Today

Early vision on multiple images

- Epipolar geometry
- Estimating depth

Project 2

Project 2 has been posted in Blackboard. You can access the data through Blackboard.

Due 11:59pm EST, Sunday, April 16

Teamwork for 2 team members.

If you have a preferred partner, please send me an email including the team members' names by **Friday, March 31**. Otherwise, you will be assigned to a team randomly. The team assignment will be posted in Blackboard by Sunday, April 2nd.

Passive Stereo

General strategy:

- **Camera calibration**
- **Correspondence**
- **3D reconstruction from matched 2D points**
    - Image rectification
    - 3D triangulation from rectified images

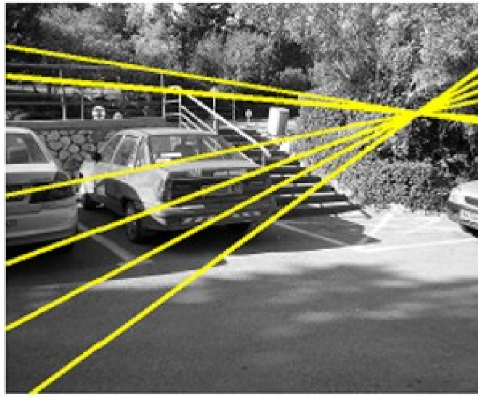
Epipolar Geometry

Epipolar constraint: For a 2D image point U_l in the left image, its corresponding point in the right image must lie on the conjugate epipolar line.

Epiploar line is the projection of the 3D line, which passes through U_l and the left camera center, on the right image.

All epipolar lines go through an epipole on one image.

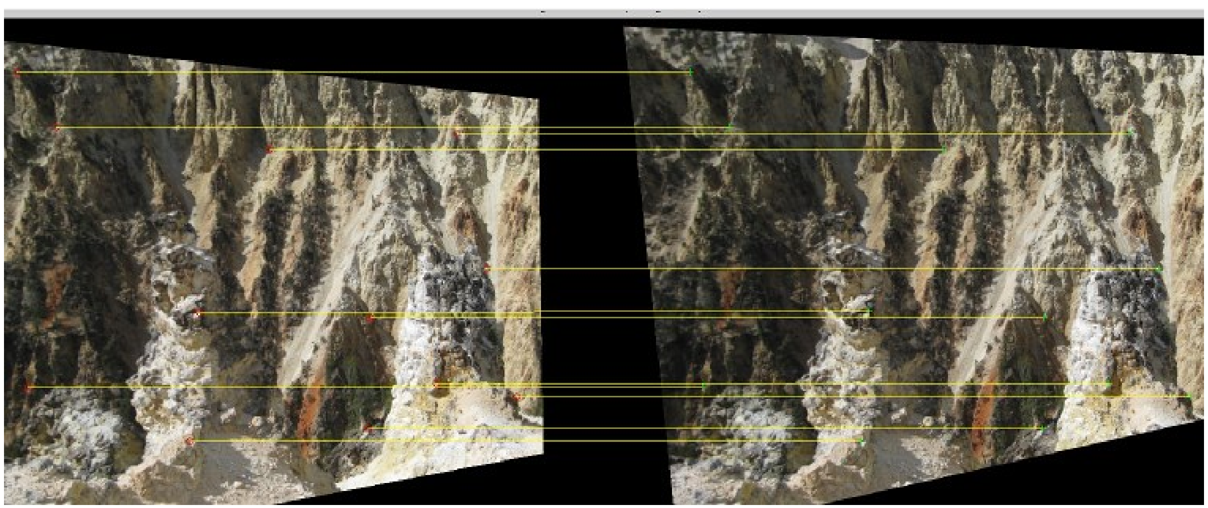
The epipole on the left (right) image is the projection of the optical center of the right (left) image on the left (right) image plane.



Epipolar Geometry in Rectified Images

In rectified images,

- the epipolar lines are parallel;
- conjugate epipolar lines are collinear and parallel to the base line.



Epipolar Geometry

P: a 3D point

$\boldsymbol{P_l}$: the coordinate of P on the left camera coordinate system

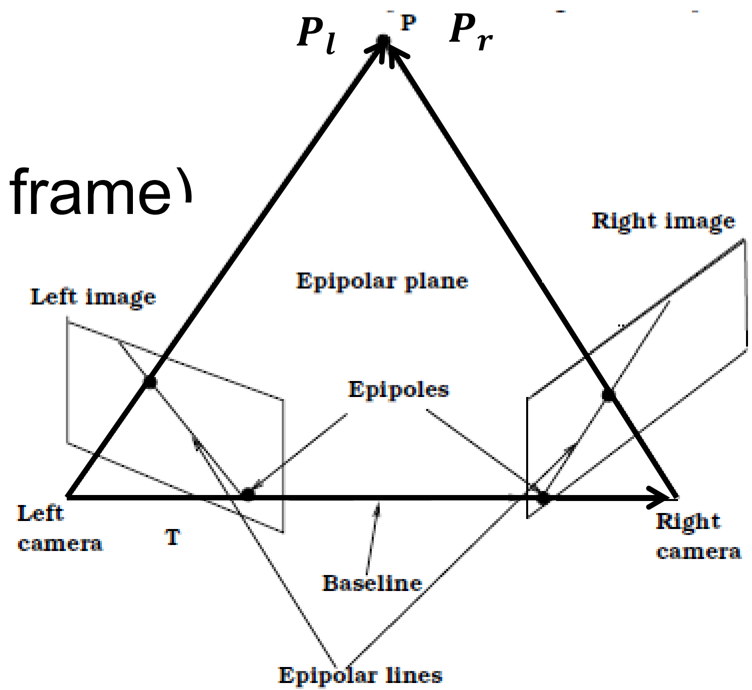
$\boldsymbol{P_r}$: the coordinate of P on the right camera coordinate system

