B

EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2022**

Lecture 7B: (Two-View Geometry, Calibration)

Scribes: Baiyu Shi, Lawrence Yunliang Chen

7.1 Summary of Two-View Geometry from Lecture 7A

We consider the setting where the camera is calibrated. Given two different views of the same scene, we want to recover the unknown camera poses as well as the 3D scene structure. To do so, we use the pinhole camera model. In this model we define 3D points $X = [X, Y, Z, 1]^T \in \mathbb{R}^4$, image points $x = [x, y, 1]^T \in \mathbb{R}^3$, perspective projection $\lambda x = X$ where λ is depth, rigid body motion $\Pi = [R, T] \in \mathbb{R}^{3\times 4}$, and rigid body motion along with perspective projection $\lambda x = \Pi X = [R, T]X$.

As shown in Figure 7.1, we can relate two different images of the same point X through $\lambda_2 x_2 = R\lambda_1 x_1 + T$.

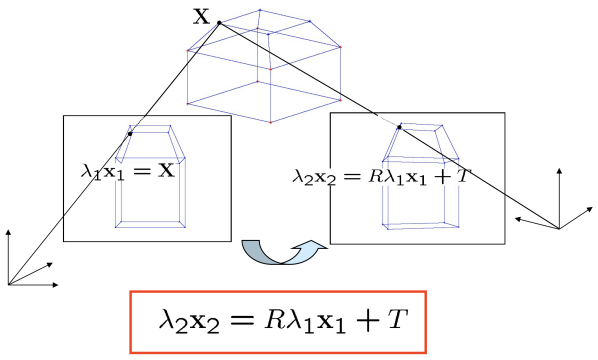


Figure 7.1: Visualization for the two-view geometry. The point X relative to the two camera frames are related by a rigid-body transformation by $X_2 = RX_1 + T$.

The only knowns in the equation $\lambda_2 x_2 = R\lambda_1 x_1 + T$ are x_1 and x_2. The other variables, λ_1, λ_2, R, and T are unknown. Through epipolar geometry we can algebraically eliminate depth from our relation equation and get

$$x_2^T \hat{T} R x_1 = 0.$$

In particular, the matrix $E = \hat{T}R \in \mathbb{R}^{3\times 3}$ in the epipolar constraint equation is called the essential matrix. It encodes the relative pose between the two cameras. The epipolar constraint is also called the essential constraint.

The Essential matrix $E = \widehat{T}R$ has some special properties. In particular, the space of all essential matrices is 5 dimensional:

- 3 Degrees of Freedom - Rotation
- 2 Degrees of Freedom - Translation (up to scale!).

We also have the following two theorems:

Theorem 1 (Essential Matrix Characterization) *A non-zero matrix E is an essential matrix iff its SVD: $E = U\Sigma V^T$ satisfies: $\Sigma = \text{diag}([\sigma_1, \sigma_2, \sigma_3])$ with $\sigma_1 = \sigma_2 \neq 0$ and $\sigma_3 = 0$ and $U, V \in SO(3)$. In other words,*

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Theorem 2 (Pose Recovery) *There are two relative poses (R, T) with $T \in \mathcal{R}^3$ and $R \in SO(3)$ corresponding to a non-zero matrix essential matrix.*

$$E = U\Sigma V^T$$
$$\left(\widehat{T}_1, R_1\right) = \left(UR_Z\left(+\frac{\pi}{2}\right)\Sigma U^T, UR_Z^T\left(+\frac{\pi}{2}\right)V^T\right)$$
$$\left(\widehat{T}_2, R_2\right) = \left(UR_Z\left(-\frac{\pi}{2}\right)\Sigma U^T, UR_Z^T\left(-\frac{\pi}{2}\right)V^T\right)$$
$$\Sigma = \text{diag}([1,1,0]) \quad R_z\left(+\frac{\pi}{2}\right) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Moreover, $(R_2, T_2) = \left(e^{\widehat{u}\pi}R_1, -T_1\right)$ are called twisted pair ambiguity.

Using these two theorems, a general procedure to estimate R and T is as follows:

1. Step 1: Given n pairs of image correspondences, find an essential matrix E that the epipolar error is minimized

$$\min_E \sum_{j=1}^{n} \left(\mathbf{x}_2^{jT} E \mathbf{x}_1^j\right)^2$$

(This comes from the requirement that $\mathbf{x}_2^T E \mathbf{x}_1 = 0$.)

- Denote $\mathrm{a} = \mathrm{x}_1 \otimes \mathrm{x}_2$

$$\mathbf{a} = [x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2]^T$$
$$E^s = [e_1, e_4, e_7, e_2, e_5, e_8, e_3, e_6, e_9]^T$$

- Rewrite $\mathbf{a}^T E^s = 0$
- Collect constraints from all points

$$\chi E^s = 0$$

$$\min_E \sum_{j=1}^{n} \left(\mathbf{x}_2^{jT} E \mathbf{x}_1^j\right)^2 \longrightarrow \min_{E^s} \|\chi E^s\|^2$$

- Note that if rank $\left(\chi^T\chi\right) < 8$, it will get a degenerate configuration.

2. Step 2: Then use SVD and Theorem 2 to decompose E into R, T.

7.2 Two View 8-point Linear Algorithm

An issue of the above procedure is that the estimated E from step 1 is in general not an essential matrix: that is, it cannot be exactly decomposed into a skew-symmetric matrix $\hat{T}$ and a rotation matrix R due to noises and errors in correspondence. The way to address this is to project E onto a manifold called the Essential manifold. We have an elegant theorem as stated below.

Theorem 3 (Projection to Essential Manifold) *If the SVD of a matrix $F \in \mathcal{R}^{3\times 3}$ is given by $F = U\,\mathrm{diag}\,(\sigma_1, \sigma_2, \sigma_3)\,V^T$ then the essential matrix E which minimizes the Frobenius distance $\|E - F\|_f^2$ is given by $E = U\,\mathrm{diag}(\sigma, \sigma, 0)V^T$ with $\sigma = \frac{\sigma_1+\sigma_2}{2}$.*

Using Theorem 3, we can refine the procedure in the previous section to do projection onto the essential manifold after linear least square estimation (LLSE) as follows:

$$E = \left\{\widehat{T}R \mid R \in SO(2), T \in S^2\right\}$$

1. Step 1: Solve the LLSE problem:
$$\min_E \sum_{j=1}^{n} \left(\mathbf{x}_2^{jT} E \mathbf{x}_1^j\right)^2$$
2. Step 2: Let F be the argmin found in Step 1. We project F onto the essential manifold using SVD:
$$\begin{aligned} F &= U\Sigma V^T \\ \Sigma' &= \mathrm{diag}(1, 1, 0) \\ E &= U\Sigma' V^T \end{aligned}$$
3. Step 3: Recover the unknown pose:
$$(\widehat{T}, R) = \left(UR_Z\left(\pm\frac{\pi}{2}\right)\Sigma U^T, UR_Z^T\left(\pm\frac{\pi}{2}\right)V^T\right).$$

7.2.1 Pose Recovery from Essential Matrix

There are exactly two pairs of (R, T) corresponding to essential matrix E and also two pairs of (R, T) corresponding to essential matrix -E since the equation is homogeneous. We will use the positive depth constraint to disambiguate the physically impossible solutions. We can use either the linear 8-point algorithm or the nonlinear 5-point algorithms but the latter may yield up to 10 solutions. For this recovery method to work, we need the translation to be non-zero and the points to be in general position rather than on the same plane or quadratic surfaces, which are degenerate configurations that is explained in 7.3.

7.2.2 3D Structure Recovery

$$\lambda_2 x_2 = R\lambda_1 x_1 + \gamma T$$

We can eliminate one of the scales by taking cross product with x_2, then we get:

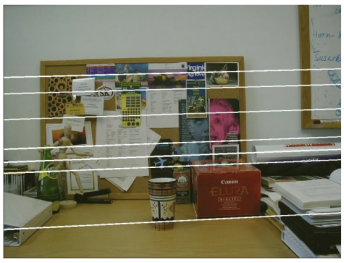
$$\lambda_1^j \hat{x_2^j} R x_1^j + \gamma \hat{x_2^j} T = 0, j = 1, 2, 3, ...n$$

Then we solve the LLSE problem:

$$M^j\overline{\lambda^j} = \begin{bmatrix} \hat{x_2^j}Rx_1^j & \hat{x_2^j}T \end{bmatrix} \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix} = 0$$

This is just the multiview version of the epipolar constraint:

$$x_2^{jT}\hat{T}Rx_1^j = 0$$



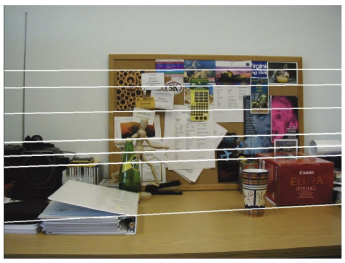


Figure 7.2: We can use epipolar geometry to tell where the corresponding image coordinate in another camera frame could lie, which are the white lines.

