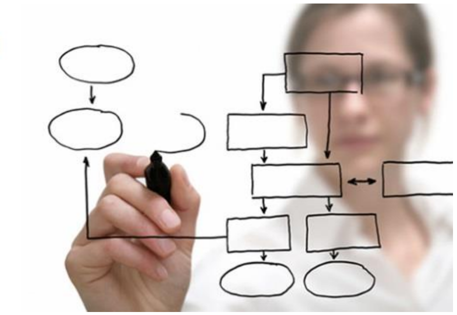
CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computing@CIT



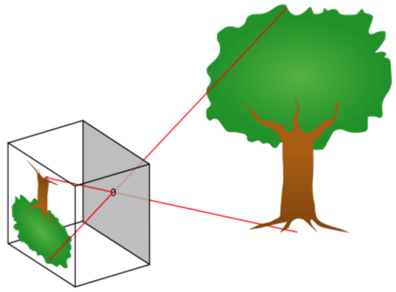
Machine Vision

Lecture 8: Multi view geometry

Central projection model

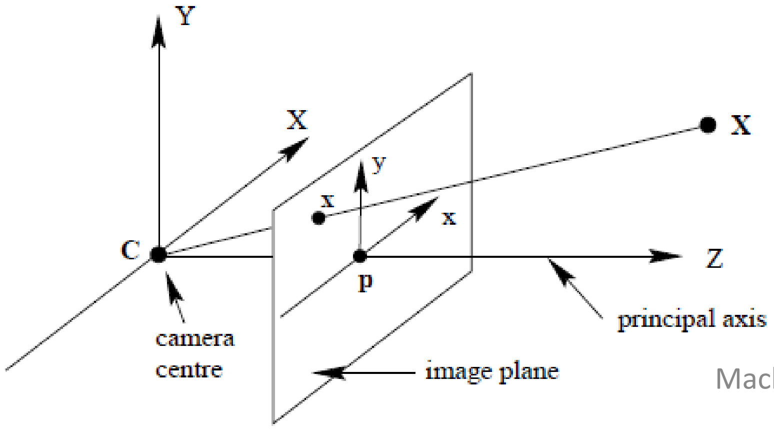
- We will first look at cameras with finite centre of projection
- A pinhole camera with focal length f located at the coordinate origin projects a 3d point

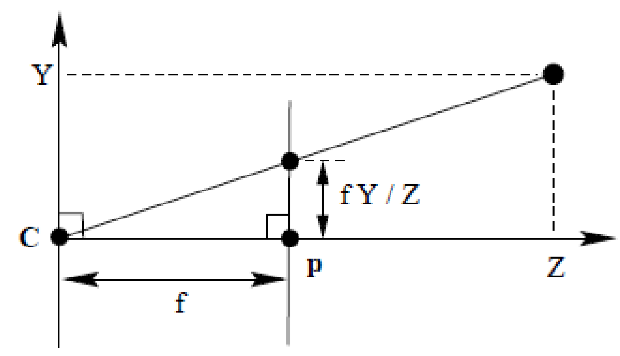
$$(X, Y, Z) \rightarrow \left(\frac{fX}{Z}, \frac{fY}{Z}\right)$$



- In homogeneous coordinates this can be expressed as

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

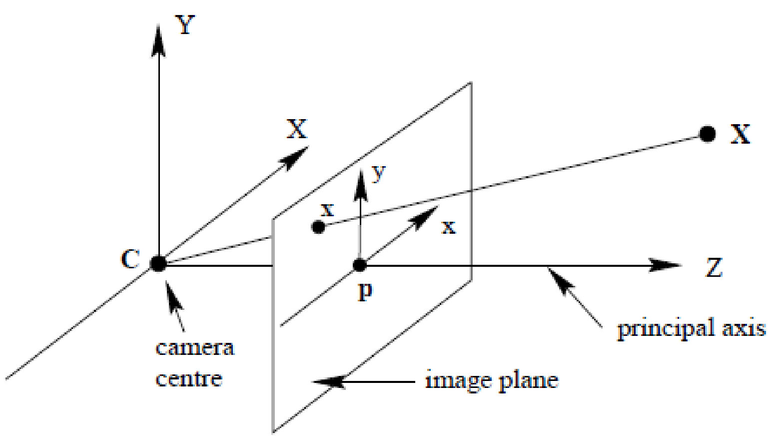


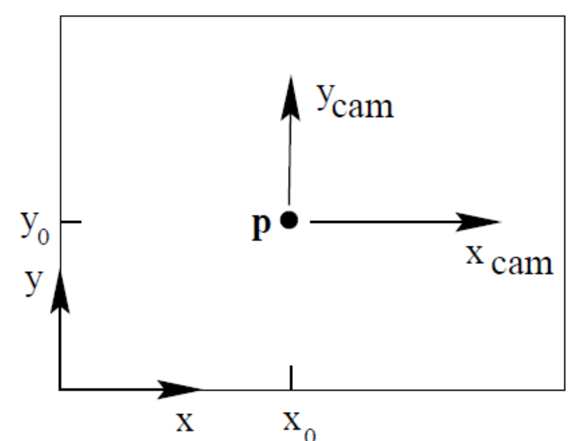


Camera calibration matrix

- We assumed that the coordinate system of the image plane is the same as the object coordinate system
- This is in general not the case, as our object world is typically measured in metres $[m^3]$ and the image world is measured in pixels $[pel^2]$
- Also the image coordinate system origin is not necessarily where the object Z-axis pierces the image plane, which we accommodate with translation by (x_0, y_o) and scaling by m_x and m_y of the image plane

$$\boldsymbol{x}' = \begin{pmatrix} fm_x & & x_0 & 0 \\ & fm_y & y_0 & 0 \\ & & 1 & 0 \end{pmatrix} \boldsymbol{X}$$





Camera calibration matrix

- We can summarise this image coordinate system transformation in the 3×3 camera calibration matrix

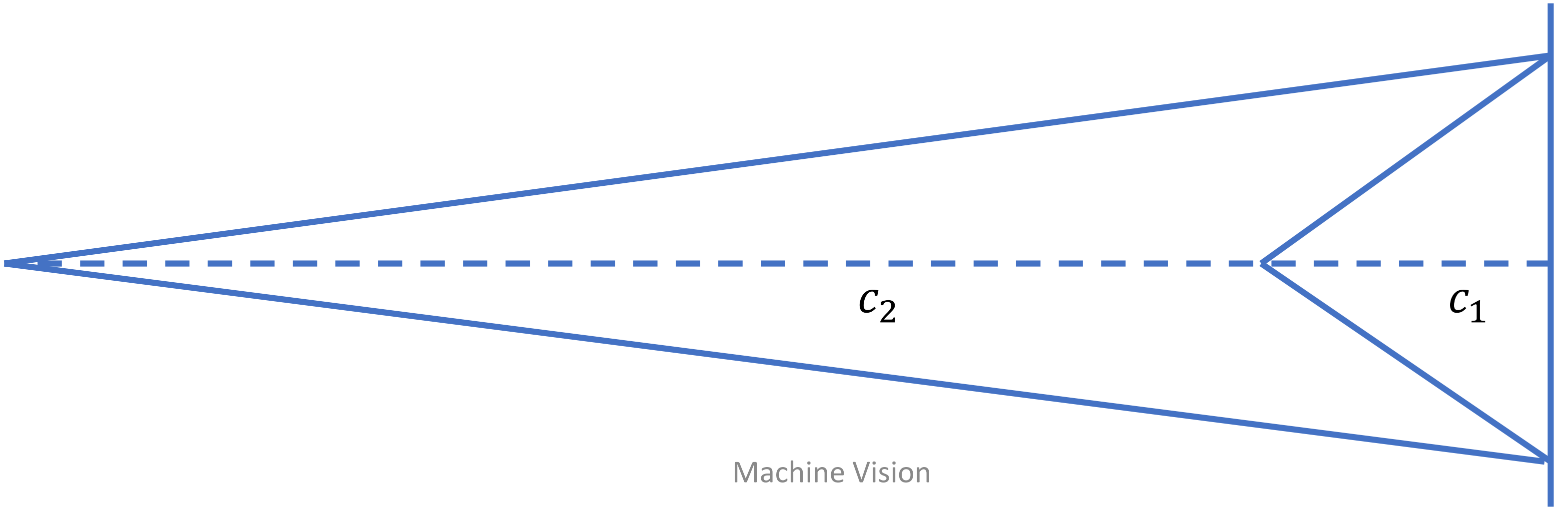
$$K = \begin{pmatrix} fm_x & s & x_0 \\ & fm_y & y_0 \\ & & 1 \end{pmatrix} = \begin{pmatrix} c & s & x_0 \\ & \alpha c & y_0 \\ & & 1 \end{pmatrix}$$

