Lecture 08: Multi-View Geometry II

1 Two-view Structure from Motion

The camera relative pose is unknown: this is e.g. the case when the two images are taken from the same camera but at different times and positions.

Problem Formulation

Given n point correspondences between two images, $\{p_1^i = (u_1^i, v_1^i),\ p_2^i = (u_2^i, v_2^i)\}$, simultaneously estimate the 3D points P^i, the camera relative-motion parameters (R, T), and the camera intrinsics K_1, K_2 that satisfy the equations:

$$\begin{aligned} \lambda_1 \cdot \begin{pmatrix} u_1^i \\ v_1^i \\ 1 \end{pmatrix} &= K_1 \cdot [I|0] \cdot \begin{pmatrix} X_w^i \\ Y_w^i \\ Z_w^i \\ 1 \end{pmatrix}, \\ \lambda_2 \cdot \begin{pmatrix} u_2^i \\ v_2^i \\ 1 \end{pmatrix} &= K_2 \cdot [R|T] \cdot \begin{pmatrix} X_w^i \\ Y_w^i \\ Z_w^i \\ 1 \end{pmatrix} \end{aligned} \tag{1.1}$$

One has two cases then: the case of calibrated cameras and the case uncalibrated cameras.

1.1 Calibrated Cameras (K_1, K_2 known)

For convenience, we use *normalized image coordinates*:

$$\begin{pmatrix} \bar{u} \\ \bar{v} \\ 1 \end{pmatrix} = K^{-1} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{1.2}$$

We want to find R, T, P^i which satisfy

$$\begin{aligned} \lambda_1 \cdot \begin{pmatrix} \bar{u}_1^i \\ \bar{v}_1^i \\ 1 \end{pmatrix} &= K_1 \cdot [I|0] \cdot \begin{pmatrix} X_w^i \\ Y_w^i \\ Z_w^i \\ 1 \end{pmatrix}, \\ \lambda_2 \cdot \begin{pmatrix} \bar{u}_2^i \\ \bar{v}_2^i \\ 1 \end{pmatrix} &= K_2 \cdot [R|T] \cdot \begin{pmatrix} X_w^i \\ Y_w^i \\ Z_w^i \\ 1 \end{pmatrix}. \end{aligned} \tag{1.3}$$

Scale Ambiguity: If we rescale the entire scene by a constant factor (i.e. similarity transformation), the projections (in pixels) of the scene points in both images remain the **same** (because the angles remain the same). This has the following effects:

- In *monocular* vision it is **not possible** to recover the absolute scale of the scene.

- In *stereo vision*, only **5 degrees of freedom** are measurable:
  - 3 parameters to describe the **rotation**.
  - 2 parameters for the **translation up to a scale** (we can only compute the direction of translation but not its length (magnitude)).

How many knowns and unknowns are we dealing with?

- $4n$ knowns: n correspondences, each one (u_1^i, v_1^i) and (u_2^i, v_2^i), $i = 1, \dots, n$.
- $5 + 3n$ unknowns: 5 for the motion up to a scale (3 rotation and 2 translation) and $3n$ which is the number of coordinates of the n 3D points.

In order for the problem to have a solution it should hold

$$\begin{aligned} 4n &\geq 5 + 3n \\ \Rightarrow n &\geq 5. \end{aligned} \tag{1.4}$$

The first analytical solution for 5 points was given by Kruppa in 1913 (10 degree order polynomial, up to 10 solutions with complex ones). In the following, we are exploring the possibility of solving the estimation of the relative motion (R, T) independently of the estimation of the structure (3D points).
Let's define the **cross product** as a matrix multiplication

$$a \times b = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \cdot \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = [a]_x \cdot b \tag{1.5}$$

1.1.1 Epipolar Geometry

Considering Figure 1, one notices the two vectors

$$\bar{p}_1 = \begin{pmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{pmatrix}, \quad \bar{p}_2 = \begin{pmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{pmatrix}. \tag{1.6}$$

We can observe that p_1, p_2, T are coplanar, i.e.:

$$\begin{aligned} p_2^T \cdot n &= 0 \\ p_2^T \cdot (T \times p_1') &= 0 \\ p_2^T \cdot (T \times (R p_1)) &= 0 \\ p_2^T \cdot [T]_\times \cdot R p_1 &= 0 \\ p_2^T \cdot E \cdot p_1 &= 0 \text{ is the epipolar constraint, where } E = [T]_\times \cdot R \text{ is the } \textbf{essential matrix}. \end{aligned} \tag{1.7}$$

This is also called the Longuet-Higgins equation. Applying the constraints results in *four different* solutions for R and T.

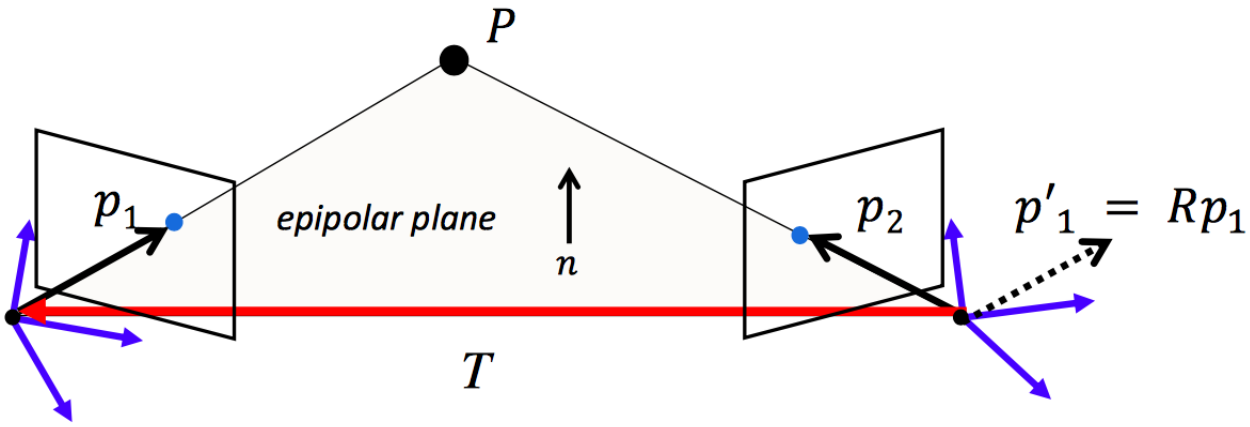


Figure 1: Epipolar constraint

Example 1. You are given the rotation matrix

$$R = \mathbb{I}_{3\times 3}, \tag{1.8}$$

and the translation vector

$$T = \begin{pmatrix} -b \\ 0 \\ 0 \end{pmatrix}. \tag{1.9}$$

The essential matrix for this formulation reads

$$\begin{aligned} E &= [T]_\times \cdot R \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & b \\ 0 & -b & 0 \end{pmatrix} \cdot \mathbb{I}_{3\times 3} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & b \\ 0 & -b & 0 \end{pmatrix}. \end{aligned} \tag{1.10}$$

1.1.2 How to compute the Essential Matrix

Kruppa's solution (with at least 5 correspondences) is not efficient. In 1996, Philipp proposed an iterative solution. In 2004, the first efficient non iterative solution was proposed. This uses **Groebner Decomposition**. The first popular solution uses 8 points and is called the **8 point algorithm or Longuet-Higgins algorithm** (still used in NASA rovers).

The 8-point Algorithm

The Essential matrix is defined by

$$\bar{p}_2^T \cdot E \cdot \bar{p}_1 = 0. \tag{1.11}$$

Each pair of point correspondences provides a linear equation. For n points we can write

$$\underbrace{\begin{pmatrix} \bar{u}_2^1 \cdot \bar{u}_1^1 & \bar{u}_2^1 \cdot \bar{v}_1^1 & \bar{u}_2^1 & \bar{v}_2^1 \cdot \bar{u}_1^1 & \bar{v}_2^1 \cdot \bar{v}_1^1 & \bar{v}_2^1 & \bar{u}_1^1 & \bar{v}_1^1 & 1 \\ \bar{u}_2^2 \cdot \bar{u}_1^2 & \bar{u}_2^2 \cdot \bar{v}_1^2 & \bar{u}_2^2 & \bar{v}_2^2 \cdot \bar{u}_1^2 & \bar{v}_2^2 \cdot \bar{v}_1^2 & \bar{v}_2^2 & \bar{u}_1^2 & \bar{v}_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{u}_2^n \cdot \bar{u}_1^n & \bar{u}_2^n \cdot \bar{v}_1^n & \bar{u}_2^n & \bar{v}_2^n \cdot \bar{u}_1^n & \bar{v}_2^n \cdot \bar{v}_1^n & \bar{v}_2^n & \bar{u}_1^n & \bar{v}_1^n & 1 \end{pmatrix}}_{Q \text{ (known)}} \cdot \underbrace{\begin{pmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{31} \\ e_{32} \\ e_{33} \end{pmatrix}}_{\bar{E} \text{ unknown}} = 0 \tag{1.12}$$

This problem can be written as

$$Q \cdot \bar{E} = 0. \tag{1.13}$$

One has two types of solution:

- **Minimal Solution**
  - $Q_{n\times 9}$ should have rank 8 to have unique (up to scale) non trivial solution $\bar{E}$.
  - Each point correspondence provides 1 independent equation.
  - Thus, 8 point correspondences are needed.
- **Over-determined Solution**
  - $n > 8$ points.
  - A solution is to minimize $||Q \cdot \bar{E}||^2$ subject to the constraint $||\bar{E}||^2 = 1$. The solution is the eigenvector corresponding to the smallest eigenvalue of matrix $Q^T \cdot Q$.
  - This can be solved with Singular Value Decomposition.
- **Degenerate Solution** if 3D points are coplanar. There is the 5 point algorithm which holds also for coplanar points.

Interpretation

With the algorithm we try to minimize the **algebraic error**

$$\sum_{i=1}^{N} (\bar{p}_2^{i^T} \cdot E \cdot \bar{p}_1^i)^2, \tag{1.14}$$

where

$$\bar{p}_2^T \cdot E \cdot \bar{p}_1 = ||\bar{p}_2|| \cdot ||E \cdot \bar{p}_1|| \cdot \cos(\theta) \tag{1.15}$$

which is not zero if p_1, p_2, T are not coplanar. When extracting solutions, four results are available. We look only at results with points **in front of both cameras** (**Cheirality Constraint**).

1.2 Uncalibrated Cameras (K_1, K_2 unknown)

It holds

$$\bar{p}_2^T \cdot E \cdot \bar{p}_1 = 0, \tag{1.16}$$

where

$$\begin{pmatrix} \bar{u}_1^i \\ \bar{v}_1^i \\ 1 \end{pmatrix} = K_1^{-1} \cdot \begin{pmatrix} u_1^i \\ v_1^i \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \bar{u}_2^i \\ \bar{v}_2^i \\ 1 \end{pmatrix} = K_2^{-1} \cdot \begin{pmatrix} u_2^i \\ v_2^i \\ 1 \end{pmatrix}. \tag{1.17}$$

By rewriting the constraint, one obtains

$$\begin{aligned} \begin{pmatrix} u_2^i \\ v_2^i \\ 1 \end{pmatrix}^T \cdot K_2^{-T} \cdot E \cdot K_1^{-1} \cdot \begin{pmatrix} u_1^i \\ v_1^i \\ 1 \end{pmatrix} = 0 \\ \begin{pmatrix} u_2^i \\ v_2^i \\ 1 \end{pmatrix}^T \cdot F \cdot \begin{pmatrix} u_1^i \\ v_1^i \\ 1 \end{pmatrix} = 0, \end{aligned} \tag{1.18}$$

where F is the **fundamental matrix**, which can be computed as

$$F = K_2^{-T} \cdot E \cdot K_1^{-1} = K_2^{-T} \cdot [T]_x \cdot R \cdot K_1^{-1}. \tag{1.19}$$

The same 8-point algorithm can be used to compute the fundamental matrix:

$$\underbrace{\begin{pmatrix} \bar{u}_2^1 \cdot \bar{u}_1^1 & \bar{u}_2^1 \cdot \bar{v}_1^1 & \bar{u}_2^1 & \bar{v}_2^1 \cdot \bar{u}_1^1 & \bar{v}_2^1 \cdot \bar{v}_1^1 & \bar{v}_2^1 & \bar{u}_1^1 & \bar{v}_1^1 & 1 \\ \bar{u}_2^2 \cdot \bar{u}_1^2 & \bar{u}_2^2 \cdot \bar{v}_1^2 & \bar{u}_2^2 & \bar{v}_2^2 \cdot \bar{u}_1^2 & \bar{v}_2^2 \cdot \bar{v}_1^2 & \bar{v}_2^2 & \bar{u}_1^2 & \bar{v}_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{u}_2^n \cdot \bar{u}_1^n & \bar{u}_2^n \cdot \bar{v}_1^n & \bar{u}_2^n & \bar{v}_2^n \cdot \bar{u}_1^n & \bar{v}_2^n \cdot \bar{v}_1^n & \bar{v}_2^n & \bar{u}_1^n & \bar{v}_1^n & 1 \end{pmatrix}}_{Q \text{ (known)}} \cdot \underbrace{\begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix}}_{\bar{F} \text{ unknown}} = 0 \tag{1.20}$$

There are **orders of magnitude** of difference, which leads to poor results with least-squares. How to solve this issue?

1.2.1 Normalized 8-point algorithm

This estimates the Fundamental matrix on a set of **normalized correspondences** (with better numerical properties) and then **unnormalizes** the result to obtain the fundamental matrix for the original given correspondences.