R: the relative rotation between two

camera frames (from right frame to left frame)

$$\boldsymbol{P}_l = \mathrm{R}\boldsymbol{P}_r + \boldsymbol{T}$$

$$\boldsymbol{P}_r = \mathrm{R}^t(\boldsymbol{P}_l - \boldsymbol{T}\ )$$



Epipolar Geometry

$\boldsymbol{P_l}$, $\boldsymbol{P_l - T}$ **and T** are coplanar → in the same epipolar plane

$$(\boldsymbol{T} \times \boldsymbol{P_l})^t (\boldsymbol{P_l} - \boldsymbol{T}) = \boldsymbol{0}$$

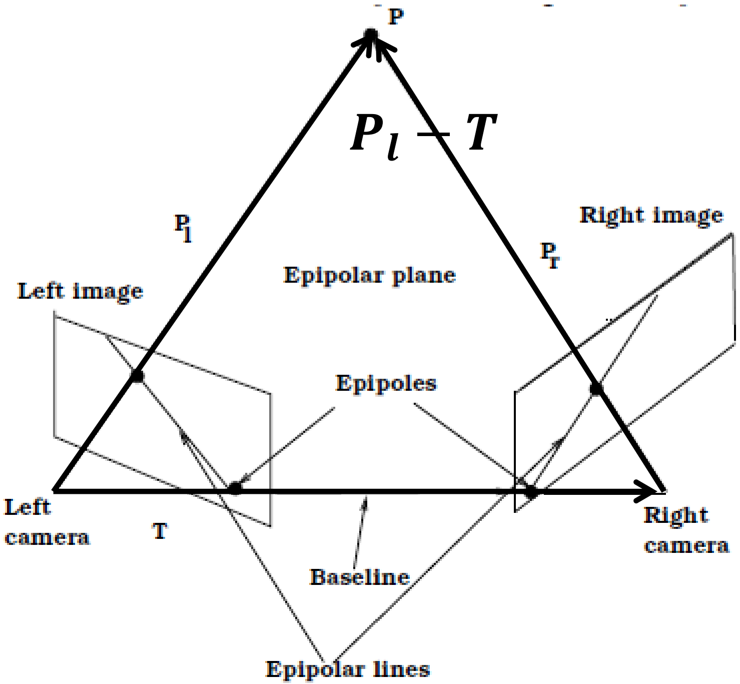
Normal of the plane

$$(\boldsymbol{T} \times \boldsymbol{P_l})^t \mathbf{R} \boldsymbol{P_r} = \boldsymbol{0}$$

Let

$$\mathbf{S} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

$$\boldsymbol{T} \times \boldsymbol{P_l} = \boldsymbol{S P_l}$$



Epipolar Geometry

$$(\mathbf{S}\boldsymbol{P}_l)^t\mathbf{R}\boldsymbol{P}_r = \mathbf{0} \leftrightarrow \boldsymbol{P}_l^t\,\mathbf{S}^t\mathbf{R}\boldsymbol{P}_r = 0$$

Let $\mathbf{E} = \mathbf{S}^t\mathbf{R}$

$$\boldsymbol{P}_l^t\,\mathbf{E}\boldsymbol{P}_r = \mathbf{0}$$

$\mathbf{E}$ is called the **essential matrix**

Since $\text{rank}(\mathbf{S}) = 2$, $\text{rank}(\mathbf{R}) = 3$ ⇨ $\text{rank}(\mathbf{E}) = 2$

The essential matrix

- establishes a link between the epipolar constraint and the relative position of the two camera systems
- has 5 degree of freedom

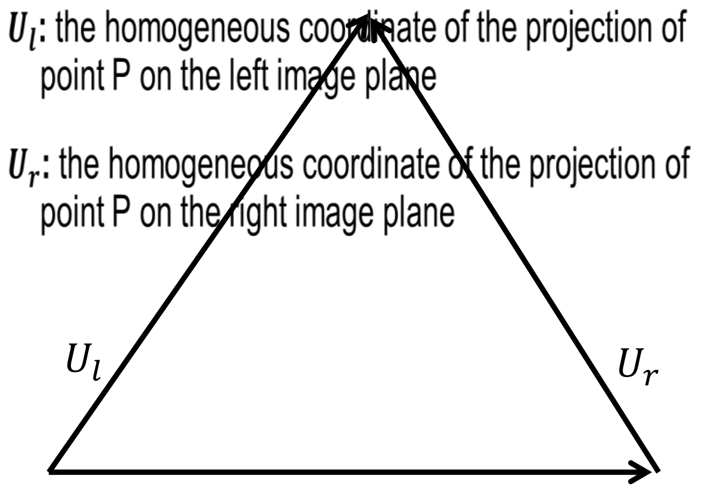
Fundamental Matrix

$\boldsymbol{U_l}$: the homogeneous coordinate of the projection of point P on the left image plane

$\boldsymbol{U_r}$: the homogeneous coordinate of the projection of point P on the right image plane

$$\lambda_l U_l = \mathbf{W}_l P_l \qquad \lambda_r U_r = \mathbf{W}_r P_r$$

Intrinsic parameters for left and right cameras



Substitute to $P_l^t \mathbf{E} P_r = 0$

$$U_l^t \mathbf{W}_l^{-t} \mathbf{E} \mathbf{W}_r^{-1} U_r = 0$$

Fundamental Matrix

Fundamental matrix

$$\mathbf{F} = \mathbf{W_l}^{-t}\mathbf{E}\mathbf{W_r}^{-1}$$

$$\boldsymbol{U_l^t}\mathbf{F}\boldsymbol{U_r} = \mathbf{0}$$

A pair of corresponding image points are related by **F**!