- Which has 5 parameters
  - The **principal length** $c = fm_x$
  - The **aspect ratio** $\alpha = m_y / m_x$
  - The **principal point** (x_0, y_0)
  - The **image skew** s
    (modern CCD cameras are manufactured with a very regular pixel grid, so typically $s = 0$ for digital images; however, $s \neq 0$ often happens when processing scanned film)

Changing the focal length (zoom)

- The <u>principal length</u> c is the distance between the projection centre and the image plane measured in the unit of the object coordinate system
- The longer the principal length, the narrower is the opening angle and the smaller is the field of view

- **Optical zoom** is changing distance of the image plane to the centre of projection, i.e. affecting the principal length c



Backward projection of rays

- The calibration matrix not only tells us how 3d points are projected into 2d coordinates
- We can also reverse the equation and obtain a way of calculating the direction in space corresponding to the image point

$$\boldsymbol{m} = \boldsymbol{K}^{-1}\boldsymbol{x}'$$

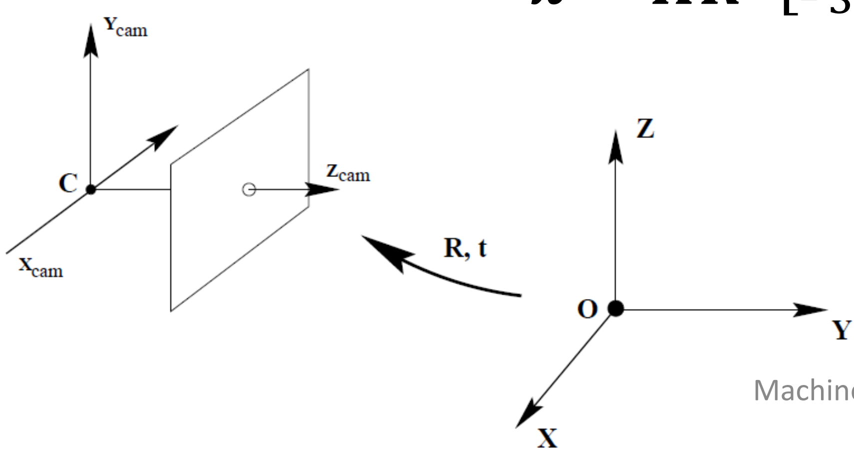
- The distance λ from the camera is unknown, but we know that the 3d scene point is somewhere on the line

$$\boldsymbol{X} = \lambda \frac{\boldsymbol{m}}{\sqrt{\boldsymbol{m}^T\boldsymbol{m}}}$$

Object coordinate system

- Thus far we have assumed the camera to be located in the origin of the 3d coordinate system
- Obviously this is not the case, and we need to accommodate this transformation between **camera coordinate system** and **world coordinate system**
- If we apply the translation and rotation to every image point we can express the full transformation from object to image coordinate system in homogeneous coordinates as follows

$$\boldsymbol{x} = \boldsymbol{K}\boldsymbol{R}^T[\boldsymbol{I}_{3\times3} \quad -\boldsymbol{t}]\boldsymbol{X}$$



The projective camera

- The projective camera can be described as a homogeneous 3×4 matrix $\boldsymbol{P} \in \mathbb{P}^{11}$ transforming homogeneous 3d scene points $\boldsymbol{X} \in \mathbb{P}^3$ into homogeneous 2d image coordinates $\boldsymbol{x} \in \mathbb{P}^2$ as

$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X}$$

- The projection matrix can be decomposed into

$$\boldsymbol{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} = \underbrace{\begin{pmatrix} c & s & x_0 \\ & \alpha c & y_0 \\ & & 1 \end{pmatrix}}_{\boldsymbol{K}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}}_{\boldsymbol{R}}^{T} \begin{pmatrix} 1 & & & -t_1 \\ & 1 & & -t_2 \\ & & 1 & -t_3 \end{pmatrix}$$

- The 11dof of $\boldsymbol{P}$ are distributed across the camera calibration $\boldsymbol{K}$ containing 5dof, the rotation of the camera $\boldsymbol{R}$ containing 3dof, and the position of the camera $\boldsymbol{t}$ containing 3dof

The projective camera

- To get the camera position $\boldsymbol{t}$ from the projection matrix we note that

$$\mathbf{P}\begin{pmatrix}\boldsymbol{t}\\1\end{pmatrix} = \boldsymbol{K}\boldsymbol{R}^T[\boldsymbol{I}_{3\times3} \quad -\boldsymbol{t}]\begin{pmatrix}\boldsymbol{t}\\1\end{pmatrix} = \boldsymbol{K}\boldsymbol{R}^T\begin{pmatrix}0\\0\\0\end{pmatrix} = \mathbf{0}$$

- Therefore the centre of projection is the right-null-space of **P**
- If we cut

$$\boldsymbol{P} = [\boldsymbol{M} \quad \boldsymbol{p}]$$

into a 3×3 submatrix

$$\boldsymbol{M} = \boldsymbol{K}\boldsymbol{R}^{\boldsymbol{T}}$$

and a 3-vector

$$\boldsymbol{p} = -\boldsymbol{K}\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{t}$$

Then we can easily calculate it using the following insight:

$$-\boldsymbol{M}^{-1}\boldsymbol{p} = \boldsymbol{R}\boldsymbol{K}^{-\boldsymbol{1}}\boldsymbol{K}\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{t} = \boldsymbol{t}$$

The projective camera

- To separate

$$\boldsymbol{M} = \boldsymbol{K}\boldsymbol{R}^T$$

- We note that this is a product of an upper diagonal matrix

$$\boldsymbol{K} = \begin{pmatrix} c & s & x_0 \\ & \alpha c & y_0 \\ & & 1 \end{pmatrix}$$

- and a rotation matrix $\boldsymbol{R}^T$, which we can separate using the RQ decomposition algorithm

```
K,Rt = scipy.linalg.rq(M)
```

General back-projection

- We already saw that the direction of the of a ray from the origin in the un-rotated coordinate frame is

$$\mathbf{m} = \mathbf{K}^{-1}\boldsymbol{x}'$$

- If we apply the coordinate system transformation, the ray originating from the centre of projection into this direction in the world coordinate frame is

$$\boldsymbol{X} = \boldsymbol{t} + \lambda \boldsymbol{R}\boldsymbol{K}^{-1}\boldsymbol{x}'$$

- Putting this all together we obtain a way to calculate the ray of 3d points corresponding to an image coordinate

$$\boldsymbol{X} = -\boldsymbol{M}^{-1}\boldsymbol{p} + \lambda \boldsymbol{M}^{-1}\boldsymbol{x}' = \boldsymbol{M}^{-1}(\lambda \boldsymbol{x} - \boldsymbol{p})$$

Triangulation of points

- Given two corresponding points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in two images

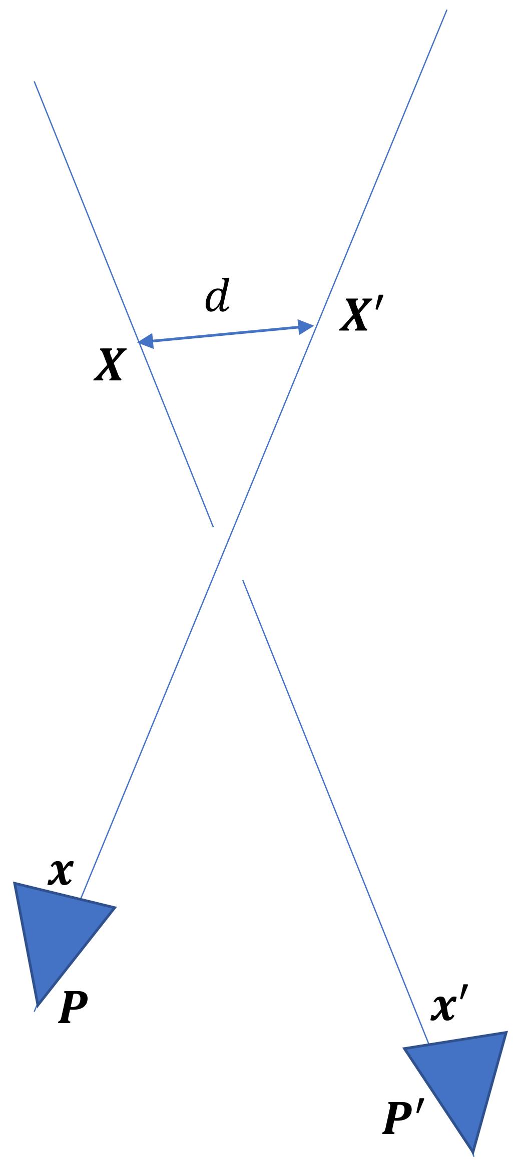
$$\boldsymbol{P} = (\boldsymbol{M} \quad \boldsymbol{p})$$
$$\boldsymbol{P}' = (\boldsymbol{M}' \quad \boldsymbol{p}')$$

- The 3d point must be on both rays

$$\boldsymbol{X} = \boldsymbol{M}^{-1}(\lambda \boldsymbol{x} - \boldsymbol{p})$$
$$\boldsymbol{X}' = \boldsymbol{M}'^{-1}(\mu \boldsymbol{x}' - \boldsymbol{p}')$$

- In the presence of noise, this is not exactly the case, therefore we try to find the point where the rays are closest, i.e. minimise the distance

$$d = \left|\boldsymbol{M}^{-1}(\lambda \boldsymbol{x} - \boldsymbol{p}) - \boldsymbol{M}'^{-1}(\mu \boldsymbol{x}' - \boldsymbol{p}')\right|^2$$



Triangulation of points

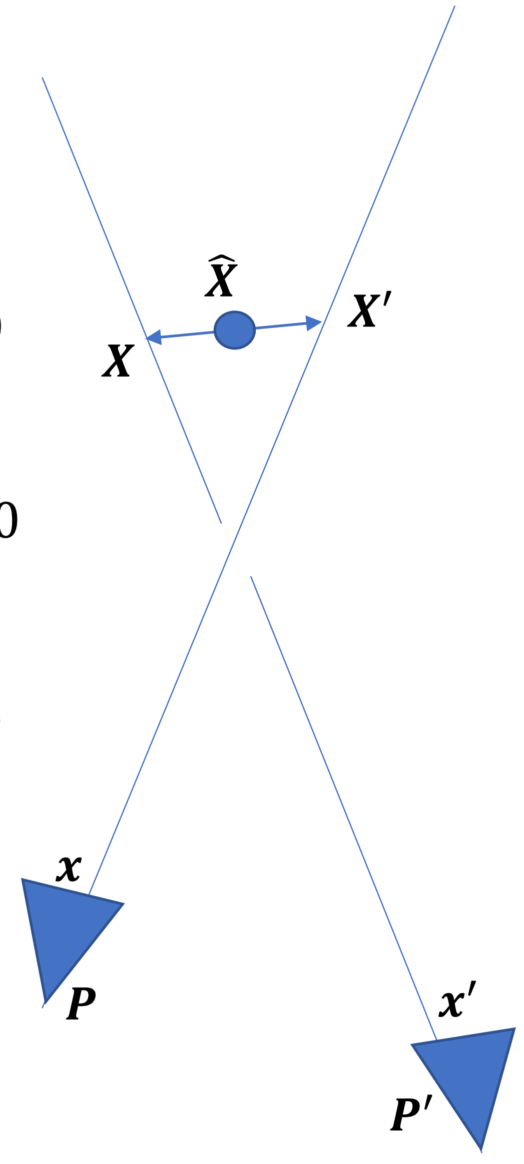
- To minimise the distance we look at the derivatives

$$\frac{\partial d}{\partial \lambda} = 2\left(\boldsymbol{M}^{-1}(\lambda \boldsymbol{x} - \boldsymbol{p}) - \boldsymbol{M}'^{-1}(\mu \boldsymbol{x}' - \boldsymbol{p}')\right)\boldsymbol{M}^{-1}\boldsymbol{x} = 0$$

$$\frac{\partial d}{\partial \mu} = 2\left(\boldsymbol{M}^{-1}(\lambda \boldsymbol{x} - \boldsymbol{p}) - \boldsymbol{M}'^{-1}(\mu \boldsymbol{x}' - \boldsymbol{p}')\right)\boldsymbol{M}'^{-1}\boldsymbol{x}' = 0$$

