1D projective transformation $\mathrm{H}_{2\times 2}$ mapping homogeneous coordinates $(0,1) \rightarrow (0,1)$ and $(\mathrm{v},1) \rightarrow (1,0)$ (which maps the vanishing point $\mathbf{v}$ to infinity).

   A suitable matrix is given by:

$$\mathrm{H}_{2\times 2} = \begin{bmatrix} 1 & 0 \\ 1 & -v \end{bmatrix}.$$

Affine 3D Measurements

- The (scaled) distance of the scene points $\tilde{\mathbf{T}}_1$ and $\mathbf{T}_2$ from $\mathbf{B}_2$ on $\mathbf{L}_2$ may then be obtained from the position of the points $\mathrm{H}_{2\times2}(\tilde{\mathrm{t}}_1, 1)^\top$ and $\mathrm{H}_{2\times2}(\mathrm{t}_2, 1)^\top$.
- Their distance ratio is then given by: $\dfrac{d_1}{d_2} = \dfrac{\tilde{t}_1(v - t_2)}{t_2(v - \tilde{t}_1)}$.



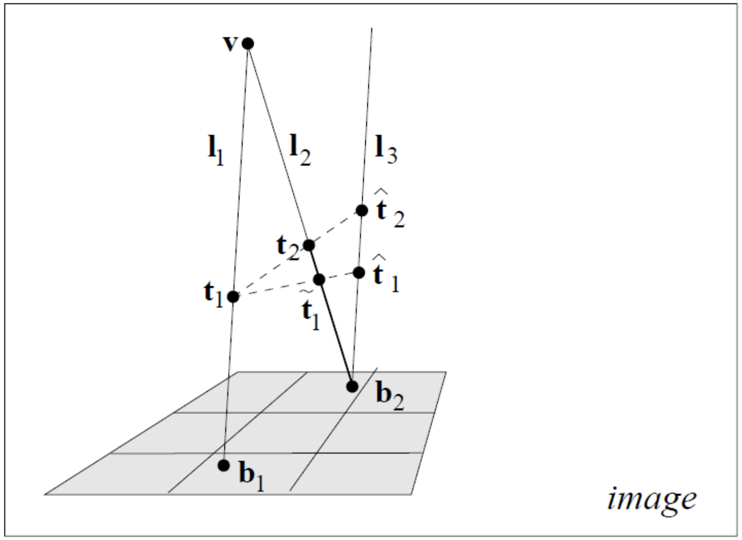


Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman

Height Measurements using Affine Properties

- Given the vanishing line of the ground plane $\mathbf{l}$ (cyan line) and the vertical vanishing point $\mathbf{v}$ (not shown).

- And using the known height of the filing cabinet, the absolute height of the two people are measured.

Image source: "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman