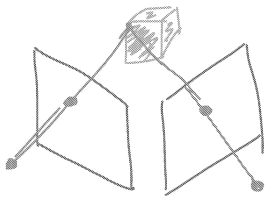
2-view geometry (stereo)

- use projection of the same 3D point in multiple views to recover structure (inverse problem)



- problems:
  - Correspondence
  - Reconstruction

Correspondence

- Approaches
  - Sparse: feature-based   handle large disparities
  - dense: correlation, SSD   produce more points
- Cases when correspondence is not possible:
  - object points not visible in both views (occlusion)
  - ambiguous points (multiple candidates)
  - uniform regions (inside points cannot be distinguished)

View image

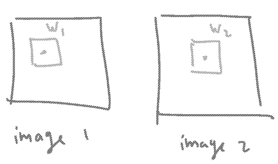
Sparse correspondence

- Find feature points (e.g. corners)
- Find local characterization (e.g. SIFT, HOG)
- Find corresponding points having similar features
- Greedy vs. optimal assignment
- Constraints for reducing # of candidates (e.g. search around current location)

Dense correspondence

- Instead of feature points compare all patches
- Instead of distance between feature vectors measure correlation or SSD
- Apply regularization to reduce errors and find correspondence in difficult areas (eg. uniform).

Dense correspondence



correlation: $\psi(w_1, w_2) = \sum_i w_1(x_i, y_i) \cdot w_2(x_i, y_i)$

SSD: Sum of Squared Deviations $\psi(w_1, w_2) = -\sum_i \left(w_1(x_i, y_i) - w_2(x_i, y_i)\right)^2$

High value = good correspondence

Dense correspondence

- Zero-mean normalized cross-correlation (ZNCC):

$$\psi(w_1, w_2) = \sum_i \frac{\left(w_1(x_i, y_i) - \mu_{w_1}\right)\left(w_2(x_i, y_i) - \mu_{w_2}\right)}{\sigma_{w_1}\,\sigma_{w_2}}$$

- Zero-mean normalized SSD (ZN SSD):

$$\psi(w_1, w_2) = \sum_i \left(\frac{\left(w_1(x_i, y_i) - \mu_{w_1}\right)}{\sigma_{w_1}} - \frac{\left(w_2(x_i, y_i) - \mu_{w_2}\right)}{\sigma_{w_2}}\right)^2$$

A stereo camera is a camera system that uses two or more lenses, typically positioned a fixed distance apart, to capture images of the same scene from slightly different perspectives. This setup mimics the way human eyes work, enabling the perception of depth and 3D structure in a scene.

Two Lenses (or Cameras):

A stereo camera has two lenses (or cameras), each capturing an image from a slightly different angle.

Stereo Vision:

By comparing the differences (disparities) between the two images, the system can calculate depth information, allowing it to reconstruct a 3D view of the scene.

Depth Perception:

Objects closer to the camera will have larger disparities (differences in position between the two images) than objects further away.

Axis-aligned stereo

- Using similar triangles:

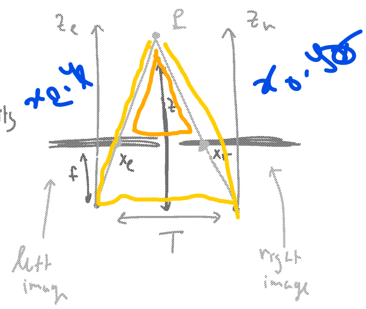
$$\frac{T}{z} = \frac{T-(-x_l+x_r)}{z-f}$$

disparity

$$= \frac{T-(x_r-x_l)}{z-f} = \frac{T-d}{z-f}$$

$$\frac{T}{z} = \frac{T-d}{z-f} \Rightarrow \boxed{z = f\,\frac{T}{d}}$$

small disparity large z



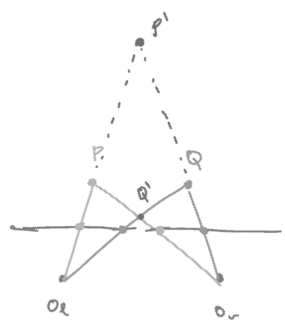
yl=yr as they are in same plan

T - base line

Ambiguity problem

Correct: P, Q

incorrect: P', Q'