- This are two linear equations, which we can easily solve for the unknown λ and μ
- The triangulated point is then determined half-way between the two points closest to each other

$$\widehat{\boldsymbol{X}} = \frac{\boldsymbol{X} + \boldsymbol{X}'}{2}$$



Back projection of lines

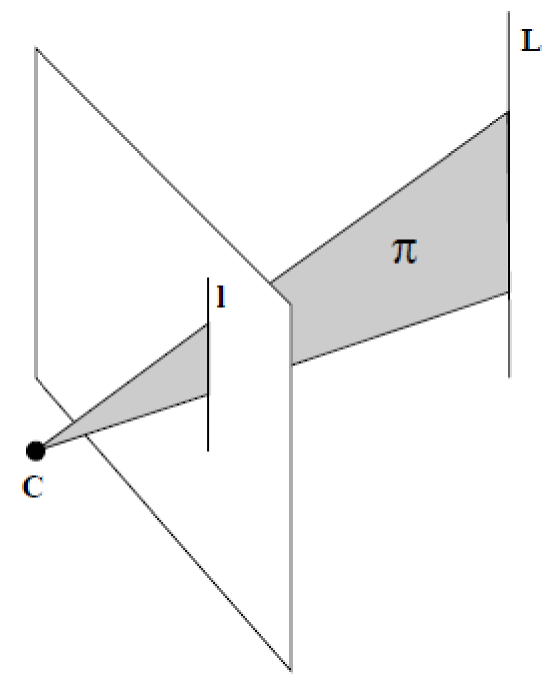
- A point $\boldsymbol{x}$ is on a line $\boldsymbol{l}$ if $\boldsymbol{l}^T\boldsymbol{x} = 0$
- All 3d points $\boldsymbol{X}$ that project somewhere on this line must fulfil

$$\boldsymbol{l}^T\boldsymbol{x} = \underbrace{\boldsymbol{l}^T\boldsymbol{P}}_{\boldsymbol{\pi}^T}\boldsymbol{X} = 0$$

- This can be considered as a plane equation of the 3d plane

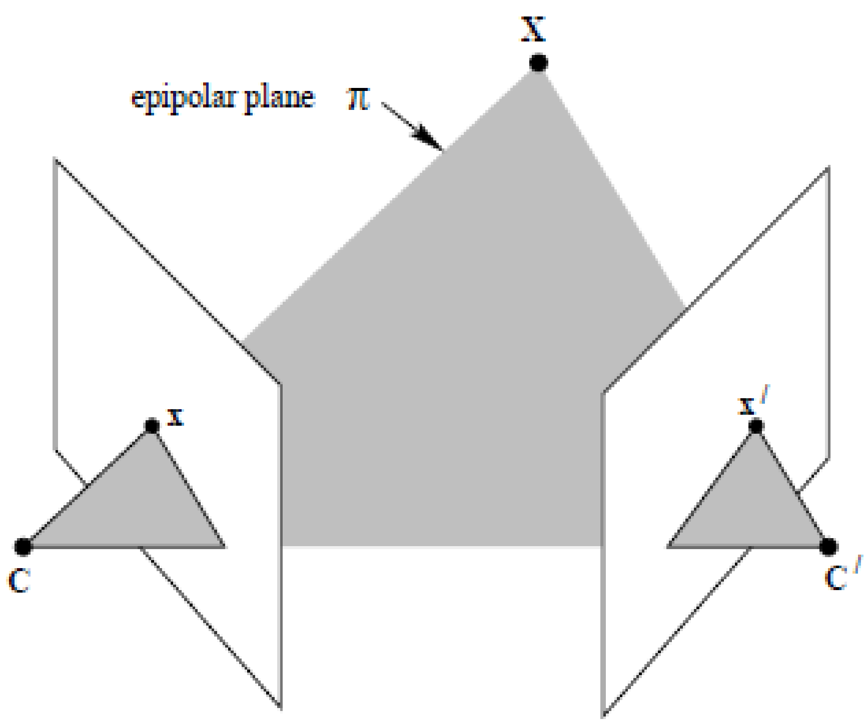
$$\boldsymbol{\pi} = \boldsymbol{P}^T\boldsymbol{l}$$

- Which is the back-projection of the image line $\boldsymbol{l}$



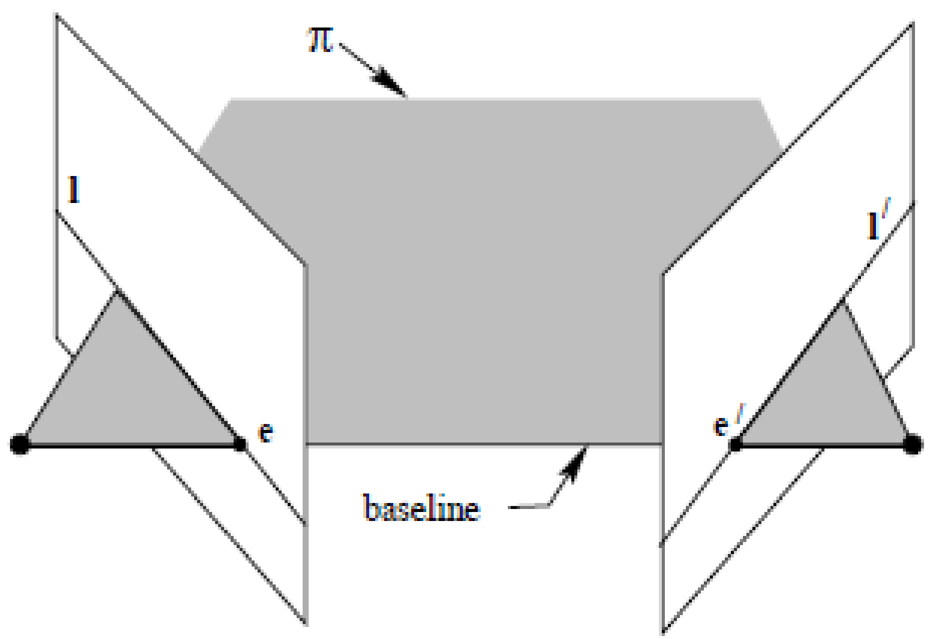
Epipolar geometry

- A single 3d point that is visible in two images defines an **epipolar plane** in 3d space, which connects the point and the two centres of projection



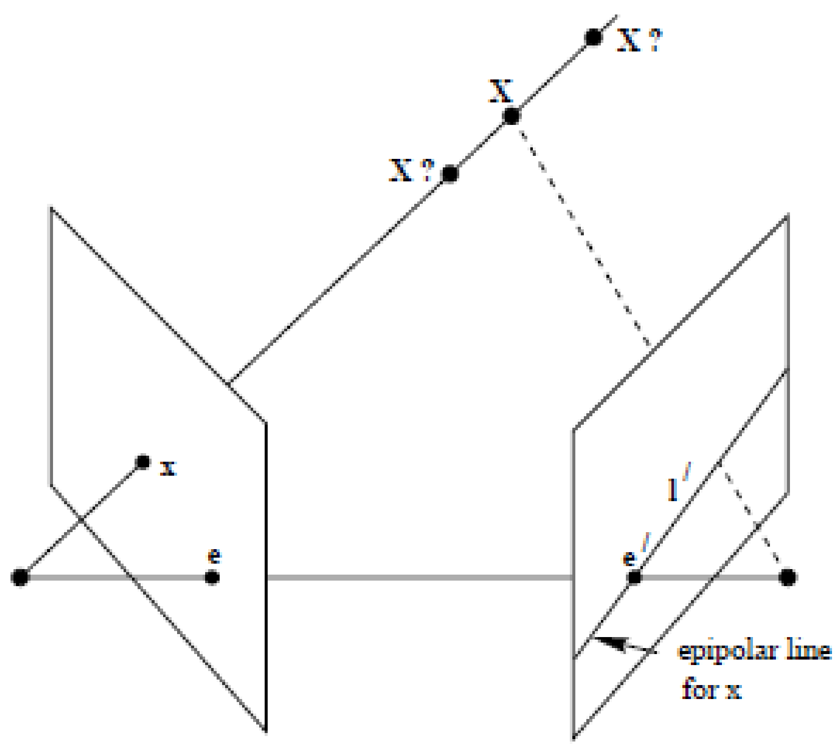
Epipolar geometry

- Because all these epipolar planes go through the centres of projection, and therefore through the **baseline** between the two images, they create corresponding **epipolar lines**
- All epipolar lines intersect in the **epipoles**, which are the intersections of the baseline with the image planes

