Lecture 6 - Structure for motion

→ So far, we have understood stereo-reconstruction from 2 views:

① given 2 cameras:

$$P = K[I|0] \qquad P' = K'[R|t]$$

② Epipolar geometry: Computing the fundamental matrix F:

$$F = K'^{-T}[t]_x R K^{-1}$$

③ Correspondence search along epipolar line:

$$l' = Fx \quad \text{(2D search becomes 1d search)}$$

→ Problem statement: Structure and motion

→ Given 2(or more) images of a scene, compute the scene structure and the camera motion.

Assumptions:

① Assume calibration of cameras (K, K') are known

② Assume scene is rigid (although it would work with non-rigid scene, it would generate outliers)

③ Start with 2 views only (epipolar geometry is not known)

→ Given 2 images, we first need to find the correspondences. (Point motion)

→ Computing points to avoid aperture problem.

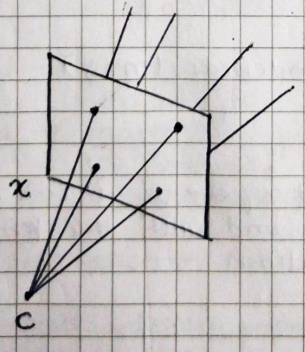
→ Work with corner, points

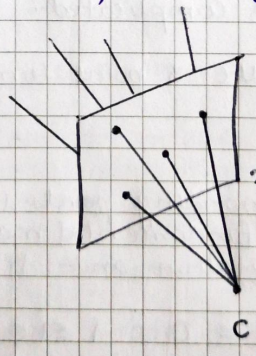
→ Due to aperture problem, we take corners to detect point correspondences and generally we use Harris corner detector

→ Once we have detected corner point in both the images, we need to track their change in position (i.e. how has a point changed in other image) Point matching.

→ Point motion → the geometric motion problem

① Given image point correspondences, $x_i \leftrightarrow x'_i$, determine R and t





→ The points in the image actually represents a ray in 3d.

→ The target is to rotate and translate the camera until the stars of rays intersect.

→ Outline of computation

① Compute the fundamental matrix "F" from point correspondences $x_i \leftrightarrow x'_i$

② Compute the camera (motion) from the fundamental matrix

$$F = K'^{-T}[t]_x R K^{-1}$$

→ obtain $P = K[I|0]$ $\quad P' = K'[R|t]$ (i.e. Position)

③ Compute the 3D structure x_i from the cameras P, P' and point correspondences $x_i \leftrightarrow x'_i$ (triangulation)

→ Camera motion.

Q/ what can be computed from point correspondences?

① Suppose we have computed $F = K'^{-T}[t]_x R K^{-1}$. can the camera motion be computed?

② F is a homogeneous matrix, so

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

i.e. the translation can only be determined upto scale

③ This is a consequence of the depth / speed ambiguity:
→ only the ratio of t and Z can be computed, since $t \to \lambda t$ and $Z \to \lambda Z$, the images are unchanged,
→ In other words:
(a) a large motion of the distant object
(b) a small motion of the nearby object

are indistinguishable from point motion alone.

④ Summary: The rotation R (3 DOF) can be determined completely, but only the translation direction (2 DOF) can be determined, not the magnitude.

→ How many points do we need as correspondences?

structure → For n points there are 3n unknowns (3d position of each point)

motion → For 2 views there are 5 unknowns (that are recoverable)

→ Each point correspondence gives 4 measurements.
→ for n points we can expect a solution if $4n \geq (3n+5)$ i.e. $n \geq 5$
but this is unstable and complicated.

→ A much more classical is the 8 point correspondences ($n=8$)

→ 8 Point - algorithm.

objective → given $n \geq 8$ corresponding points (2D) $\{x_i \leftrightarrow x'_i, i = 1 \dots n\}$, can we compute the fundamental matrix, and with this known, can we extract the rotation and translation.

$$x_i'^T F x_i = 0 \qquad 1 \leq i \leq n$$

Solution → Each point correspondence $x_i \leftrightarrow x'_i$ generates one constraint on 'F'

$$(x_i' \; y_i' \; 1) \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = 0$$

$$x_i' x_i f_{11} + x_i' y_i f_{12} + x_i' f_{13} + y_i' x_i f_{21} + y_i' y_i f_{22} + y_i' f_{23} + x_i f_{31} + y_i f_{32} + f_{33} = 0$$

⇒ Least square solution of the 8 point algorithm.

