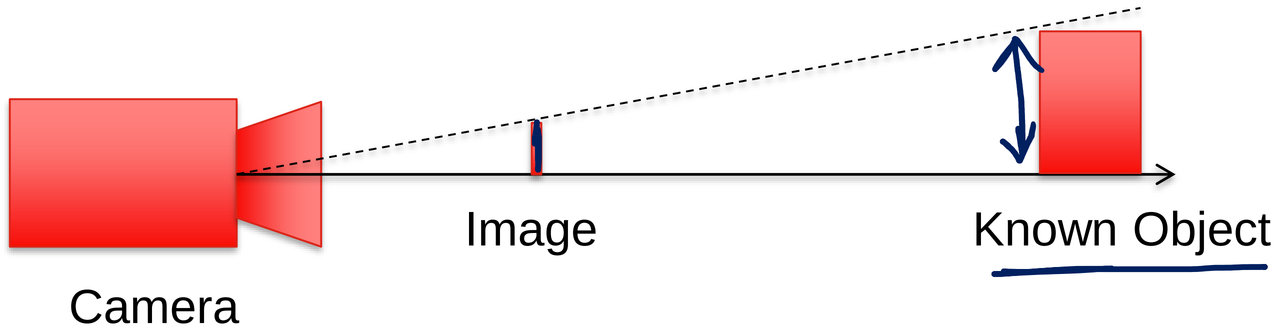
Content

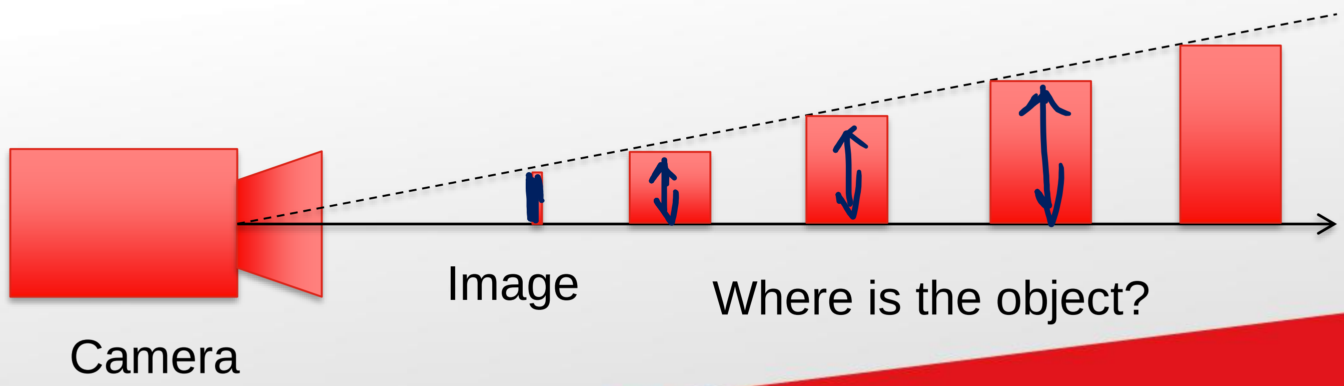
- Segmenting Multiple Blobs
- 3D Pose Estimation for Known Objects
  - Introduction
  - Camera Intrinsic Parameters
  - Camera Extrinsic Parameters
  - Camera Calibration
  - 3D Pose Estimation
- Depth Perception for Arbitrary Objects
  - Introduction
  - Stereo Disparity
  - Correspondence Problem
  - Non-coplanar Cameras

Introduction to Depth Perception

- If we have a calibrated camera AND a known object / model, then we know the depth (i.e. z-distance) of the object by doing pose estimation.

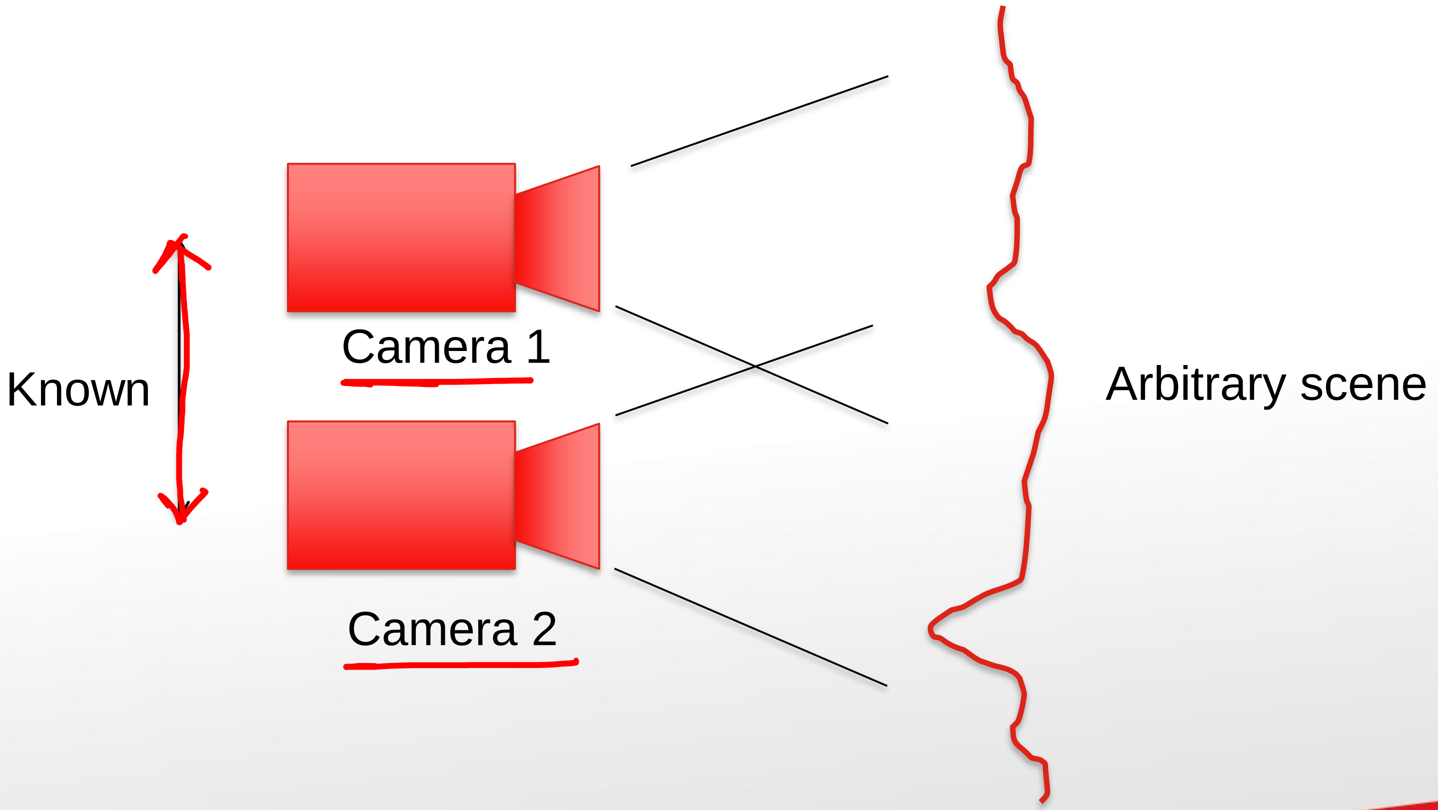


- However, if we do not know the object / model, then the depth is unknown!



Introduction to Depth Perception

- To be able to find out the distance for unknown / arbitrary objects, we need two calibrated cameras, with known relative pose.



Introduction to Depth Perception

- Stereo: Getting 3D information from 2 or more images.
- This method is used by human and animals to estimate distance:

https://www.zeiss.com

http://thestoneset.com/tigers-eye/

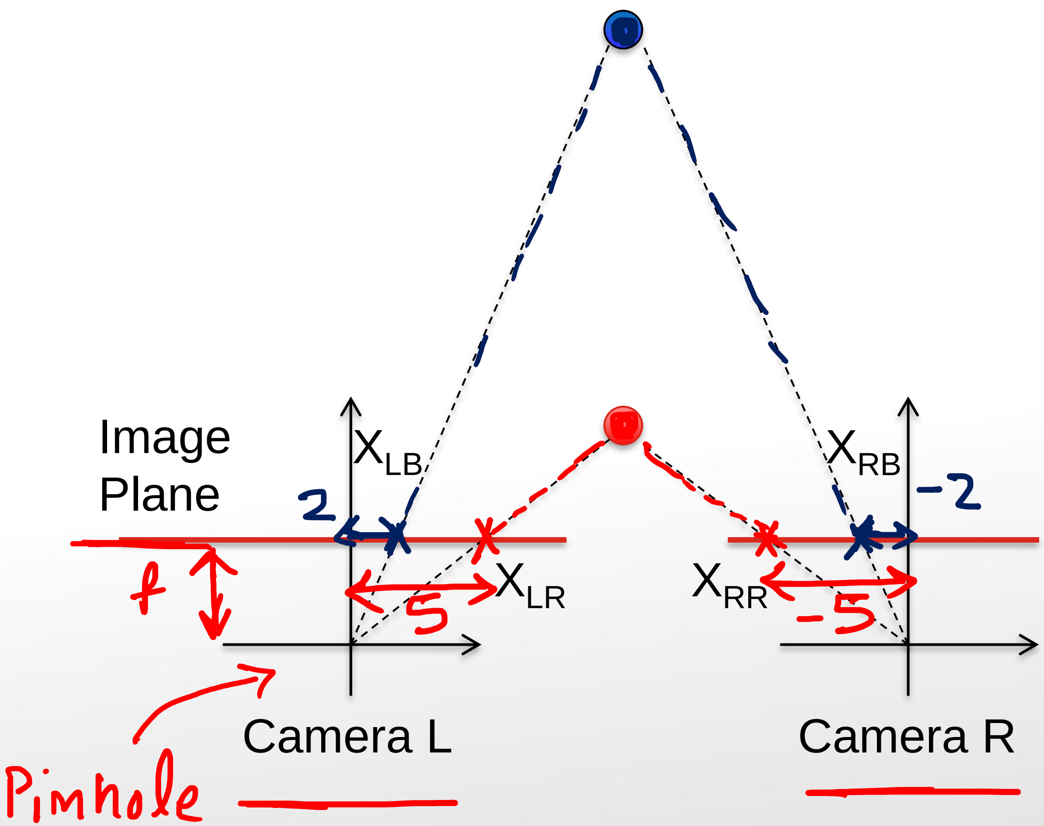
- And now, it's also used by computers / robots.

http://pfrommer.us/stereo-vision

Content

Stereo Disparity

- But how does having two eyes or two cameras solve the depth issue?
- Let's look at the following situation:



- The rays emanating from the cameras will pass through the points X_{LB}, X_{LR}, X_{RR} and X_{RB} on the image planes, before hitting the red and blue points.
- Assume the following numbers:
  - $X_{LB} = 2$
  - $X_{LR} = 5$
  - $X_{RR} = -5$
  - $X_{RB} = -2$

Stereo Disparity

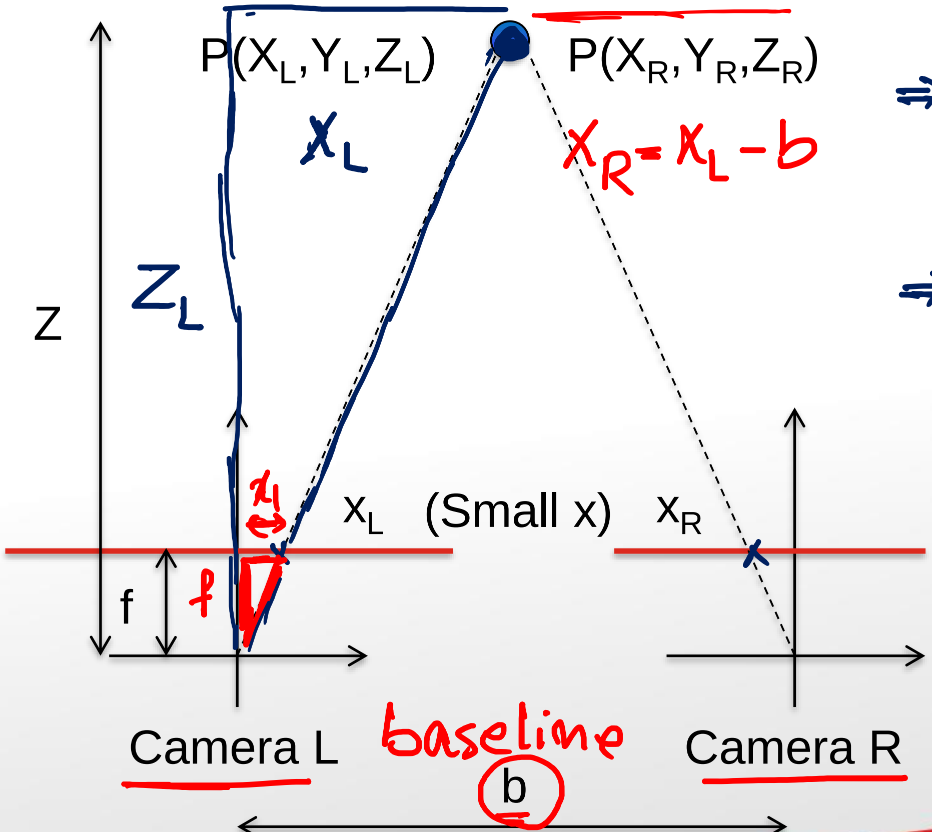
- We now compute the "disparity", i.e. the coordinate difference of a particular point in the left and right cameras.
  - For red point: $X_{RR} - X_{LR} = (-5) - (5) = -10 \rightarrow$ absolute disparity 10
  - For blue point: $X_{RB} - X_{LB} = (-2) - (2) = -4 \rightarrow$ absolute disparity 4
- As can be seen, a nearby point (red) gives a large disparity, and a faraway point (blue) gives as smaller disparity.

Disparity provides information about depth!

- Experiment: Close your right eye and look at two objects at different distance using your left eye. Switch you eye (left & right) continuously and you will notice that the nearer objects moves further between your eyes.

Stereo Disparity

- What is the equation between disparity and depth?
- Assume both cameras are coplanar, but right camera is located at a known distance "b" (called "baseline") from the left camera in the x-direction.



$$\frac{x_L}{f} = \frac{X_L}{Z_L} \Rightarrow x_L = f\frac{X_L}{Z_L}$$

$$\frac{x_R}{f} = \frac{X_R}{Z_R} \Rightarrow x_R = f\frac{X_R}{Z_R}$$

However, $Z_L = Z_R = Z$

And $X_R = X_L - b$

Therefore: $x_R = f\dfrac{(X_L - b)}{Z}$

Stereo Disparity

- The disparity "d" is defined as:

$$d = x_L - x_R = f\frac{X_L}{Z} - f\frac{(X_L - b)}{Z} = f\frac{b}{Z}$$

- Thus, the relationship between disparity (d) and depth (Z) is:

$$Z = f\frac{b}{d}$$

  - Inverse relationship: Smaller Z gives larger d, and vice versa.

- E.g. if d = 10 pixels, f = 400 pixels, b = 20cm

$$Z = f\frac{b}{d} = 400\text{pixels} \cdot \frac{20\text{cm}}{10\text{pixels}} = 800\text{cm}$$

Content

- Segmenting Multiple Blobs
- 3D Pose Estimation for Known Objects
  - Introduction
  - Camera Intrinsic Parameters
  - Camera Extrinsic Parameters
  - Camera Calibration
  - 3D Pose Estimation
- Depth Perception for Arbitrary Objects
  - Introduction
  - Stereo Disparity
  - Correspondence Problem
  - Non-coplanar Cameras

Correspondence Problem

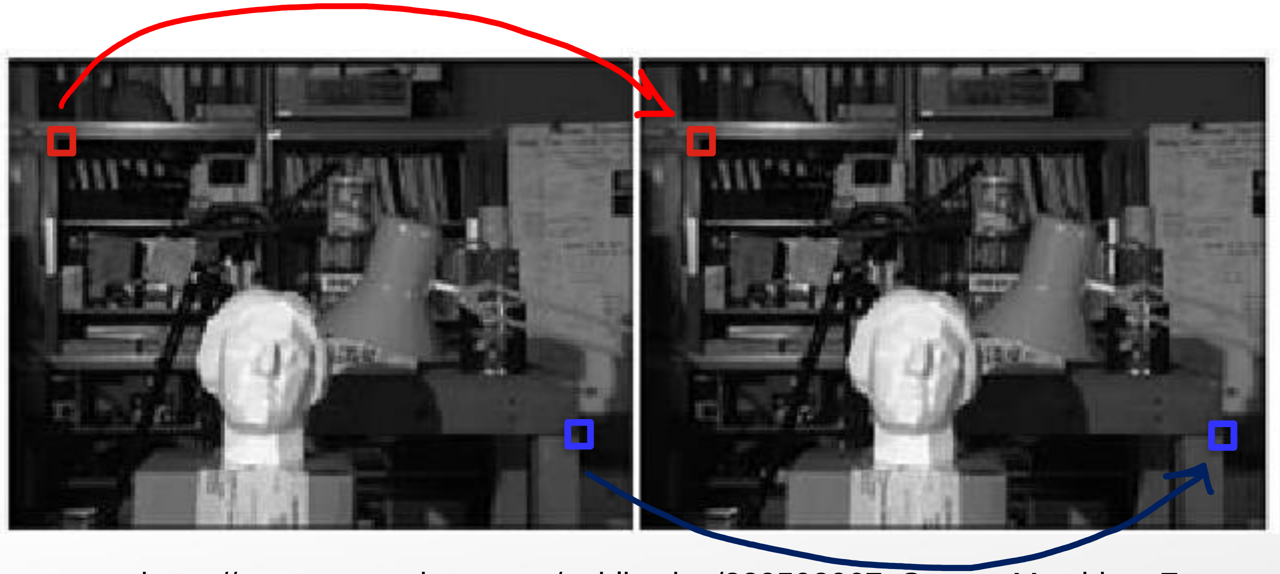
- In Summary:
- If you know:
  - The intrinsic parameters of the cameras (in this case f)
  - The relative pose between the cameras (in this case b)
- If you measure
  - An image point in the left camera
  - A corresponding point in the right camera
- You can intersect the rays (triangulate) to find the absolute point position.

This is called the "Correspondence Problem", and is in fact the most difficult problem in stereo vision!

- How to write an algorithm to find the exact matching points?

Correspondence Problem

- For example, how to write an algorithm that recognises that the region marked by the blue boxes / red boxes on both pictures as being the "same" regions?

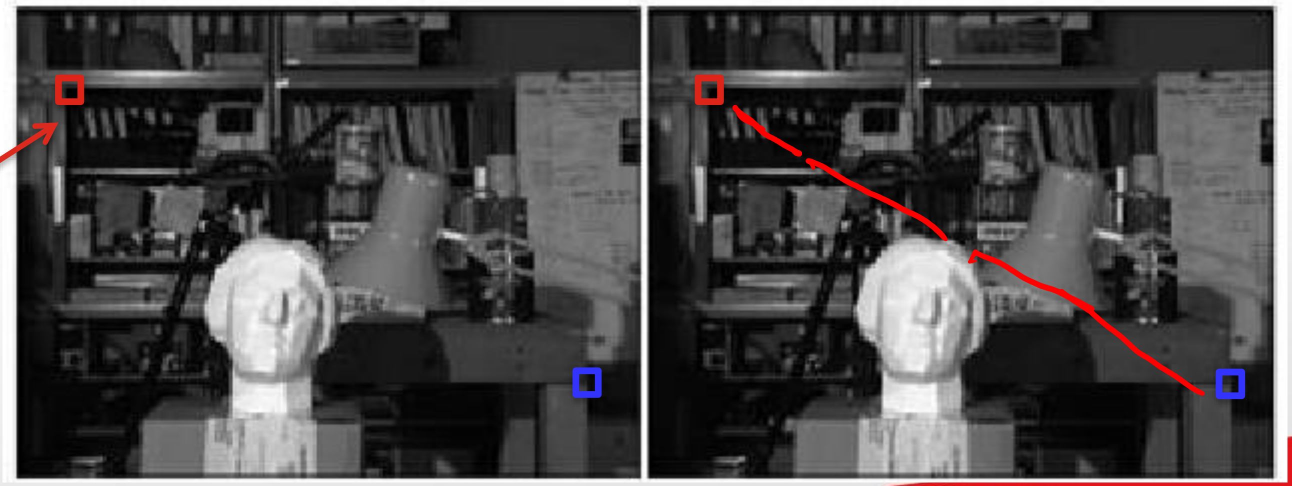


https://www.researchgate.net/publication/229592067_Stereo_Matching_From_the_Basis_to_Neuromorphic_Engineering/figures?lo=1

Feature-Based Matching

- Two major approaches:
  - Feature-based:
    - Pick a feature type (e.g. edges / corners) using detection methods.
    - Define a matching criteria (e.g. orientation and contrast sign)
    - Then look for matches within disparity range.
    - Other points in between features can be linearly interpolated.

Corner



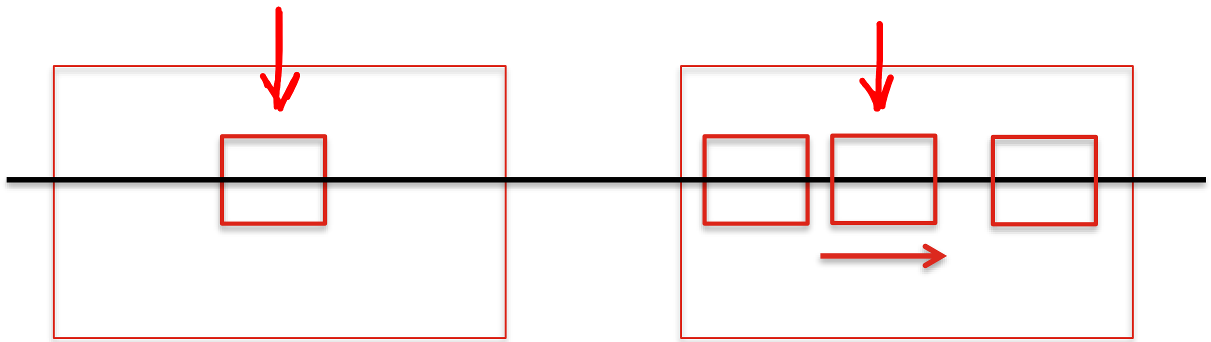
Region-Based Matching

- Region-based:
  - Forget about features.
  - Pick a region in the image, and find the matching region in the 2nd image by
    - – minimizing some measure, e.g. sum of squared difference (SSD), sum of absolute difference (SAD) etc; or
    - – maximizing some measure, e.g. (normalized) cross correlation

200	200	200	200	200
200	200	200	200	200
200	200	0	0	0
200	200	0	0	0
200	200	0	0	0

200	200	200	200	200
200	200	200	200	200
200	200	0	0	0
200	200	0	0	0
200	200	0	0	0

Region-Based Matching



Epipolar line for coplanar cameras with no y-shift

- Sum of squared difference (SSD):

$$\Rightarrow \quad SSD = \sum_{(i,j)\in \text{window}} \left(I_L(i,j) - I_R(i,j)\right)^2$$

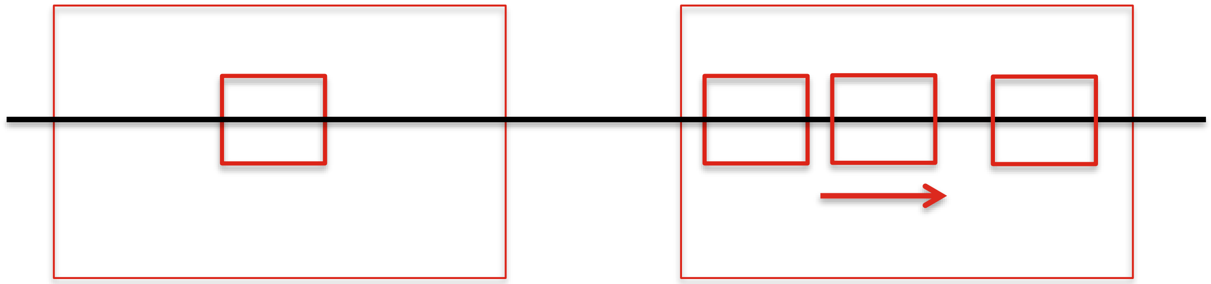
Surrounding Pixels

Element-wise

- Sum of absolute difference (SAD):

$$SAD = \sum_{(i,j)\in \text{window}} \left|I_L(i,j) - I_R(i,j)\right|$$

Region-Based Matching

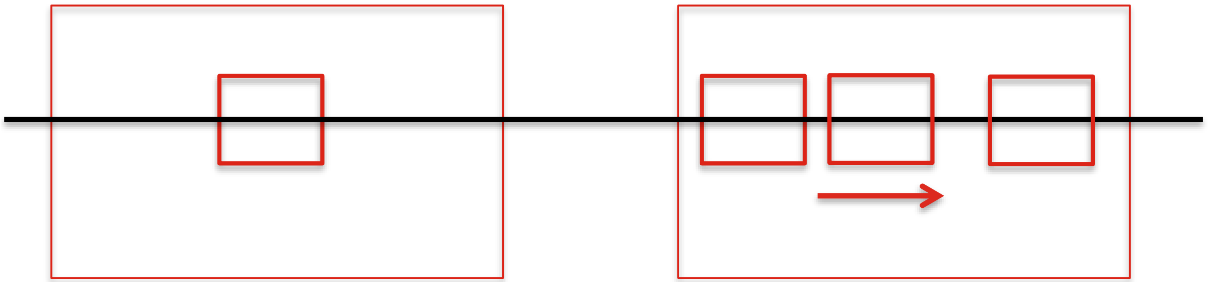


Epipolar line for coplanar cameras with no y-shift

- As we move along the epipolar line, the SSD or SAD would look something like this:

- The minimum error would thus correspond to the matching point.

Region-Based Matching



Epipolar line for coplanar cameras with no y-shift

- Cross Correlation (CC)

$$CC = \sum_{(i,j)\in \text{window}} \left(I_L(i,j) \times I_R(i,j) \right)$$

Surrounding Pixels

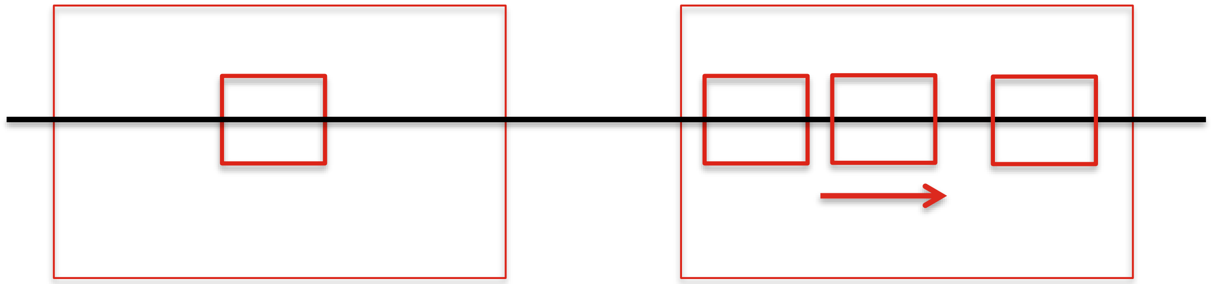
Element-wise

- Normalized Cross Correlation (NCC) to remove the effect of different illumination:

Mean

$$NCC = \sum_{(i,j)\in \text{window}} \left(\frac{I_L(i,j) - \bar{I}_L}{\sqrt{\sum \left(I_L(i,j) - \bar{I}_L \right)^2}} \times \frac{I_R(i,j) - \bar{I}_R}{\sqrt{\sum \left(I_R(i,j) - \bar{I}_R \right)^2}} \right)$$

Region-Based Matching



Epipolar line for coplanar cameras with no y-shift

- As we move along the epipolar line, the CC or NCC would look something like this:

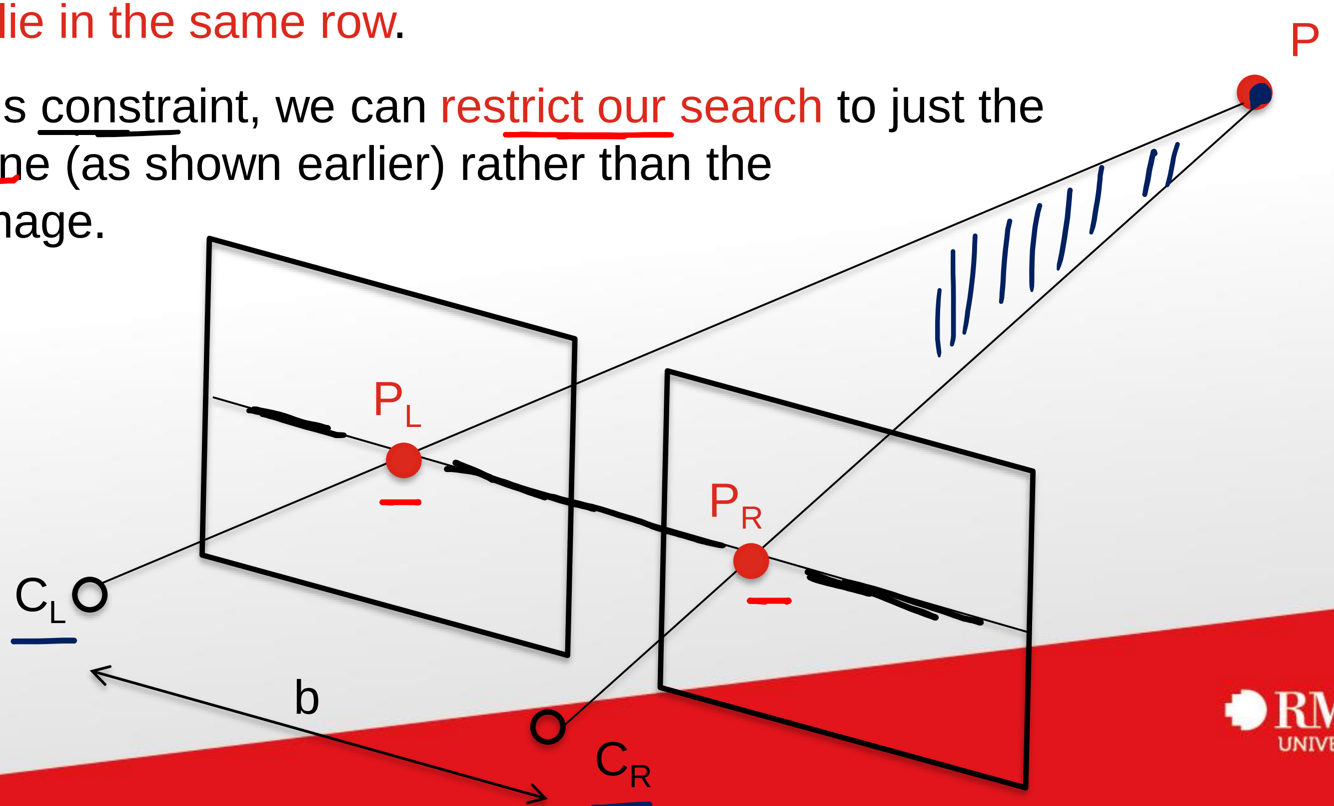
- The maximum correlation would thus correspond to the matching point.