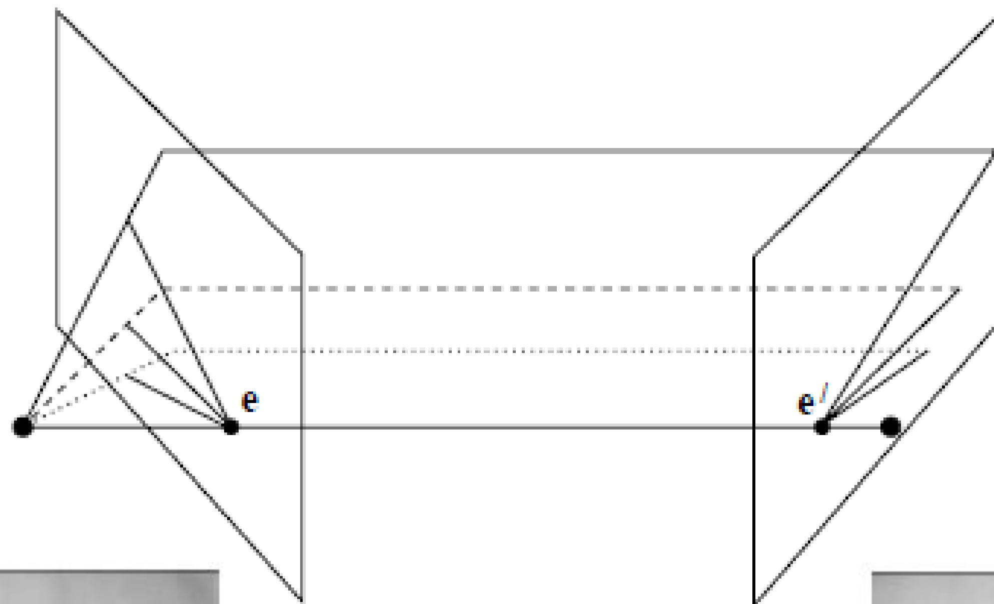


Epipolar geometry

- When looking for a point correspondence of a point $\boldsymbol{x}$ in another image, we don't know where it is due to the unknown distance
- However, knowing the epipolar geomery we can restrict the search to the **epipolar line** in the second image



Epipolar geometry



Fundamental matrix

- Let the two camera matrices be

$$\boldsymbol{P} = \boldsymbol{K}(\boldsymbol{I} \quad \boldsymbol{0})$$
$$\boldsymbol{P}' = \boldsymbol{K}'\boldsymbol{R}^T(\boldsymbol{I} \quad -\boldsymbol{t})$$

- Then the point $\boldsymbol{x}$ in the first image back-projects to the line

$$\boldsymbol{X} = \lambda \boldsymbol{K}^{-1}\boldsymbol{x}$$

- which projects into the second image at

$$\boldsymbol{x}' = \boldsymbol{K}'\boldsymbol{R}^T(\boldsymbol{X} - \boldsymbol{t}) = \boldsymbol{K}'\boldsymbol{R}^T(\lambda \boldsymbol{K}^{-1}\boldsymbol{x} - \boldsymbol{t})$$

- The epipole is the image of the centre of projection ($\lambda = 0$)

$$\boldsymbol{e}' = -\boldsymbol{K}'\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{t}$$

- Now the epipolar line through the epipole $\boldsymbol{e}$' and $\boldsymbol{x}$' is

$$\boldsymbol{l} = \boldsymbol{e}' \times \boldsymbol{x} = \boldsymbol{e}' \times (\lambda \boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{K}^{-1}\boldsymbol{x} + \boldsymbol{e}') = \boldsymbol{S}[\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{t}]\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{K}^{-1}\boldsymbol{x}$$

Fundamental matrix

- A point $\boldsymbol{x}$' in the second image is on the epipolar line

$$\boldsymbol{l} = \boldsymbol{S}[\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{t}]\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{K}^{-1}\boldsymbol{x}$$

- If

$$\boldsymbol{l}^T\boldsymbol{x}' = \boldsymbol{x}'^T\boldsymbol{F}\boldsymbol{x} = 0$$

- with the 3×3 **fundamental matrix**

$$\boldsymbol{F} = \boldsymbol{S}[\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{t}]\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{K}^{-1}$$

Fundamental matrix

- In conclusion, two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ can only refer to the same scene point if they obey the following equation

$$\boldsymbol{x}'^T \boldsymbol{F} \boldsymbol{x} = 0$$

- Obviously, this equation is homogeneous, i.e. the scale of $\boldsymbol{F}$ does not alter the result
- Also, because $\boldsymbol{S}[\boldsymbol{e}']$ has rank 2, the fundamental matrix is always singular

$$\det \boldsymbol{F} = 0$$

- These two condition mean that the fundamental matrix has 7 degrees of freedom

Fundamental matrix

- If $\boldsymbol{F}$ is the fundamental matrix of the image pair $(\boldsymbol{P}, \boldsymbol{P}')$, then $\boldsymbol{F}^T$ is the fundamental matrix of the image pair $(\boldsymbol{P}', \boldsymbol{P})$

- For a point $\boldsymbol{x}$ in the first image, the epipolar line in the second image is

$$\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x}$$

- For a point $\boldsymbol{x}'$ in the second image, the epipolar line in the first image is

$$\boldsymbol{l} = \boldsymbol{F}^T \boldsymbol{x}'$$

Calculating the epipoles

- The fundamental matrix is singular, and the epipoles are the left and right null-spaces

$$\boldsymbol{F}\boldsymbol{e} = \boldsymbol{0}$$
$$\boldsymbol{F}^{T}\boldsymbol{e}' = \boldsymbol{0}$$

- To calculate the epipoles we can use the singular value decomposition, with the epipole being the singular vector corresponding to the smallest singular value of $\boldsymbol{F}$

```
U,S,V = np.linalg.svd(F)
e = V[2,:]
```

Estimating the Fundamental matrix

- The fundamental matrix can be calculated from 7 point correspondences $\boldsymbol{x}_{\boldsymbol{i}}' \leftrightarrow \boldsymbol{x}_{\boldsymbol{i}}$

- Each point correspondence provides a condition

$$\boldsymbol{x}_{\boldsymbol{i}}'^{T} \boldsymbol{F} \boldsymbol{x}_{\boldsymbol{i}} = 0$$

- Or equivalent using the Kronecker product

$$\underbrace{\left(\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{T}} \otimes \boldsymbol{x}_{\boldsymbol{i}}'^{\boldsymbol{T}}\right)}_{\boldsymbol{a}_{\boldsymbol{i}}^{\boldsymbol{T}}} vec[\boldsymbol{F}] = 0$$

Estimating the Fundamental matrix

- These 7 equations can be stacked into a 7×9 matrix

$$A = \begin{pmatrix} \boldsymbol{a}_1^T \\ \vdots \\ \boldsymbol{a}_7^T \end{pmatrix}$$

- For which we calculate the two null-vectors $\boldsymbol{A}\boldsymbol{f}_1 = \mathbf{0}$ and $\boldsymbol{A}\boldsymbol{f}_2 = \mathbf{0}$ using the singular value decomposition

```
U,S,V = np.linalg.svd(A)
F1 = V[8,:].reshape(3,3).T
F2 = V[7,:].reshape(3,3).T
```