Idea: Transform image coordinates so that they are in the range $[-1, 1] \times [-1, 1]$. One way is to apply the rescaling and shift proposed in Figure 2 A more popular one is to rescale the two point sets such that the centroid of each is 0 and the mean standard deviation $\sqrt{2}$. This can be done for every point as follows

$$\hat{p}^i = \frac{\sqrt{2}}{\sigma} \cdot (p^i - \mu), \tag{1.21}$$

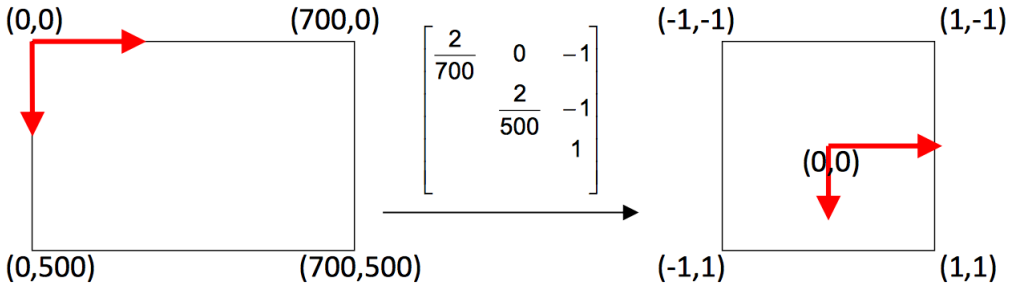


Figure 2: Shift for normalized algorithm.

where

$$\mu = \frac{1}{N}\sum_{i=1}^{n} p^i \tag{1.22}$$

is the centroid of the set and $\sigma = \frac{1}{N}\sum_{i=1}^{n} ||p^i - \mu||^2$ is the mean standard deviation. This transformation can be expressed in matrix form

$$\hat{p}^i = \begin{pmatrix} \frac{\sqrt{2}}{\sigma} & 0 & -\frac{\sqrt{2}}{\sigma}\mu^x \\ 0 & \frac{\sqrt{2}}{\sigma} & -\frac{\sqrt{2}}{\sigma}\mu^y \\ 0 & 0 & 1 \end{pmatrix} \cdot p^i. \tag{1.23}$$

The algorithm at the end reads

1. Normalize point correspondences: $\hat{p}_1 = B_1 \cdot p_1$, $\hat{p}_2 = B_2 \cdot p_2$.
2. Estimate $\hat{F}$ using normalized coordinates $\hat{p}_1, \hat{p}_2$.
3. Compute F from $\hat{F}$:

$$\begin{aligned} \hat{p}_2^T \cdot \hat{F} \cdot \hat{p}_1 &= 0 \\ p_2^T \cdot B_2^T \cdot \hat{F} \cdot B_1 \cdot p_1 &= 0 \\ \Rightarrow F &= B_2^T \cdot \hat{F} \cdot B_1 \end{aligned} \tag{1.24}$$

1.2.2 Error Measures

The quality of the estimated fundamental matrix can be measured by looking at cost functions. The first is defined using the Epipolar Constraint

$$err = \sum_{i=1}^{N} (\bar{p_2^i}^T \cdot E \cdot p_1^i)^2. \tag{1.25}$$

This error will exactly be 0 if computed from 8 points. For more points this will not be 0 because of image noise/outliers. Better methods are

Directional Error

Sum of the angular distances to the Epipolar plane: $err = \sum_i (\cos(\theta_i))^2$, where

$$cos(\theta) = \left(\frac{p_2^T \cdot E \cdot p_1}{||p_2^T|| \cdot ||E \cdot p_1||} \right). \tag{1.26}$$

Squared Epipolar-Line-to-point Distances

$$err = \sum_{i=1}^{N} d^2(p_1^i, l_1^i) + d^2(p_2^i, l_2^i). \tag{1.27}$$

Cheaper than reprojection error: does not require point triangulation!

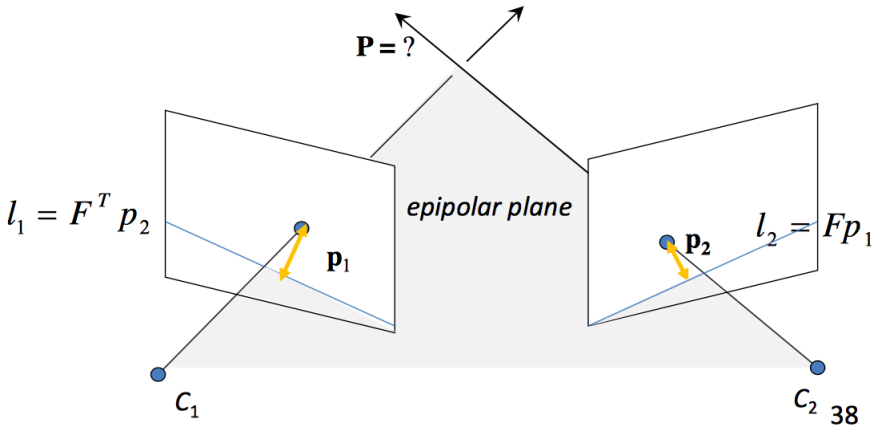


Figure 3: Epipolar Line Distance.