7.3 Two-view Geometry – Planar Case, Homography

Suppose we happen to pick points from the same plane, it becomes the planar degenerate configs. We denote the normal vector to the plane as N. And suppose that the first camera's frame is related to the second camera's frame by (R, T). Then we have:

$$aX + bY + cZ = d$$
$$\frac{1}{d}N^T\boldsymbol{X} = 1$$
$$\lambda_2 x_2 = R\lambda_1 x_1 + T$$
$$\lambda_2 x_2 = (R + \frac{1}{d}TN^T)\lambda_1 x_1$$
$$x_2 \sim Hx_1$$
$$H = \left(R + \frac{1}{d}TN^T\right)$$

H is the linear mapping relating two corresponding planar points in two views.

We can eliminate the depth information of $x_2 \sim Hx_1$ to obtain the equation $\widehat{\mathbf{x}_2}H\mathbf{x}_1 = 0$.
Linear estimation of H: $H_L = \lambda H$.
Normalization of H: $H = H_L/\sigma_3$.
Decomposition of $H = \left(R + \frac{1}{d}TN^T\right)$ into 4 solutions:

$R_1 = W_1 U_1^T$ $N_1 = \widehat{v_2} u_1$ $\frac{1}{d} T_1 = (H - R_1) N_1$	$R_3 = R_1$ $N_3 = -N_1$ $\frac{1}{d} T_3 = -\frac{1}{d} T_1$	$R_2 = W_2 U_2^T$ $N_2 = \widehat{v_2} u_2$ $\frac{1}{d} T_2 = (H - R_2) N_2$	$R_4 = R_2$ $N_4 = -N_2$ $\frac{1}{d} T_4 = -\frac{1}{d} T_2$

$$H^T H = V \Sigma V^T \quad V = [v_1, v_2, v_3] \quad \sum = \text{diag}\left(\sigma_1^2, \sigma_2^2, \sigma_3^2\right)$$

$$u_1 \doteq \frac{\sqrt{1-\sigma_3^2} v_1 + \sqrt{\sigma_1^2 - 1} v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \quad u_2 \doteq \frac{\sqrt{1-\sigma_3^2} v_1 - \sqrt{\sigma_1^2 - 1} v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}}$$

$$U_1 = [v_2, u_1, \widehat{v_2} u_1], \quad W_1 = \left[H v_2, H u_1, \widehat{H v_2} H u_1\right]$$

$$U_2 = [v_2, u_2, \widehat{v_2} u_2], \quad W_2 = \left[H v_2, H u_2, \widehat{H v_2} H u_2\right]$$

Summing up, to recover motion and pose from homography, we need to:

1. Step 1: Have at least 4 point correspondences: $\widehat{x}_2^j H x_1^j = 0$
2. Step 2: Use the fact that H_L^s is a nullspace of χ to approximate the homography matrix, where a is rows of χ:

$$\chi H_L^s = 0$$
$$a = x_1^j \otimes \widehat{x_2^j}$$

3. Step 3: Normalize and Decompose the homography matrix and then choose solutions that satisfy the positive depth constraint:

$$H = H_L / \sigma_3$$
$$H^T H = V \Sigma V^T$$

We can also transform between Homography matrix and Essential matrix.
With two generic points,

$$E = \widehat{T} H, T \sim \hat{l}_2^1 l_2^2$$

With three planar points,

$$H = \widehat{T^T} E + T v^T, v \in R^3$$

Some special cases of Homography:

1. Rotation case:

$$\lambda_2 x_2 = R \lambda_1 x_1$$
$$\widehat{x_2} R x_1 = 0$$

2. Reflection Case, symmetrical object:

$$g = (R, 0)$$
$$R = \text{diag}(-1, 1, 1])$$

3. Translation Case

Figure 7.3: Different views related by only rotation stitched together. This also explains why we can get good panoramic photos of the wild, e.g. when on top of a mountain. It is because the distance to the plane is too far away and large d makes T negligible and H almost the same as R, making it different camera views only differ by a rotation.

7.4 Two View Motion and Structure Recovery Summary

For two discrete views with general motion and general structure:

1. Use 8 points (in reality we may use many more correspondences and let them vote for the best essential matrix) to estimate the essential matrix.
2. Decompose the essential matrix.
3. Use positive depth constraint (points constructed should be in front of the camera) to select the appropriate solutions and conduct 3D reconstruction.

For two discrete views with general motion but planar structure:

1. Use 4 points to estimate the homography matrix H (may also utilize the special homography).
2. Normalize and decompose the H matrix and select the solutions that satisfy the positive depth constraint.
3. Recover 3D structure and camera poses.