A fundamental matrix encodes

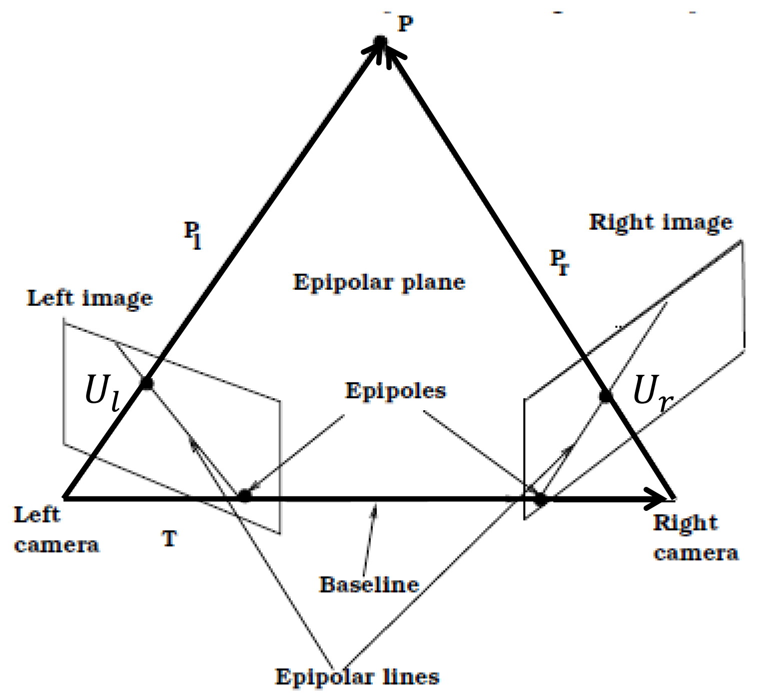
- the relative orientation and translation

between two camera systems

- the intrinsic parameters of the cameras

What is the rank of **F**?

rank(**F**)=2 because of **E**



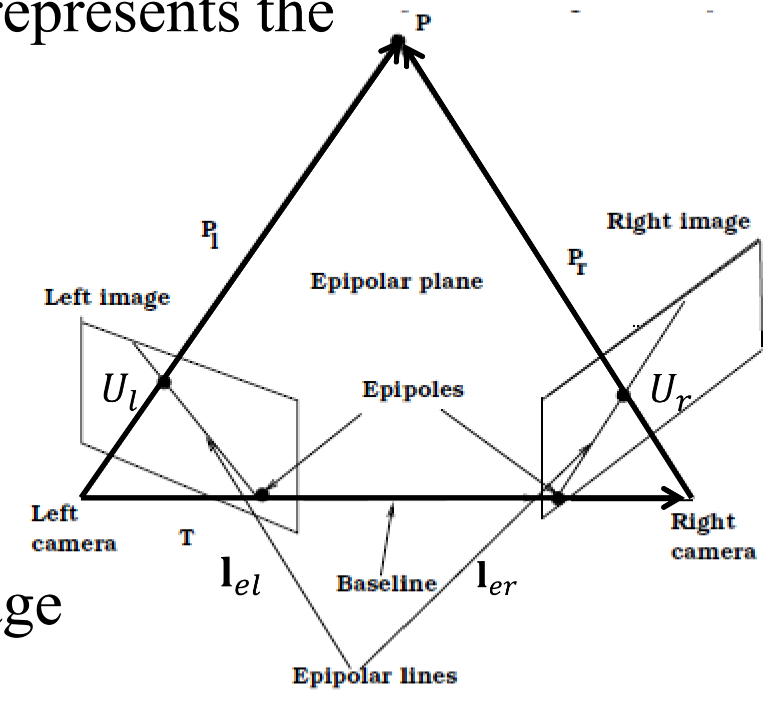
Information from Fundamental Matrix

$$\boldsymbol{U}_l^t \mathbf{F} \boldsymbol{U}_r = \mathbf{0}$$

Epipolar constraint: $\boldsymbol{U}_l^t(\mathbf{F}\boldsymbol{U}_r) = \mathbf{0}$ and $(\mathbf{F}^t \boldsymbol{U}_l)^t \ \boldsymbol{U}_r = \mathbf{0}$

Recall line function: $\mathbf{l} \cdot \mathbf{p} = \mathbf{0}$, where $\mathbf{l}$ represents the line parameters

$\mathbf{l}_{el} = \mathbf{F}\boldsymbol{U}_r$



$\boldsymbol{U}_l$ is on a line determined by $\mathbf{l}_{el}$

- Epipolar line on the left image
- Corresponding to $\boldsymbol{U}_r$ on the right image