General stereo

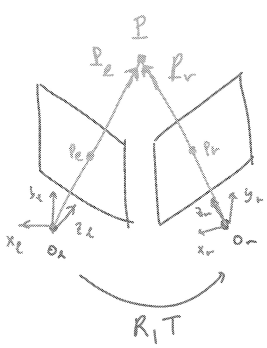
P = point in world coordinates

P_l = point in left coordinates

P_r = point in right coordinates

p_l = image point in left coordinates

p_r = image point in right coordinates



Getting relative rotation and translation

calibration $\Rightarrow \begin{cases} R_L, T_L \\ R_r, T_r \end{cases}$ rotation/translation of left/right camera w.r.t. world

$$M_{left \leftarrow right} = \underbrace{M_{left \leftarrow world}}_{R_l^T T_l^{-1}} \; \underbrace{M_{world \leftarrow right}}_{(R_r^T T_r^{-1})^{-1} = T_r R_r}$$

$$= R_l^T T_l^{-1} T_r R_r$$

Getting relative rotation and translation

$$M_{left \leftarrow right} = R_l^T T_l^{-1} T_r R_r$$

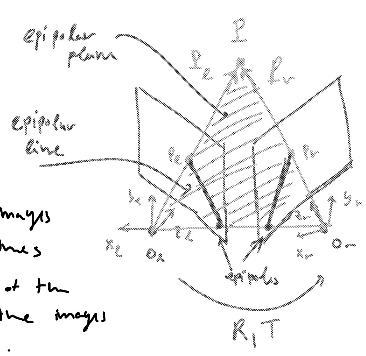
$$= \begin{bmatrix} R_l^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -T_l \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & T_r \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_r & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} R_l^T & -R_l^T T_l \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_r & T_r \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_l^T R_r & R_l^T (T_r - T_l) \\ 0 & 1 \end{bmatrix}$$

$$\Rightarrow \boxed{R = R_l^T R_r} \quad \boxed{T = R_l^T (T_r - T_l)}$$

Epipolar geometry

- The points O_l, O_r, P define epipolar plane
- The intersection of the epipolar plane with the images define epipolar lines
- The intersection of the baseline with the images defines the epipoles.



the inersection of all the epipolar lines also meet at epipoles....which may be inside or outside the image. The epipoles move to infinity in opp direction, if the images are more coplanar

Epipolar geometry

* Properties of epipolar lines:
  - every epipolar line must pass through the epipole
  - epipoles may be inside or outside the image
  - epipoles may be at infinity (when the image planes become parallel).

Epipolar constraint

- Given a point p_e in the left image p_e, O_e, O_r define an epipolar plane.
- The intersection of this epipolar plane with the right image defines an epipolar line on it.
- The point p_r corresponding to p_e on the left image must be on this epipolar line

Essential matrix

- P_r and P_e are expressed in different coordinate systems. We would like to move to the left coordinate system.
- Given that the right view is rotated and translated by R, T w.r.t. left, it is easy to move from left to right:

$P_r = R^T(P_e - T)$

$\Rightarrow P_e = R P_r + T$ ← P_r represented in left coords

Essential matrix

- Moving from camera to image coords:

$$p_l = \frac{f_l}{z_l} P_l$$

$$p_r = \frac{f_r}{z_r} P_r$$

- Cross product as matrix multiplication:

$$A \times B = [A]_\times B \equiv \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

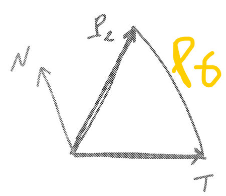
rank 2 skew-symmetric matrix

Essential matrix

- To specify an epipolar plane we need to find a vector normal to it.

A plane is found with the help of the vector normal to it. The normal vector is found by the cross product of 2 vectors on the plan

$$N = T \times P_l$$



- The vector P_r is on the epipolar plane:

$$P_r \cdot N = 0$$

Note that the dot product of the vectors on the plane and the normal vector is zero

- Moving P_r to left coordinates:

$$(RP_r + T) \cdot N = 0$$

Everthing should be same coordinate system. Pl,T in left system, so does N. therefore, we convert Pr in right system to left

Essential matrix

$$(RP_r + T) \cdot N = 0 \Rightarrow (RP_r + \underbrace{T) \cdot (T \times P_l}_{=0}) = 0$$

$$\Rightarrow RP_r \cdot (T \times P_l) = 0$$

$$\Rightarrow (RP_r)^T [T]_\times P_l = 0$$

$$\Rightarrow P_r^T \underbrace{R^T [T]_\times}_{\equiv E} P_l = 0$$

3×3

$$\Rightarrow P_r^T E P_l = 0$$

Essential matrix

- The essential matrix

$$\boxed{E = R^T [T]_x}$$

rank 2 (because of $[T]_x$)

- Epipolar constraint:

$$p_r^T E \, p_\ell = 0$$

- Epipolar constraint in image coordinates:

$$\boxed{p_r^T E \, p_\ell = 0}$$

So, now given a point pl, we wanted to find the equation of the epipolar line on the right image. and later we want to find the point pr on the right image which is on this epipolar line

We know that $l^T p = 0$ for all point p (x,y,1) on the line l (a,b,c)

The line l is given by ax+by+c = 0 and a,b are normal vector of the line and -c is the distance from origin