Estimating the Fundamental matrix

- The fundamental matrix we are looking for is now

$$\boldsymbol{F} = \alpha \boldsymbol{F_1} + (1-\alpha)\boldsymbol{F_2}$$

- To determine the value for alpha we use the singularity constraint

$$\det(\alpha \boldsymbol{F_1} + (1-\alpha)\boldsymbol{F_2}) = 0$$

- This is a degree 3 polynomial in the unknown α, which we can easily solve

Estimating the Fundamental matrix

- In case there are 8 points or more, we can also apply the DLT algorithm we have seen before by stacking all points into

$$A = \begin{pmatrix} \boldsymbol{a}_{\mathbf{1}}^T \\ \vdots \\ \boldsymbol{a}_{\mathbf{8}}^T \end{pmatrix}$$

- The fundamental matrix is then found as the singular vector corresponding to the smallest singular value
- In this case the singularity constraint needs to be applied (again using the singular value decomposition)

```
U,S,V = np.linalg.svd(A)
F = V[8,:].reshape(3,3).T
U,S,V = np.linalg.svd(F)
F = np.matmul(U,np.matmul(np.diag([S[0],S[1],0]),V))
```

Projective invariance

- The fundamental matrix for a pair of cameras $(\boldsymbol{P}, \boldsymbol{P}')$ and a pair of cameras $(\boldsymbol{PH}, \boldsymbol{P}'\boldsymbol{H})$ is the same for all 3d homograpgies $\boldsymbol{H}$
- Therefore, the knowing the fundamental matrix determines the 3d scene up to a 3d projective transformation only

- If necessary we can choose the canonical cameras

$$\boldsymbol{P} = [\boldsymbol{I} \quad \boldsymbol{0}]$$
$$\boldsymbol{P}' = [\boldsymbol{S}[\boldsymbol{e}']\boldsymbol{F} \quad \boldsymbol{e}']$$

- And determine the necessary homography later from other information (camera calibration)

Image rectification

- If apply any 2d homographies $\boldsymbol{H}$ and $\boldsymbol{H}'$ to both images

$$\hat{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{x}$$
$$\hat{\boldsymbol{x}}' = \boldsymbol{H}'\boldsymbol{x}'$$

- then the fundamental matrix between the two images transforms according to

$$\hat{\boldsymbol{F}} = \boldsymbol{H}'^{-T}\boldsymbol{F}\boldsymbol{H}^{-1}$$

- This is particularly useful, if we want to transform the images to achieve a given target fundamental matrix $\hat{\boldsymbol{F}}$

Image rectification

- An important special case is related to how our two eyes are arranged:
  - Both eyes are identical: $\boldsymbol{K} = \boldsymbol{K}'$
  - Both eyes look into the same direction: $\boldsymbol{R} = \boldsymbol{I}$
  - The translation is horizontal only: $\boldsymbol{t} = \begin{pmatrix} b & 0 & 0 \end{pmatrix}^T$

- In this case the fundamental matrix is

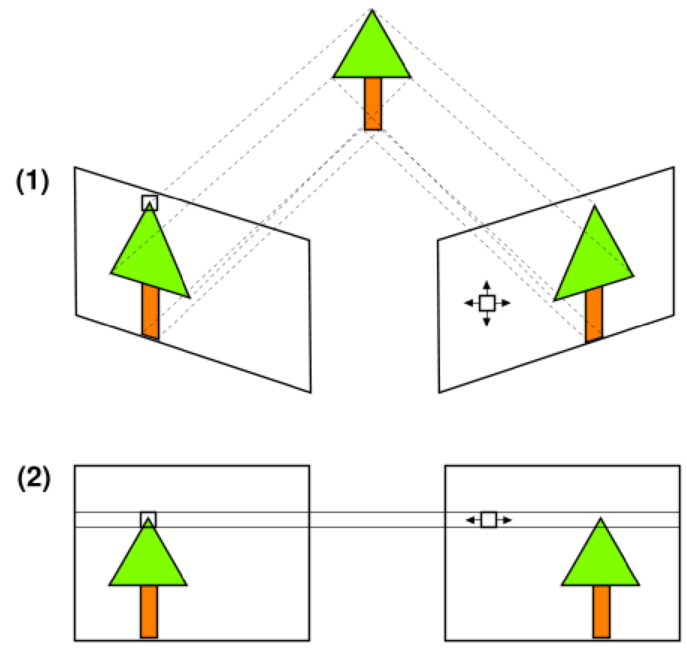
$$\boldsymbol{F} = \boldsymbol{S}[\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{t}]\boldsymbol{K}'\boldsymbol{R}^T\boldsymbol{K}^{-1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -cb \\ 0 & cb & 0 \end{pmatrix}$$

Image rectification

- To achieve this special configuration we therefore need to find homographies so that

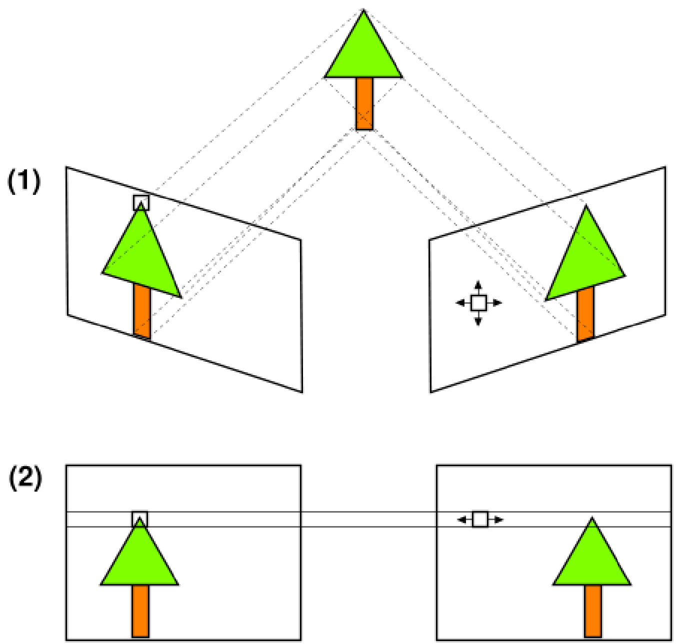
$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} = \boldsymbol{H'}^{-T}\boldsymbol{F}\boldsymbol{H}^{-1}$$

- There are many homographies $\boldsymbol{H}$ and $\boldsymbol{H}'$ that fulfil these equations
- Typically we will choose these transformations so that they minimally distort the original input images
- We also make sure that corresponding epipolar lines align