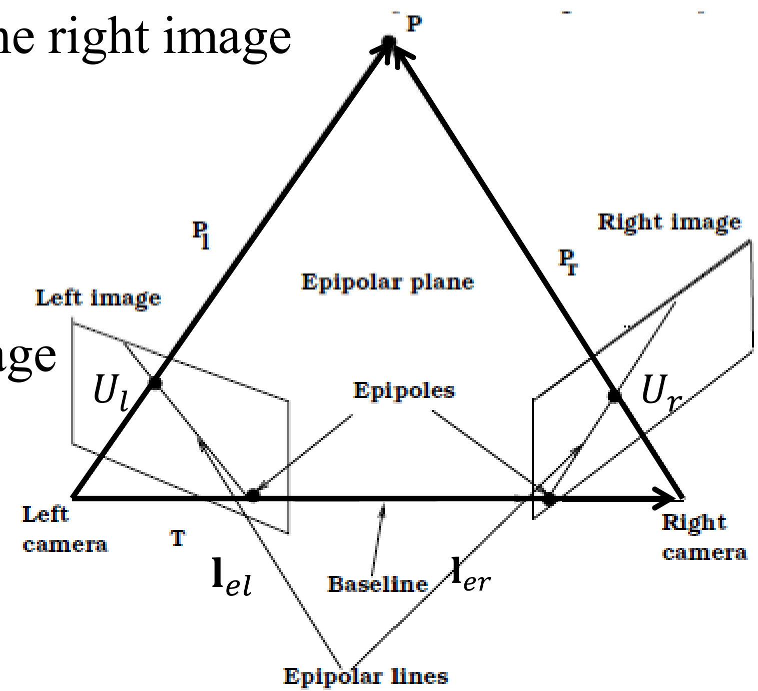
Information from Fundamental Matrix

$$U_l^t \mathbf{F} U_r = 0$$

- Epipolar constraint: $U_l^t(\mathbf{F}U_r) = 0$ and $(\mathbf{F}^t U_l)^t\ U_r = 0$
- $\mathbf{l}_{el} = \mathbf{F}U_r$ Epipolar line on the left image corresponding to U_r on the right image

Similarly

$\mathbf{l}_{er}=\mathbf{F}^t U_l$ Epipolar line on the right image corresponding to U_l on the left image



Given F and a point on one image, we can estimate the corresponding epipolar line

Information from Fundamental Matrix

$$U_l^t \mathbf{F} U_r = \mathbf{0}$$

Since all epipolar line pass through the epipole,

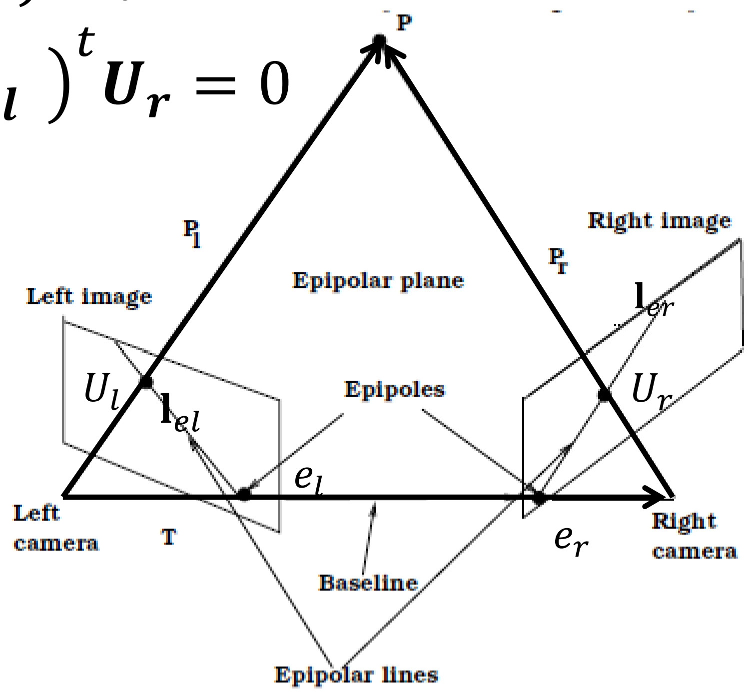
$$e_r^t \mathbf{l}_{er} = \mathbf{0} \rightarrow e_r^t \mathbf{F}^t U_l = 0 \quad \rightarrow (\mathbf{F} e_r)^t U_l = 0$$

$$e_l^t \mathbf{l}_{el} = \mathbf{0} \rightarrow e_l^t \mathbf{F} U_r = 0 \quad \rightarrow (\mathbf{F}^t e_l)^t U_r = 0$$

$\Rightarrow$

$$\mathbf{F} e_r = 0$$

$$\mathbf{F}^t e_l = 0$$



Given F, the epipole of right (left) image is the null eigenvector of **F** ($\mathbf{F}^t$)

Fundamental Matrix

- Fundamental matrix is defined on the pixel coordinate (row-column image coordinate)

- Given fundamental matrix, the epipolar geometry can be reconstructed without camera calibration!
  - Estimate conjugate epipolar lines

$$\mathbf{l}_{el}=\mathbf{F}\boldsymbol{U_r} \qquad \mathbf{l}_{er}=\mathbf{F}^{\boldsymbol{t}}\boldsymbol{U_l}$$

  - Estimate epipoles - the epipole of right (left) image is the null eigenvector of **F** ($\mathbf{F}^{\boldsymbol{t}}$)

$$\mathbf{F}\boldsymbol{e_r} = 0 \qquad \mathbf{F}^{\boldsymbol{t}}\boldsymbol{e_l} = 0$$

- **F** has a rank 2 and is determined by 7 independent parameters up to a scaling factor

Algorithm to Estimate F

Properties of F

- 7 DoFs
- Singularity: det(**F**) = 0
- Determined up to a scale factor

Solutions to F

- **7-point algorithm**
  - $\mathbf{F} = \alpha \mathbf{F_1} + (1 - \alpha)\mathbf{F_2}$
- **8-point algorithm**
  - Linear solution with singularity constraint enforced

8-point Algorithm

Major steps:

- Linear solution
- Constraint enforcement

8-point Algorithm

$$U_l^t \mathbf{F} U_r = \mathbf{0}$$

For N pairs of matched 2D points $U_l = (x_l, y_l, 1)^t$ and $U_r = (x_r, y_r, 1)^t$, we can have a linear function of the 9 unknown in F

$$\mathbf{A}^{N\times 9}\ \mathbf{v}^{9\times 1} = 0$$

where v=$[F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}]^t$

$$\mathbf{A}=\begin{bmatrix} x_{l_1}x_{r_1} & x_{l_1}y_{r_1} & x_{l_1} & y_{l_1}x_{r_1} & y_{l_1}y_{r_1} & y_{l_1} & x_{r_1} & y_{r_1} & 1 \\ \vdots & & & & \cdots & & & & \vdots \\ x_{l_N}x_{r_N} & x_{l_N}y_{r_N} & x_{l_N} & y_{l_N}x_{r_N} & y_{l_N}y_{r_N} & y_{l_N} & x_{r_N} & y_{r_N} & 1 \end{bmatrix}$$

How many pairs of corresponding points needed?

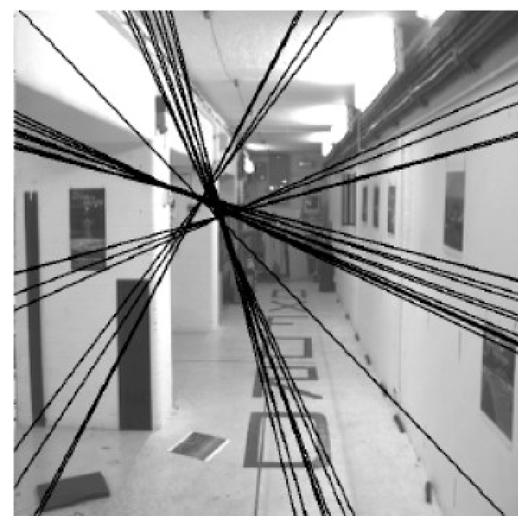
At least 8 pairs of non-coplanar points

Step1: 8-point Algorithm

Linear solution

- Form a linear equation system $\mathbf{A}^{N\times 9}\ \mathbf{v}^{9\times 1} = 0$
- Compute SVD on A as $\mathbf{A}=\mathbf{U}_{\mathrm{A}}\mathbf{D}_{\mathrm{A}}\mathbf{V}_{\mathrm{A}}{}^{\mathbf{t}}$
- The solution of **v** is the last column of $\mathbf{V}_{\mathrm{A}}$
- Construct the initial **F** matrix

What is the issue with the initial **F** matrix?

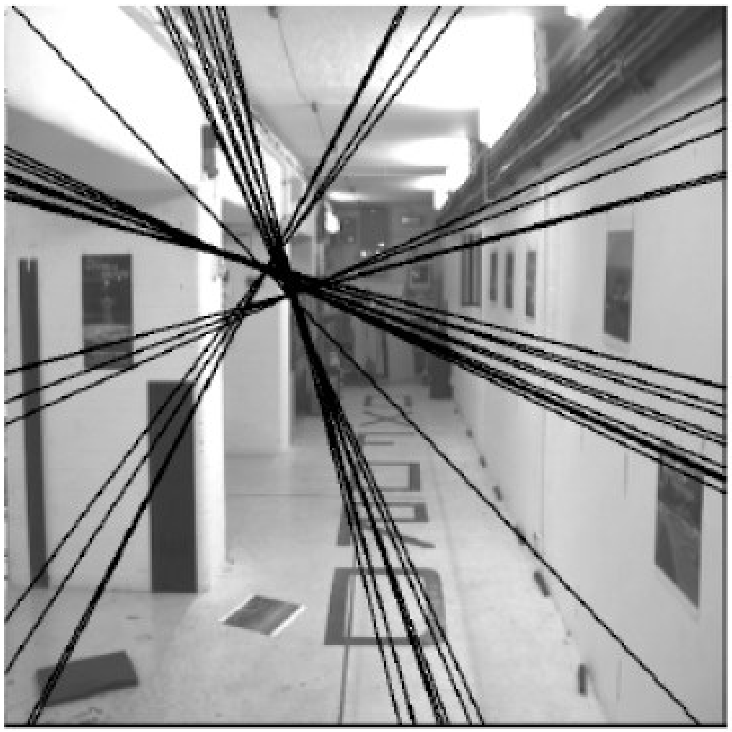


Step2: 8-point Algorithm

Enforce singularity constraint

- Perform SVD on **F** as $\mathbf{F}=\mathbf{U_F}\mathbf{D_F}\mathbf{V_F}^{\mathbf{t}}$
- Replace $\mathbf{D}_F$ with $\mathbf{D}_F{}'$ such that the last singular value of $\mathbf{D}_F{}'$ is 0
- Reconstruct $\mathbf{F'}=\mathbf{U_F}\mathbf{D_F}{}'\mathbf{V_F}^{\mathbf{t}}$

Singularity Constraint



Without the constraints, epipolar lines are not intersected at the same point

With the constraints, epipolar lines are intersected at the same point -- epipole

Normalized 8-point Algorithm

Normalized 8-point algorithm can improve the numerical stability

- **Normalize the image coordinates**
  - Translate the points such that the centroid is at origin
  - Apply a scaling such that the RMS distance of the points from origin is $\sqrt{2}$
- **Estimate F from the normalized 2D points**
- **Recover the original F by denormalization**

http://en.wikipedia.org/wiki/Eight-point_algorithm#The_normalized_eight-point_algorithm

Compute E from F

$$\mathbf{F} = \mathbf{W}_l{}^{-\mathbf{t}}\mathbf{E}\mathbf{W}_r{}^{-\mathbf{1}}$$