Region-Based Matching

- Choice of Window Size:
- Smaller window:
  - Good precision, more details
  - Sensitive to noise
- Larger window:
  - Robust to noise
  - Reduced precision, less details

Epipolar Constraint

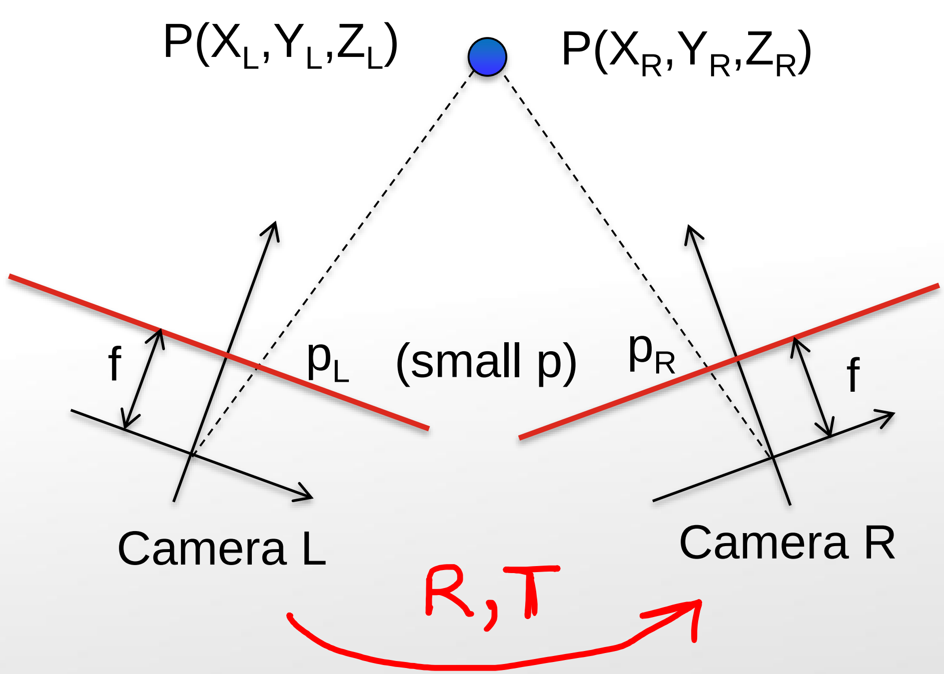
- We have used the term "Epipolar" just now. What does that mean?
- As shown in figure below, C_L, C_R and P form a plane.
- The image points would definitely lie on this plane.
  - This is called the Epipolar constraint.
- For coplanar cameras with no y-shift, the image points on both cameras would thus lie in the same row.
- By using this constraint, we can restrict our search to just the horizontal line (as shown earlier) rather than the complete image.



Content

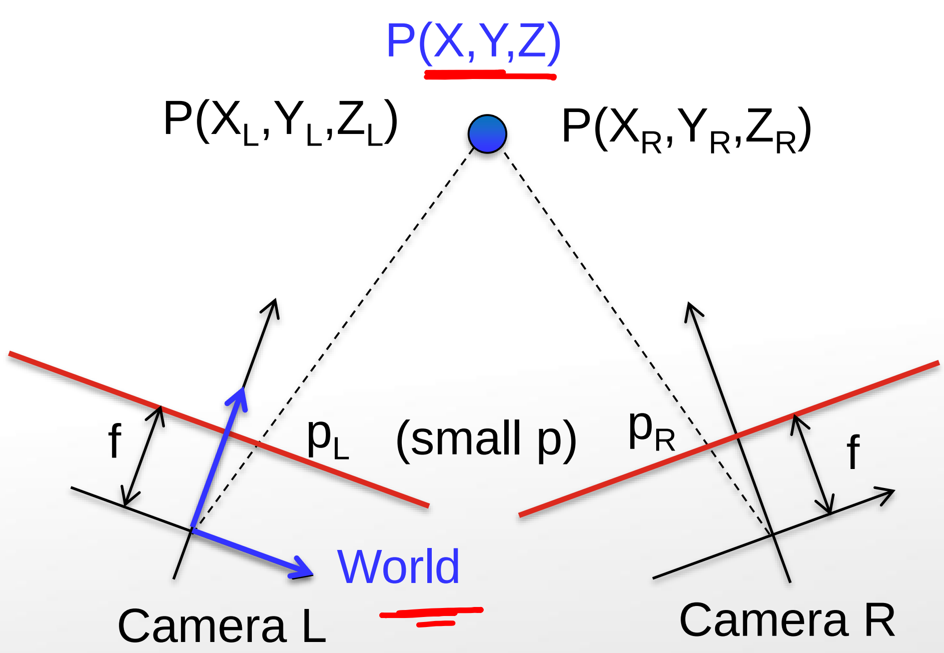
Non-Coplanar Cameras

- We have so far looked at the case where the cameras are co-planar.
- What if the cameras are not co-planar?
- Assumption: we know the relative pose of the cameras, and they are also calibrated.



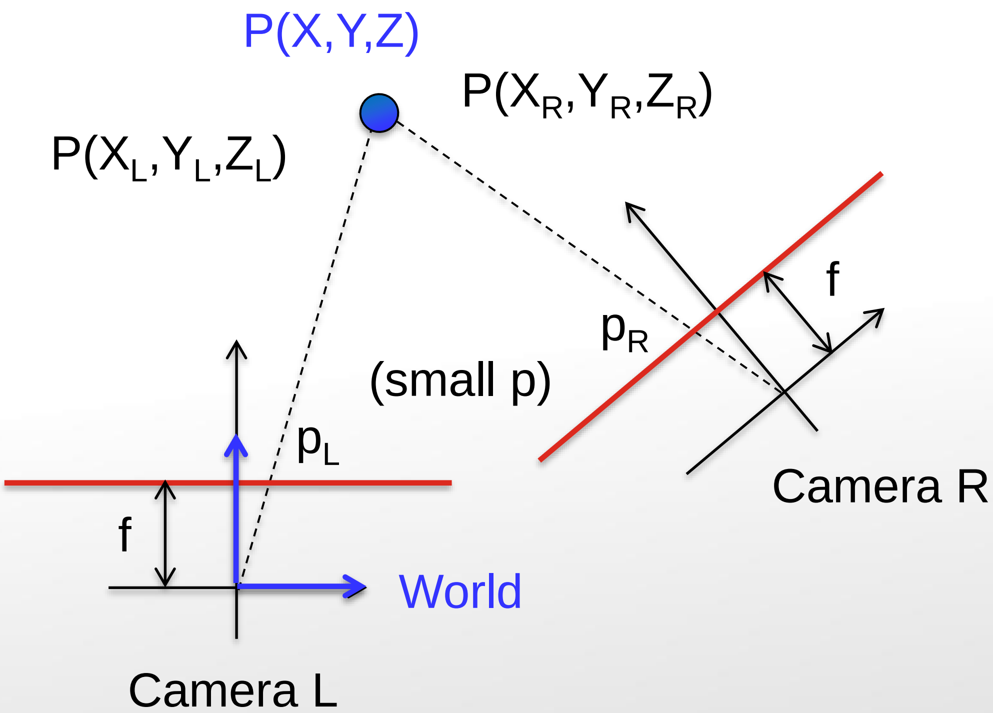
Non-Coplanar Cameras

- Let's fix the world coordinate frame at the left camera frame:



Non-Coplanar Cameras

- The view below is rotated for easier visualisation of theory later:



Reminder on Camera Matrix

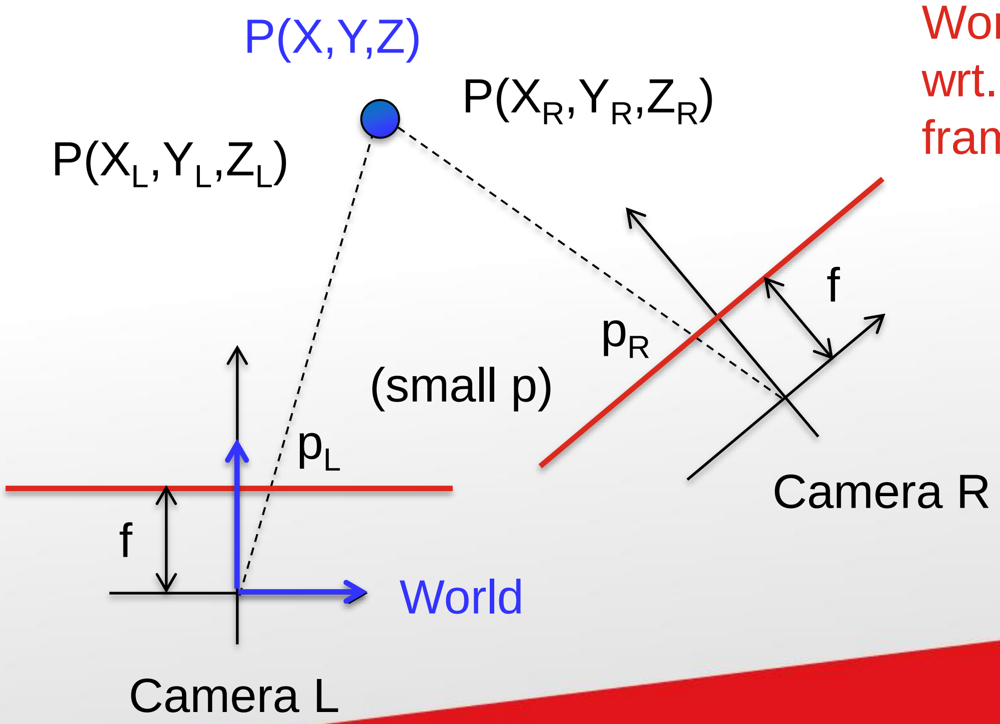
- A quick reminder of what we learnt earlier:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim K \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = K\left(R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \right) = K\begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Pixel coordinate

World coordinate

World frame wrt. Camera frame



$$K = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\alpha_x = \frac{f}{dx} \qquad \alpha_y = \frac{f}{dy}$$

Left Camera Matrix

- For the left camera:

$$\Rightarrow \quad [R \quad T] = [\underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I} \; \underbrace{\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}}_{0}]$$

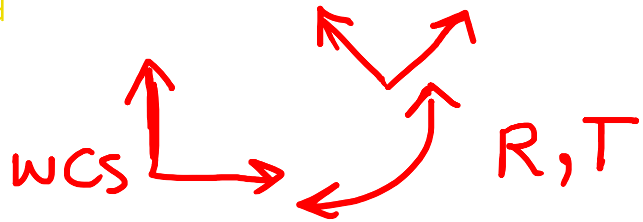
- Therefore:

$$\Rightarrow \quad \begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} \sim K[R \quad T]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \underline{\begin{bmatrix} \alpha_{lx} & 0 & x_{l0} \\ 0 & \alpha_{ly} & y_{l0} \\ 0 & 0 & 1 \end{bmatrix}}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} \alpha_{lx} & 0 & x_{l0} & 0 \\ 0 & \alpha_{ly} & y_{l0} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{P}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} p_{l11} & p_{l12} & p_{l13} & p_{l14} \\ p_{l21} & p_{l22} & p_{l23} & p_{l24} \\ p_{l31} & p_{l32} & p_{l33} & p_{l34} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Right Camera Matrix



- For the right camera:

$$[R \quad T] = [{}^R_L R \quad {}^R P_{LORG}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

- Therefore:

$$\Rightarrow \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \sim K[R \quad T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_{rx} & 0 & x_{r0} \\ 0 & \alpha_{ry} & y_{r0} \\ 0 & 0 & 1 \end{bmatrix}}_{K} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_{rx} r_{11} + x_{r0} r_{31} & \boxed{\alpha_{rx} r_{12} + x_{r0} r_{32}} & \alpha_{rx} r_{13} + x_{r0} r_{33} & \alpha_{rx} t_x + x_{r0} t_z \\ \alpha_{ry} r_{21} + y_{r0} r_{31} & \alpha_{ry} r_{22} + y_{r0} r_{32} & \alpha_{ry} r_{23} + y_{r0} r_{33} & \alpha_{ry} t_y + y_{r0} t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

right camera

$$= \begin{bmatrix} p_{r11} & \boxed{p_{r12}} & p_{r13} & p_{r14} \\ p_{r21} & p_{r22} & p_{r23} & p_{r24} \\ p_{r31} & p_{r32} & p_{r33} & p_{r34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Solving for X Y Z

- Remember that "proportional" means "equal up to scale", and thus the cross products of the left and right items are zero.
- Thus, for the left camera:

$$\begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} \times \begin{bmatrix} p_{l11} & p_{l12} & p_{l13} & p_{l14} \\ p_{l21} & p_{l22} & p_{l23} & p_{l24} \\ p_{l31} & p_{l32} & p_{l33} & p_{l34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- And for the right camera:

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \times \begin{bmatrix} p_{r11} & p_{r12} & p_{r13} & p_{r14} \\ p_{r21} & p_{r22} & p_{r23} & p_{r24} \\ p_{r31} & p_{r32} & p_{r33} & p_{r34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Solving for X Y Z

- If we carry out the cross product, the first two rows of the left camera vector equations are:

$$\begin{bmatrix} (y_l p_{l31} - p_{l21})X + (y_l p_{l32} - p_{l22})Y + (y_l p_{l33} + p_{l23})Z + (y_l p_{l34} - p_{l24}) \\ (p_{l11} - x_l p_{l31})X + (p_{l12} - x_l p_{l32})Y + (p_{l13} - x_l p_{l33})Z + (p_{l14} - x_l p_{l34}) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ * \end{bmatrix}$$

- Similarly, for the right camera, the first two rows are:

$$\begin{bmatrix} (y_r p_{r31} - p_{r21})X + (y_r p_{r32} - p_{r22})Y + (y_r p_{r33} + p_{r23})Z + (y_r p_{l34} - p_{r24}) \\ (p_{r11} - x_r p_{l31})X + (p_{r12} - x_r p_{l32})Y + (p_{r13} - x_r p_{l33})Z + (p_{r14} - x_r p_{l34}) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ * \end{bmatrix}$$