7.5 Uncalibrated Geometry & Stratification

7.5.1 Uncalibrated Epipolar Geometry

Let $\mathbf{X} = [X, Y, Z, W]^T \in \mathbb{R}^4, \quad (W = 1)$ be the world coordinates of a point. Recall that for a calibrated camera, we have

- Image plane coordinates: $\mathbf{x} = [x, y, 1]^T$
- Camera extrinsic parameters: $g = (R, T)$
- Perspective projection: $\lambda \mathbf{x} = [R, T]\mathbf{X}$

In contrast, for an uncalibrated camera, we need to multiply the right hand side by the camera calibration matrix K, and we get:

- Pixel coordinates: $\mathbf{x}' = K\mathbf{x}$
- Projection matrix: $\lambda \mathbf{x}' = \Pi \mathrm{X} = [KR, KT]\mathbf{X}$

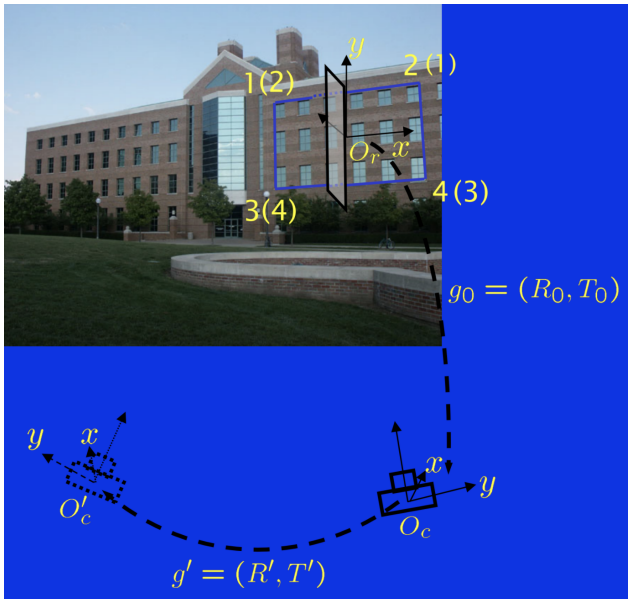


Figure 7.4: In this figure, we have utilized the symmetry about the y axis and even though we only have one camera view, with use of symmetry and reflection matrix, we can create a "virtual" camera view with coordinate frame as O'_c.

Remark. In practice, there are many different settings in which we have some partial knowledge about the camera and the scene, and the approach to solving them will depend on the knowledge we have. Here are a couple of cases:

- If K is known, then we are back to the calibrated case $\mathbf{x} = K^{-1}\mathbf{x}'$
- If K is unknown, there are three possible scenarios:
  - Calibration with complete scene knowledge (a rig) - estimate
  - Uncalibrated reconstruction despite the lack of knowledge of K
  - Autocalibration (recover K from uncalibrated images)
- Use partial knowledge about K
  Parallel lines, vanishing points, planar motion, constant intrinsic matrix

Similar to the derivation of the epipolar geometry involving the essential matrix, we can derive the epipolar constraint when the calibration matrix K is involved.

$$\lambda_2 K\mathbf{x}_2 = KR\lambda_1\mathbf{x}_1 + KT$$

Let $T' = KT, K\mathbf{x}_1 = \mathbf{x}'_1, K\mathbf{x}_2 = \mathbf{x}'_2$, we have

$$\lambda_2\mathbf{x}'_2 = KR\lambda_1 K^{-1}\mathbf{x}'_1 + T'$$

Multiplying both sides by $\widehat{T'} = \widehat{KT} \propto K^{-T}\widehat{T}K^{-1}$, we get

$$\lambda_2\widehat{T'}\mathbf{x}'_2 = \widehat{T'}KRK^{-1}\mathbf{x}'_1$$

Multiplying both sides by $\mathbf{x}'_2$, we get

$$\mathbf{x}_2'^T\widehat{T'}KRK^{-1}\mathbf{x}'_1 = 0$$
$$\mathbf{x}_2'^T K^{-T}\widehat{T}RK^{-1}\mathbf{x}'_1 = 0.$$

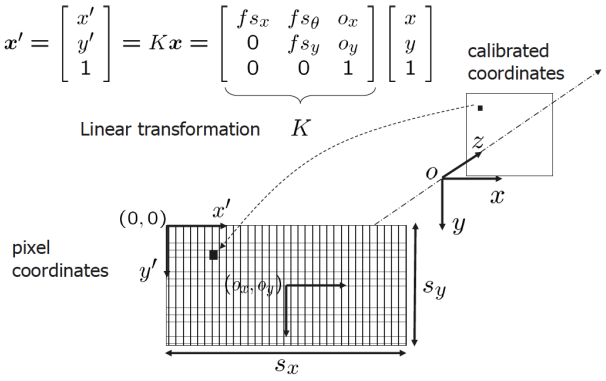


Figure 7.5: Linear transformation between the pixel coordinates of an uncalibrated camera and the calibrated coordinates.