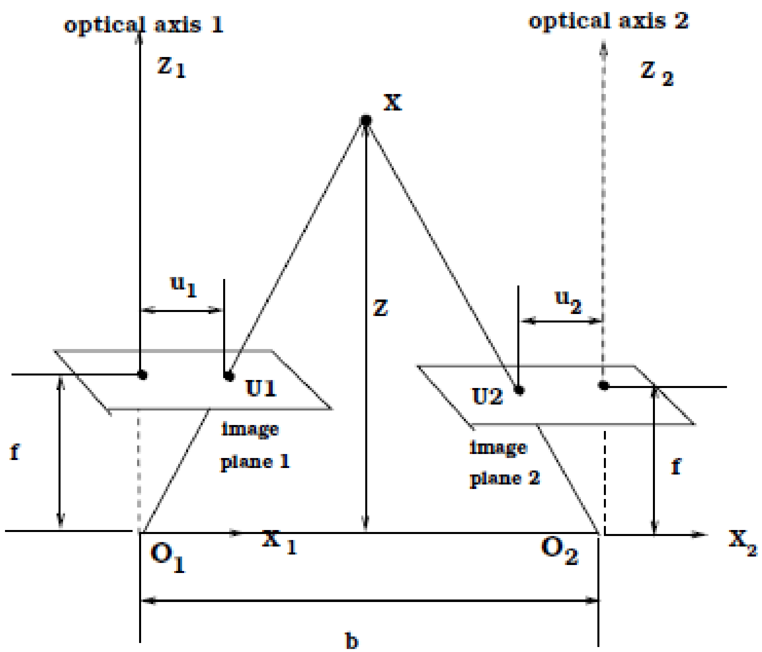
Given intrinsic parameters, E can be computed from F as

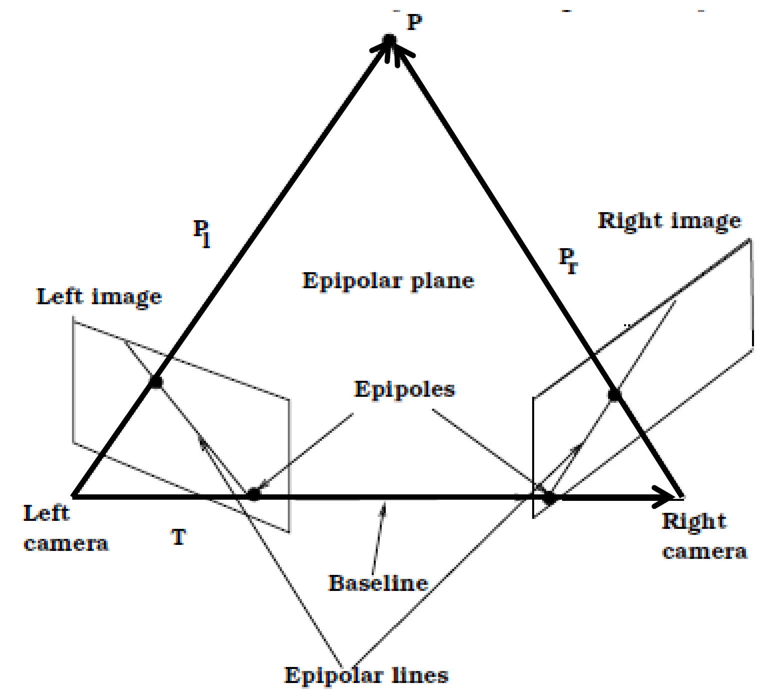
$$\mathbf{E} = \mathbf{W}_l{}^{\mathbf{t}}\mathbf{F}\mathbf{W}_r$$

Recover 3D Information – Estimate Depth

Depth can be recovered given the disparity in the rectified geometry:

$$Z = \frac{fb}{u_1 - u_2}$$





Recover 3D Information – Estimate Depth

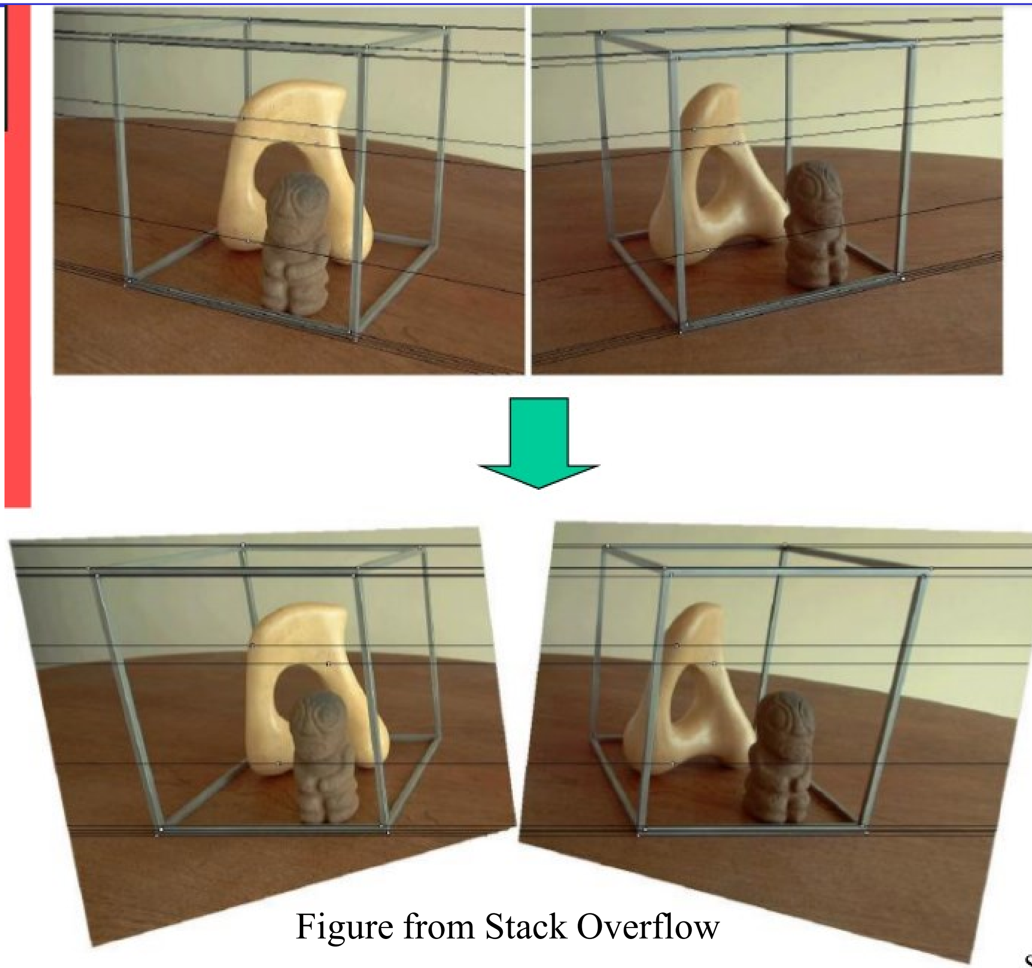


Figure from Stack Overflow

S

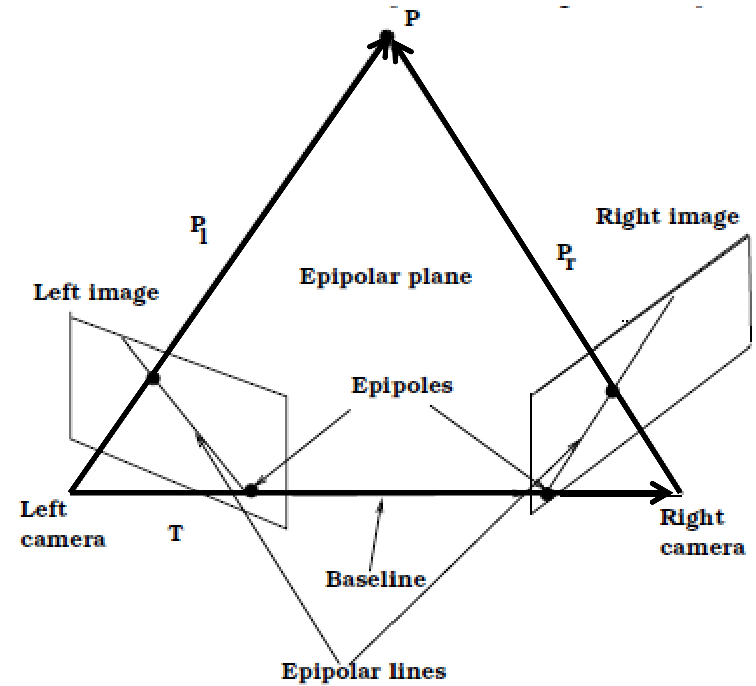
How to rectify images

Recover 3D Information – Estimate Depth

How to rectify images?

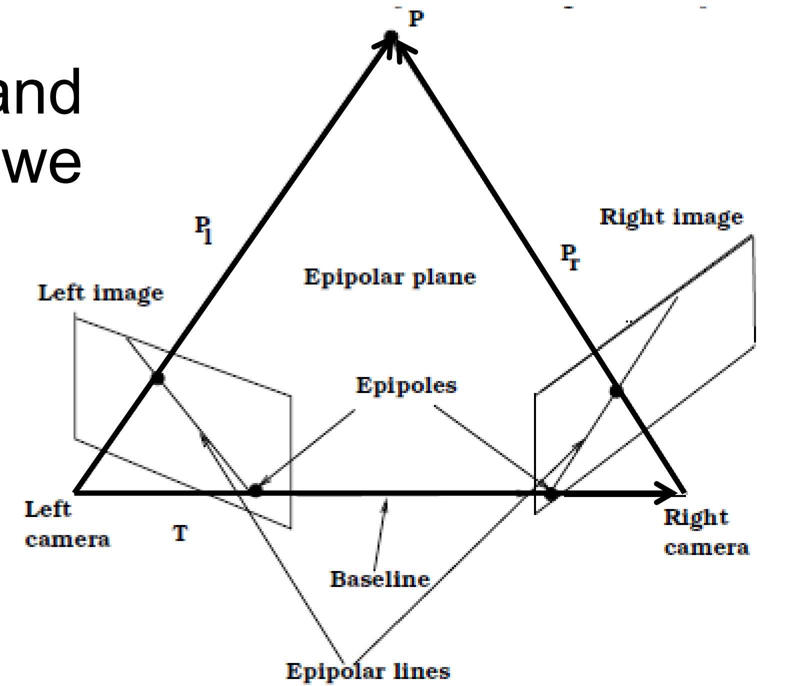
$$P_l = \mathrm{R}P_r + T$$

How to get **R** & **T**?