Solving for X Y Z

- The four equations can then be brought into the "$A\theta = b$" form:

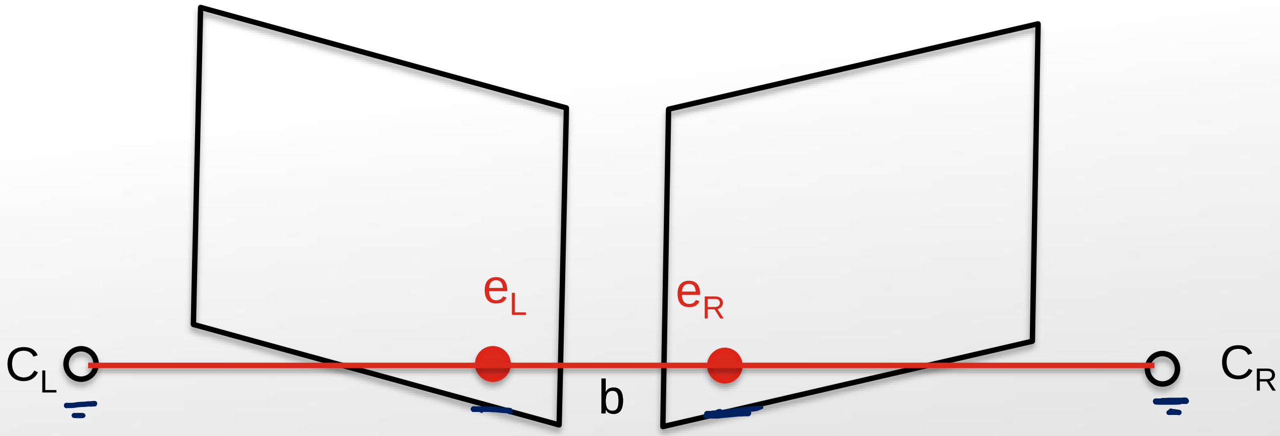
$$\begin{bmatrix} (y_l p_{l31} - p_{l21}) & (y_l p_{l32} - p_{l22}) & (y_l p_{l33} + p_{l23}) \\ (p_{l11} - x_l p_{l31}) & (p_{l12} - x_l p_{l32}) & (p_{l13} - x_l p_{l33}) \\ (y_r p_{r31} - p_{r21}) & (y_r p_{r32} - p_{r22}) & (y_r p_{r33} + p_{r23}) \\ (p_{r11} - x_r p_{l31}) & (p_{r12} - x_r p_{l32}) & (p_{r13} - x_r p_{l33}) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (p_{l24} - y_l p_{l34}) \\ (x_l p_{l34} - p_{l14}) \\ (p_{r24} - y_r p_{l34}) \\ (x_r p_{l34} - p_{r14}) \end{bmatrix}$$

- We can finally solve for X, Y, Z using least squares method.

$$\Rightarrow \quad \theta = \left(A^T A\right)^{-1} A^T b$$

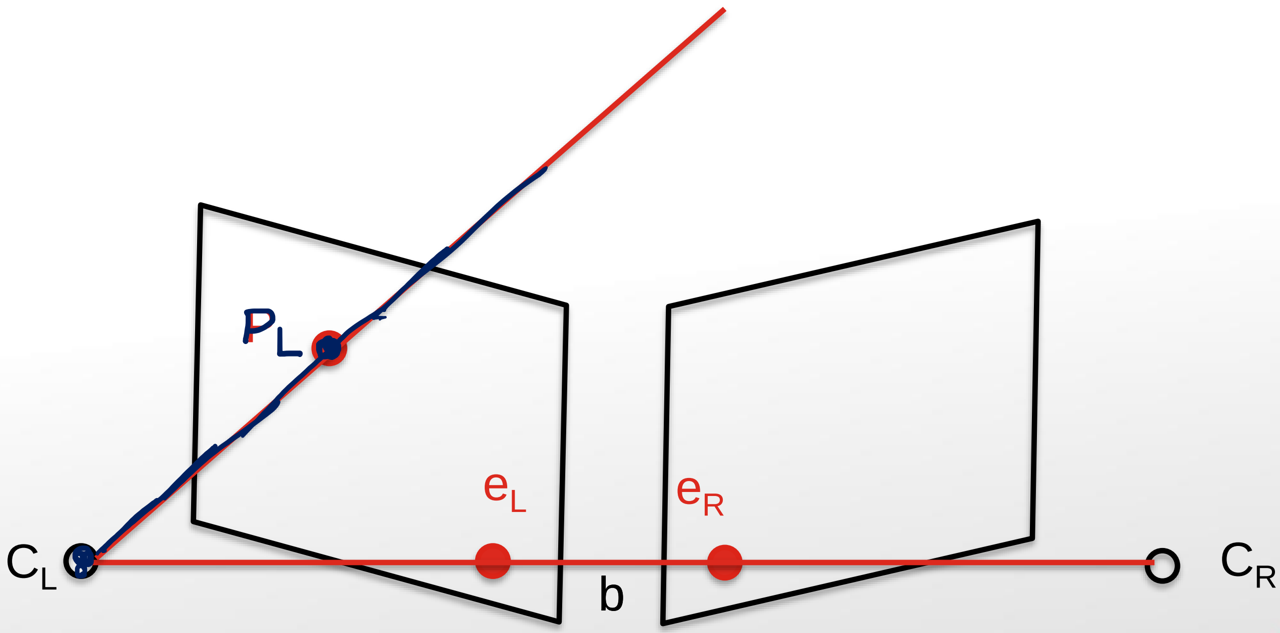
Correspondence Problem

- The concept of correspondence matching is still the same (as coplanar case) – We need to find matching portions of the left and right images.
  - Use SSD, SAD, CC or NCC as discussed earlier.
- However, the epipolar line is not necessarily horizontal anymore!
- Firstly, introduce the terms "epipoles (e_L and e_R)", i.e. the points where the camera baseline (C_L-C_R line) hits the left and right image planes.