To summarize, the Epipolar constraint is

$$\mathrm{x}_2'^T \underbrace{K^{-T}\widehat{T}RK^{-1}}_{F} \mathbf{x}_1' = 0$$

$$\mathbf{x}_2'^T F \mathbf{x}_1' = 0.$$

The Fundamental matrix is

$$F = K^{-T}\widehat{T}RK^{-1} = \widehat{T}'KRK^{-1}.$$

The relationships between the epipolar lines l_1, l_2, epipoles e_1, e_2, and fundamental matrix are similar to those with the essential matrix. Table 7.1 shows a side-by-side comparison.

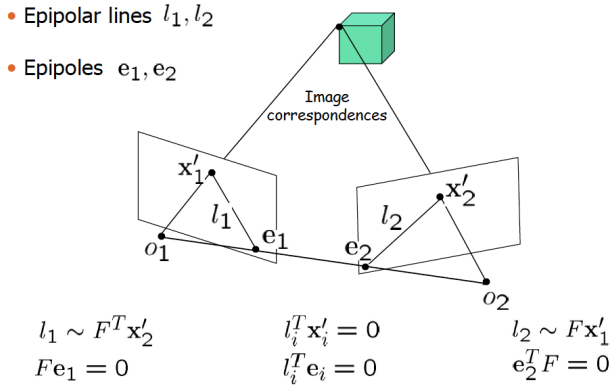


Figure 7.6: Epipolar geometry in the uncalibrated setting.

7.5.2 Geometric Stratification

Facing with an uncalibrated problem where we don't know K, we need to use the epipolar geometry and constraint $(x_2')^T F x_1' = 0$ to recover the fundamental matrix F. The geometric picture is still the same, as corresponding features from two views still must be coplanar per the epipolar geometry.

	Calibrated case	Uncalibrated case
Image point	x	$' = Kx$
Camera (motion)	$g = (R, T)$	$g' = \left(KRK^{-1}, KT\right)$
Epipolar constraint	$x_2^T E x_1 = 0$	$\left(x_2'\right)^T F x_1' = 0$
Fundamental matrix	$E = \widehat{T}R$	$F = \widehat{T'}KRK^{-1}, T' = KT$
Epipoles	$Ee_1 = 0, e_2^T E = 0$	$Fe_1 = 0, e_2^T F = 0$
Epipolar lines	$\ell_1 = E^T x_2, \ell_2 = E x_1$	$\ell_1 = F^T x_2', \ell_2 = F x_1'$
Decomposition	$E \mapsto [R, T]$	$F \mapsto \left[\left(\widehat{T'}\right)^T F, T'\right]$
Reconstruction	Euclidean: $\boldsymbol{X}_e$	Projective: $\boldsymbol{X}_p = H\boldsymbol{X}_e$

Table 7.1: Summary of Two-View Geometry.

The procedure is known as stratification, where we first reconstruct an X_p, but it will be different from the true structure X_e by a general projective transformation H such that $X_p = HX_e$. We can decompose H into an affine part H_a and a projective part H_p, where $H = H_p H_a$. We can thus restore the parallel line using the inverse of the projective transformation to get $X_a = H_p^{-1} X_p$ (called "affine upgrade") and then restore the angles using $X_e = H_a^{-1} X_a$ (called "Euclidean upgrade"). The procedure is summarized in Table 7.2.

	Camera projection	3-D structure
Euclidean	$\Pi 1e = [K, 0], \Lambda_{2e} = [KR, KT]$	$X_e = g_e X = \begin{bmatrix} R_e & T_e \\ 0 & 1 \end{bmatrix} X$
Affine	$\Pi 2a = [KRK^{-1}, KT],$	$X_a = H_a X_e = \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix} X_e$
Projective	$\Pi 2p = [KRK^{-1} + KTv^T, v_4 KT],$	$X_p = H_p X_a = \begin{bmatrix} I & 0 \\ -v^T v_4^{-1} & v_4^{-1} \end{bmatrix} X_a$

Table 7.2: Summary of Two-View Geometry.