Recover 3D Information – Estimate Depth

Assume camera parameters (intrinsic and extrinsic) are known for both cameras, we can estimate the relative rotation and translation between two cameras:



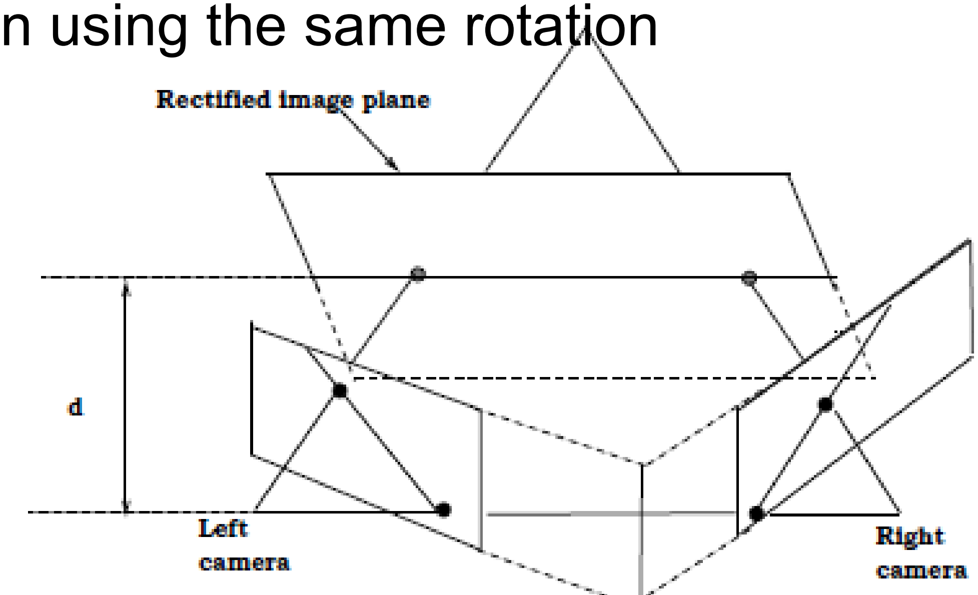
$$\mathrm{R} = \mathrm{R_l}\mathrm{R_r}^{\mathrm{t}} \qquad T = T_l - \mathrm{R}T_r$$

Rectification

Major steps (Trucco & Verri):

- Rotate the left camera such that the epipole goes to infinity along horizontal axis
- Rotate the right camera with **R**
- Rotate the right camera again using the same rotation matrix for the left camera
- Image reprojection: create

  new 2D images



courtesy of Dr. Qiang Ji

Rectification

Step1: Construct a rotation matrix R_{lrect} for left camera such that the image plane is parallel to the baseline.

Define the new axes

- the new x-axis $[1,0,0]^t$ should be on the base line – along the same direction as **T**

$$\mathbf{v}_1 = \frac{\mathbf{T}}{||\mathbf{T}||}$$

In the original left camera frame

- the new y-axis must be orthogonal to the x-axis
  – cross product of **T** and optical axis $[0,0,1]^t$

$$\mathbf{v}_2 = \mathbf{v}_1 \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{t_x^2 + t_y^2}} \begin{bmatrix} t_y \\ -t_x \\ 0 \end{bmatrix}$$

- the new z-axis is determined as the cross product of the other two

$$\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$$

Rectification

Step1: Construct a rotation matrix for left camera such that the image plane is parallel to the baseline R_{lrect}.

Let $\mathbf{r}_{lre_1}$, $\mathbf{r}_{lre_2}$, and $\mathbf{r}_{lre_3}$ be the three rows of R_{lrect}

$$R_{lrect}\mathbf{v}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix} \Rightarrow \mathbf{r}_{lre_1} = \mathbf{v}_1 = \frac{\mathbf{T}}{\|\mathbf{T}\|}$$

$$R_{lrect}\mathbf{v}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix} \Rightarrow \mathbf{r}_{lre_2} = \mathbf{v}_2 = \frac{1}{\sqrt{t_x^2 + t_y^2}}\begin{bmatrix}t_y\\-t_x\\0\end{bmatrix}$$

$$R_{lrect}\mathbf{v}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix} \Rightarrow \mathbf{r}_{lre_3} = \mathbf{v}_3 = \mathbf{r}_{l_1} \times \mathbf{r}_{l_2}$$

$$\mathbf{R}_{lrect} = \begin{bmatrix}\mathbf{r}_{lre_1}^t\\\mathbf{r}_{lre_2}^t\\\mathbf{r}_{lre_3}^t\end{bmatrix}$$

Rectification

Step 2: image reprojection for the left image

$U_l = (c_l, r_l, 1)^t$ → Original image point before rectification

$U_l{}' = (c_l{}', r_l{}', 1)^t$ → New image point after rectification

What is the relationship between U_l and $U_l{}'$?

They are actually created by the projection of the same 3D point on two image planes: same $\boldsymbol{W_l}$, rotated by $\boldsymbol{R_{lrect}}$

$$\lambda_l \begin{bmatrix} c_l \\ r_l \\ 1 \end{bmatrix} = \boldsymbol{W_l} \boxed{\boldsymbol{P_l}} \qquad \lambda_l{}' \begin{bmatrix} c_l{}' \\ r_l{}' \\ 1 \end{bmatrix} = \boldsymbol{W_l R_{lrect}} \boxed{\boldsymbol{P_l}}$$

$$\Rightarrow \quad \lambda_l \begin{bmatrix} c_l{}' \\ r_l{}' \\ 1 \end{bmatrix} = \boldsymbol{W_l R_{lrect} W_l}^{-1} \begin{bmatrix} c_l \\ r_l \\ 1 \end{bmatrix}$$

Rectification

Step 3: Compute the rotation matrix for the right camera

$$\mathbf{R}_{rrect} = \mathbf{R}_{lrect}\mathbf{R}$$

Step 4: image reprojection for the right image

$$\lambda_r \begin{bmatrix} {c_r}' \\ {r_r}' \\ 1 \end{bmatrix} = \boldsymbol{W_r R_{rrect} W_r}^{-1} \begin{bmatrix} c_r \\ r_r \\ 1 \end{bmatrix}$$