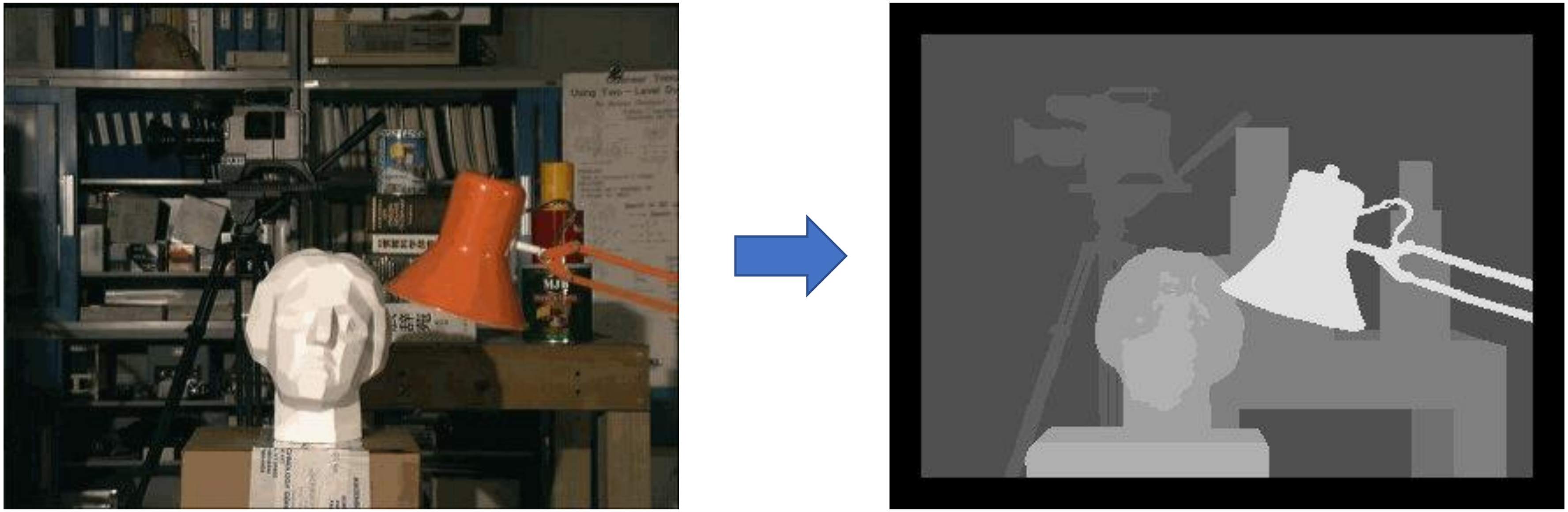
Disparity

- Corresponding epipolar lines are aligned in a rectified image, therefore the depth of a 3d point only affects the horizontal displacement between the images
- The matching problem for a rectified image pair is therefore 1-dimensional
- This horizontal displacement between the images is called **disparity**



Dense stereo

- Algorithms that solve this 1d search problem and calculate the disparity, and therefore the depth, for each pixel are called **dense stereo** algorithms



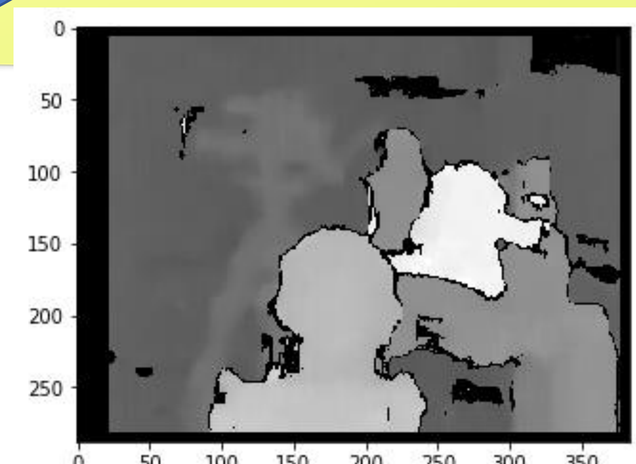
Dense stereo

- There are several dense stereo algorithms for rectified images, most of them combining smoothness constraints and similarity measures

Block Matching compares patches to calculate similarity metric

```
stereo = cv2.createStereoBM(numDisparities=16, blockSize=15)
disparity = stereo.compute(imgL,imgR)
```

The disparity is linked to depth, therefore the search range can be limited

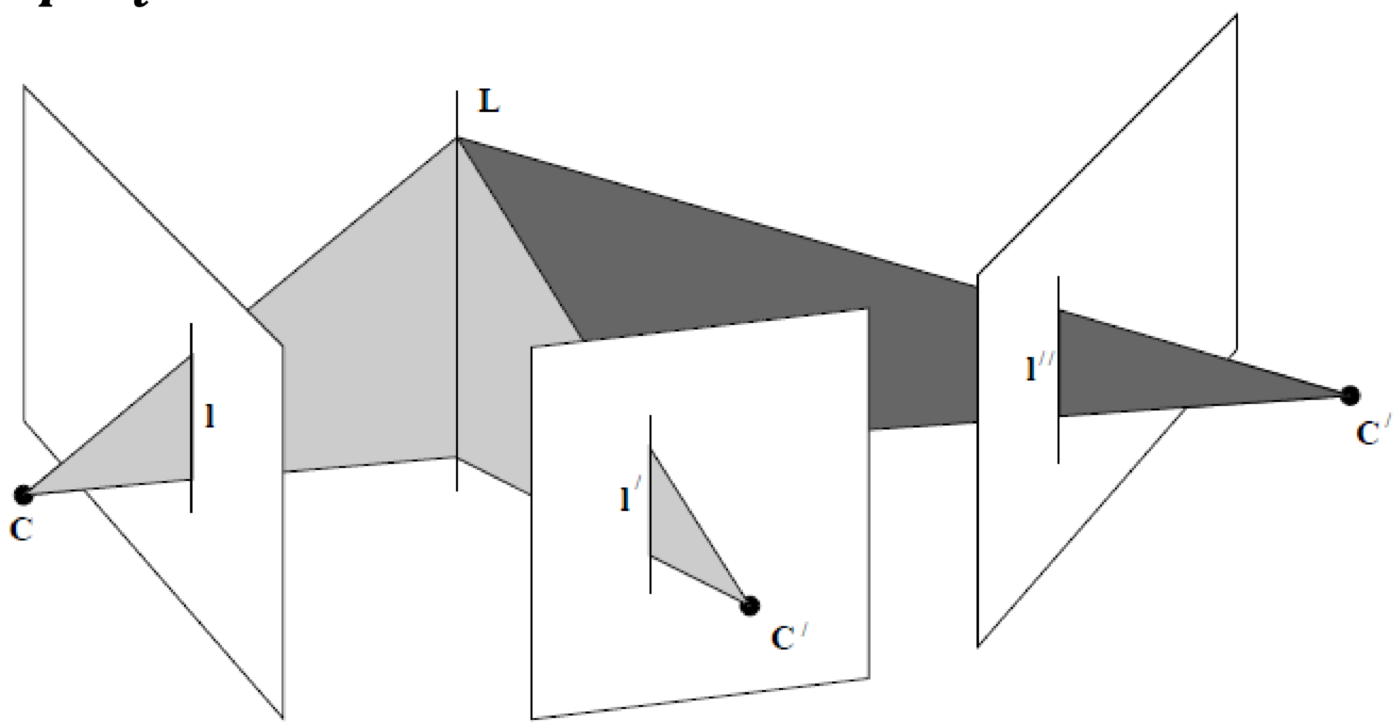


Stereoscopic images

- Another application of rectified images is to present them to each eye individually
- Because this disparity estimation is how our spatial perception works, humans will perceive the scene in 3d then

Trifocal tensor

- We now will look very briefly at the geometry of three images
- Three lines $\boldsymbol{l}$, $\boldsymbol{l}'$, $\mathbf{l}''$ must all back-project onto a single line $\boldsymbol{L}$ in space
- These back-projections are the three planes $\boldsymbol{\pi} = \boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{l}$, $\boldsymbol{\pi}' = \boldsymbol{P}'^{\boldsymbol{T}}\boldsymbol{l}'$ and $\boldsymbol{\pi}'' = \boldsymbol{P}''^{\boldsymbol{T}}\boldsymbol{l}''$