Given: n corresponding points (n is typically hundreds) with noise on their measured 2D positions in the image

→ For $n > 8$ point correspondences, set up a $n \times 9$ matrix "A" & solve:

$$Af = \begin{pmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & & & & & & & & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{pmatrix} f = 0$$

($n \times 9$)

f → 9 dimen. vector (9×1)

→ In general, there will not be an exact solution to $Af = 0$
→ A linear solution, that minimizes $\|Af\|^2$, subject to $\|f\|^2 = 1$ is obtained from the eigenvector with smallest eigenvalue ($\neq 0$) of A^TA.

→ Eg. for 8 points, A is an 8×9 matrix and f can be computed as the null vector of A, i.e. f is determined up to scale.
→ This solution do not require camera calibration matrix "K" and "K'"

⇒ Computing "F"

* As we know that "F" is a rank 2 matrix:

① To enforce that F is of rank 2, F is replaced by F' that minimizes $\|F - F'\|_F^2$ with subject to rank constraints.

② This is achieved by SVD: Let $F = U\Sigma V^T$, with

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \text{ Let } \Sigma' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

③ Then $F' = U\Sigma' V^T$ is the solution.

⇒ Automatic computation of F

① Step 1 → Computing image points (interest points) (Eg. Harris corner detector) around 100
② Step 2 → match the points b/w the images. This is usually done in 2 steps
(a) Proximity → search with disparity window
(b) Cross-correlation → on intensity neighbourhood.

⇒ Robust Estimation – RANSAC (for estimation of F)

① Compute interest points in each image and point matches.
② Randomly select 8 samples from point correspondences
③ Compute the fundamental matrix from these 8 point correspondences.
④ Measure the support: the no. of inliers within threshold distance of epipolar lines.
⑤ After N iterations, choose the F with largest no. of inliers.

Note: ① Calibrated case: 5 point RANSAC for essential matrix "E" estimation.

② Planar case: If all points lie on a plane, then the problem degenerate to estimate Homography "H".

⇒ Computing Essential matrix "E"

Essential matrix can be computed by fundamental matrix "F" by

$$F = K'^{-T}[t]_x R K^{-1} \qquad E = [t]_x R = K'^{T} F K$$

★ Instead of computing F and extracting E from it, we can compute E directly from normalised point correspondences.

↑ numerically more stable

⇒ Properties of "E":

① We already know that any essential matrix can be decomposed into a product of skew-symmetric matrix and an orthogonal matrix

$$E = [t]_x R$$

② Moreover, the skew-symmetric matrix can be expressed in:

$$[t]_x = kU \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} U^T = kU \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{orthogonal matrix}} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} U^T$$

③ Therefore, we can get the singular value decomposition of the following ~~be~~ form for any essential matrix "E".

$$E = [t]_x R = U \underbrace{\begin{pmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Rank 2}} \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}}_{\text{orthogonal matrix}} U^T R = U \Sigma V^T$$

④ ★ As we know, that E is a rank 2 matrix:

(a) To enforce that E is of rank 2 matrix with 2 similar singular values, E is replaced by E' that minimizes $\|E - E'\|_F^2$ with subjects to rank constraints.

(b) This is achieved by SVD. Let $E = U \Sigma V^T$, with

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \qquad \Sigma' = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{with } \sigma = \frac{\sigma_1 + \sigma_2}{2}$$

(c) Then $E' = U \Sigma' V^T$ is the solution.

⇒ Decomposing E - SVD.

$$E = U \Sigma V^T \qquad E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

with $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ we get.

$[t]_x = UZU^T$ and $R = UWV^T$ or $R = UW^TV^T$ which are the solutions of E

⇒ Decomposing E - Extraction of Cameras.

① From the essential matrix, compute the translational vector "t" as the left-null vector. This determines t upto scale.

$$E^T t = 0$$

② Compute the Rotation matrix R from E, there are 2 solutions "R_1" and "R_2".

③ Set $P = K[I|0]$ for the first camera.

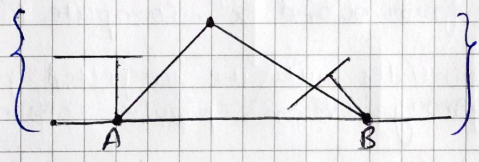
④ For the 2nd camera, we have 4 solutions.

$$P' = K'[R_1|Mt] \qquad P' = K'[R_1|-Mt] \qquad P' = K'[R_2|Mt]$$

$$P' = K'[R_2|-Mt]$$

* The depth has to be positive (in front) in order to be a solution and be considered as a point in 3d.

we do the same for many no. of points.