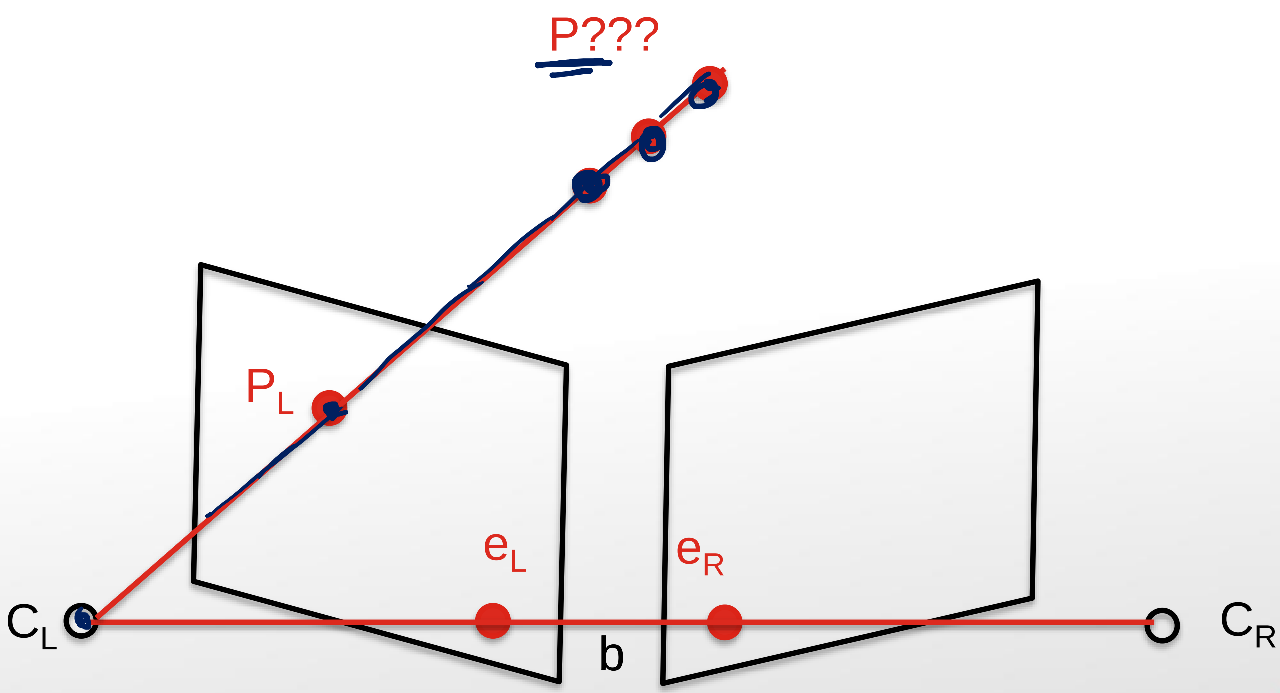
Epipolar Constraint

- Next, assume we have an image point on the left image.
- So the ray from C_L through P_L is known.



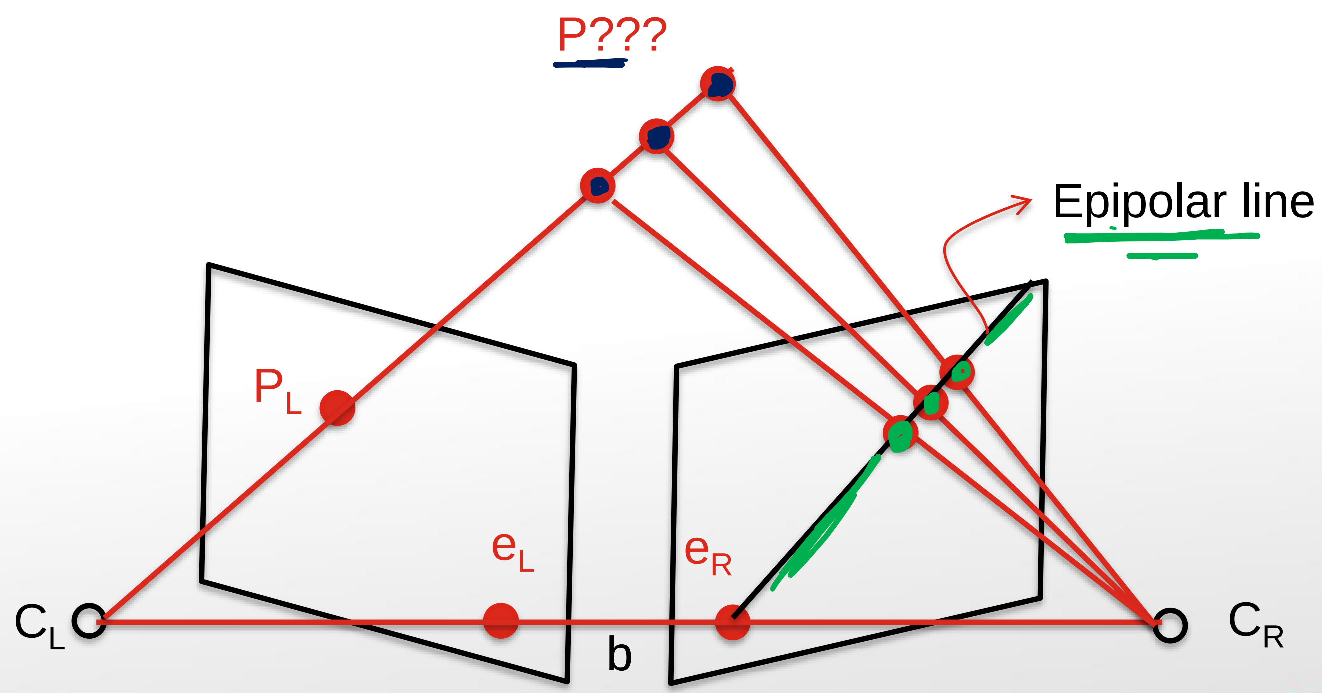
Epipolar Constraint

- However, where is the actual point P?
- It must lie along the C_L-P_L ray!



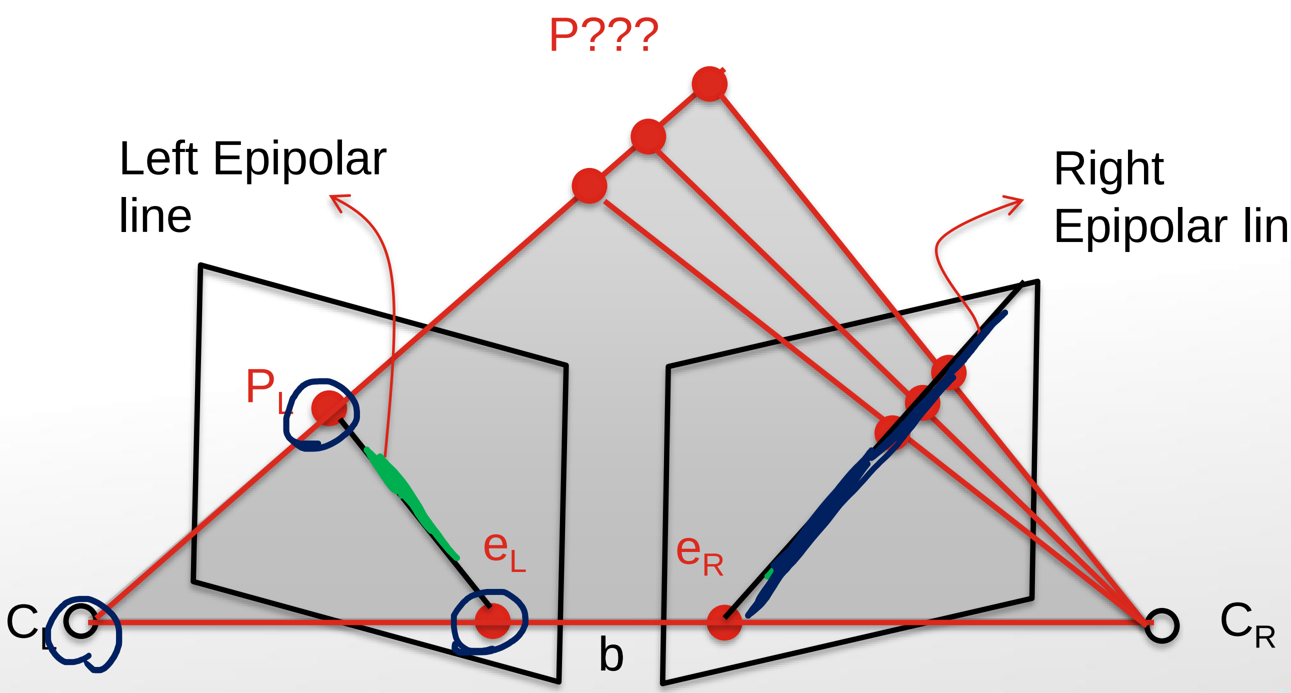
Epipolar Constraint

- If we project all these possible P's to the right camera,
- They would all lie along the epipolar line as shown in the figure.
  - We only need to search along this line for correspondence matching!



Epipolar Constraint

- Note that the left and right epipolar lines lie on the epipolar plane, which contains C_L, P_L and e_L.
- Thus once we know these three points, we can find out the epipolar lines.



Tutorial Assignments

- There is no tutorial assignment for this week.

Thank you!

Have a good evening.

RMIT UNIVERSITY