Trifocal tensor

- We now will look briefly at the geometry of three images
- Three lines $\boldsymbol{l}$, $\boldsymbol{l}'$, $\mathbf{l}''$ must all back-project onto a single line $\boldsymbol{L}$ in space
- These back-projections are the three planes $\boldsymbol{\pi} = \boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{l}$, $\boldsymbol{\pi}' = \boldsymbol{P}'^{\boldsymbol{T}}\boldsymbol{l}'$ and $\boldsymbol{\pi}'' = \boldsymbol{P}''^{\boldsymbol{T}}\boldsymbol{l}''$
- All points $\boldsymbol{X}$ on the line $\boldsymbol{L}$ must therefore be incident to all three lines, i.e.

$$\begin{pmatrix} \boldsymbol{\pi}^T \\ \boldsymbol{\pi}'^T \\ \boldsymbol{\pi}''^T \end{pmatrix} \boldsymbol{X} = \boldsymbol{0}$$

- Because the line $\boldsymbol{L}$ is a one-dimensional entity (in addition to the homogeneity of the equation), the null-space of this matrix must be 2-dimensional
- This is called the tri-focal constraint

Trifocal tensor

- The tri-focal constraint can be expressed as stating that the line

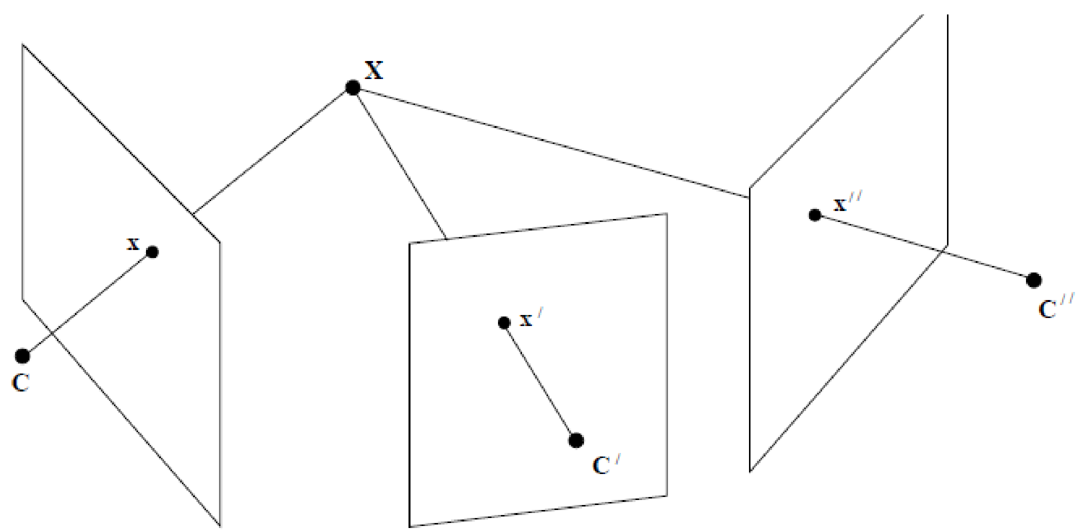
$$\boldsymbol{l} = \begin{pmatrix} \boldsymbol{l}'^T \boldsymbol{\mathcal{T}}_{\mathbf{1}} \boldsymbol{l}'' \\ \boldsymbol{l}'^T \boldsymbol{\mathcal{T}}_{\mathbf{2}} \boldsymbol{l}'' \\ \boldsymbol{l}'^T \boldsymbol{\mathcal{T}}_{\mathbf{3}} \boldsymbol{l}'' \end{pmatrix}$$

- The $3 \times 3 \times 3$ tensor $[\boldsymbol{\mathcal{T}}_{\mathbf{1}}, \boldsymbol{\mathcal{T}}_{\mathbf{2}}, \boldsymbol{\mathcal{T}}_{\mathbf{3}}]$ describing this relationship is called the **tri-focal tensor**

Tri-focal geometry

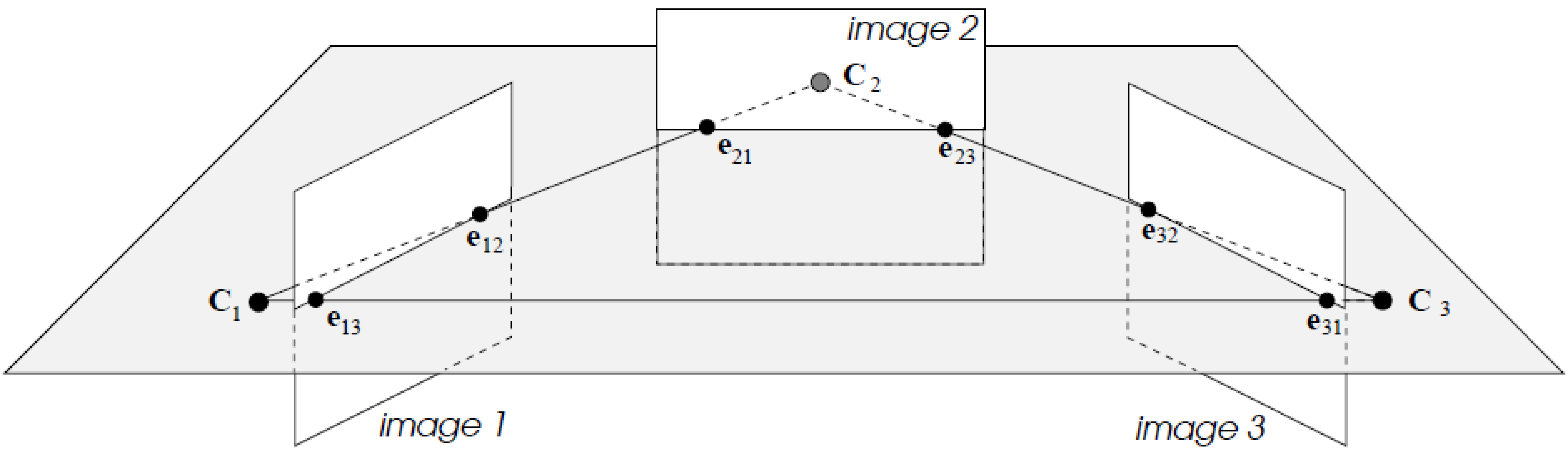
- Why is this important? Why can't we just use the pair-wise relationships provided by the epipolar geometry of mutual pairs of images?
- If we, for example want to transfer a point correspondence $\boldsymbol{x} \leftrightarrow \boldsymbol{x}'$ from one image pair into a third image, we could simply calculate the intersection of the epipolar lines in that image, i.e.

$$\boldsymbol{x}'' = (\boldsymbol{F}_{31}\boldsymbol{x}) \times (\boldsymbol{F}_{32}\boldsymbol{x}')$$



Tri-focal geometry

- Why is this important? Why can't we just use the pair-wise relationships provided by the epipolar geometry of mutual pairs of images?
- Unfortunately, this point transfer via epipolar lines does not work in the **tri-focal plane** connecting all three projection centres
- Point transfer via the tri-focal tensor is possible, though



Thank you for your attention!