Front of the camera. Depth is positive.

⇒ What happens if we have more than 2 views?

① The two view ambiguities do not get worse
↳ There is overall slight ambiguity.

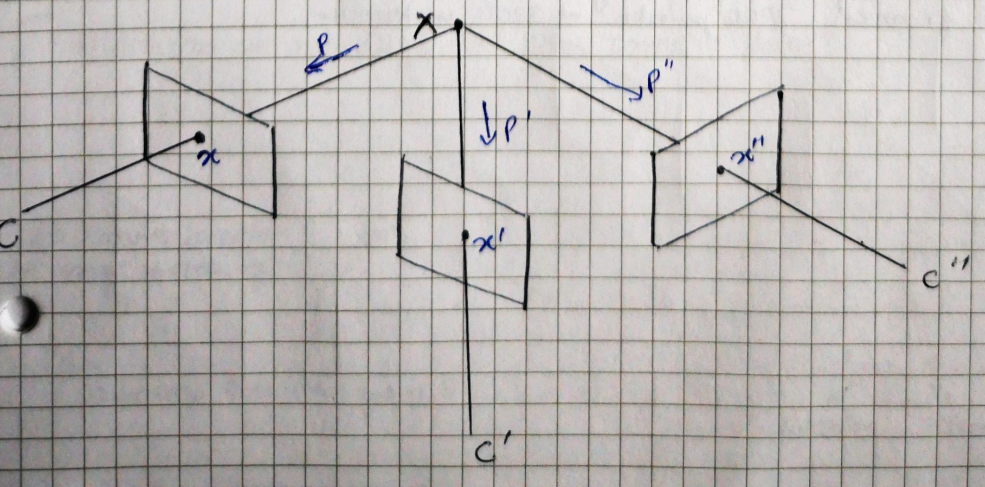
② matching: matches can be verified.

③ Estimation → Accuracy increased by using more measurements.

⇒ Notation for 3 or more views.

① for 3 views, the camera are P, P' and P'' and a 3d point is imaged as

$$x = PX \quad x' = P'X \quad x'' = P''X$$

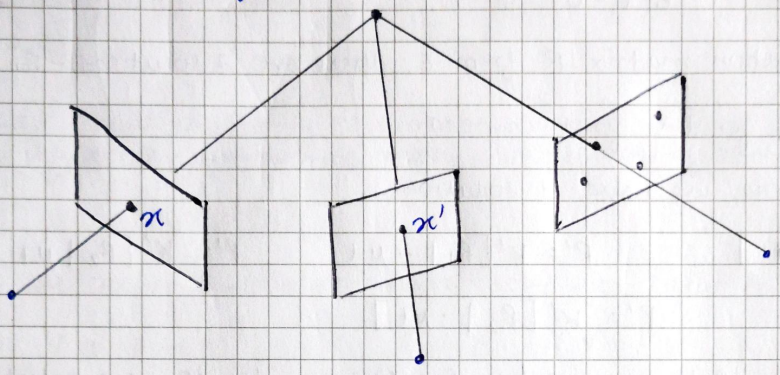


⑤ For m views, a point X_j is imaged in the "i"th view as

$$x^i_j = P^i X_j$$

③ Correspondences over 3 or more views

① Given: the cameras P, P' and P'', the matching point x and x'
② Find: the matching point in the 3rd view.



Algorithm:

① Compute the 3d point from x and x'. Compute P'', project it into the 3rd view.
② The matching point coincides with the projected point.
③ Point constraints is stronger than epipolar constraint.

⇒ Structure and motion - Problem statement

Given: m matching image points x^i_j over m views

Find: the cameras P_i and the 3D point X_j such that $x^i_j = P^i X_j$

$$\min_{P^i X_j} \sum_{j \in \text{points}} \sum_{i \in \text{views}} \underbrace{d(x^i_j, P^i X_j)^2}_{\text{as small as possible}}$$

} minimize using gradient descent

① No. of parameters → (a) for each camera there are 6 parameters (3R, 3T)
(b) for each 3D point there are 3 parameters

② a total of $6m + 3n$ parameters must be estimated
eg. 50 frames, 1000 points → 3300 unknowns.