Reprojection Error

Sum of the **Squared Reprojection Errors**

$$err = \sum_{i=1}^{N} ||p_1^i - \pi_1(P^i)||^2 + ||p_2^i - \pi_2(P^i, R, T)||^2 \tag{1.28}$$

Computation is expensive because of point triangulation, but is the **most accurate**!

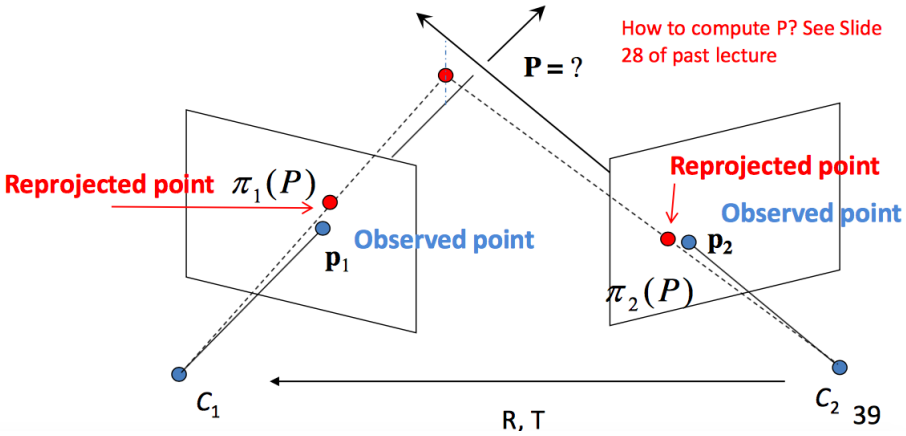


Figure 4: Reprojection Error.

2 Robust Structure From Motion

Matched points are usually contaminated by **outliers**. Causes for this are

- Change in view point and illumination.
- Image noise.
- Occlusions.
- Blur.

The task of removing them is called **Robust Estimation**. Since error is integrating over time, this represents a major issue.

2.1 RANSAC (Random Sample Consensus)

Ransac is the standard method for **model fitting in the presence of outliers** (noise points or wrong data). It can be applied to all problems where the goal is to estimate parameters of a model from the data. An easy example is RANSAC for **line fitting:**

1. Select sample of 2 points at random.
2. Calculate model parameters that fit the data in the sample.
3. Calculate error function for each data point.
4. Select data that supports current hypothesis.
5. Repeat.
6. Select the set with the maximum number of inliers obtained within k iterations.

How many iterations are needed? All pairwise combinations : $\frac{N\cdot(N-1)}{2}$. This is computationally unfeasible if N is too large. With a probabilistic approach, one can reduce this number drastically:
Let w be the number of inliers/N, N be the total number of data points. We can think of w as

$$w = P(\text{selecting an inlier-point out of the dataset}). \tag{2.1}$$

We assume that the 2 points necessary to estimate a line are selected independently, i.e.

$$\begin{aligned} w^2 &= P(\text{both selected points are inliers}) \\ 1-w^2 &= P(\text{at least one of these two points is outlier}) \end{aligned} \tag{2.2}$$

Let k indicate the number of RANSAC iterations so far, then

$$(1-w^2)^k = P(\text{RANSAC never selected two points both inliers}) \tag{2.3}$$

Let p be the probability of success:

$$\begin{aligned} 1-p &= (1-w^2)^k \\ \Rightarrow k &= \frac{\log(1-p)}{\log(1-w^2)}. \end{aligned} \tag{2.4}$$

Remark. Think of having $p = 0.99$ and $w = 0.5$, then $k = 16$, which is dramatically fewer than all combinations. **The number of points does not influence that!**.

RANSAC applied to general model fitting is:

1. Initial: let A be a set of N points.
2. Repeat.
3. Randomly select a sample of s points from A.
4. Fit a model from the s points.
5. Compute the distances of all other points from this model.
6. Construct the inlier set (i.e. count the number of points whose distance is $< d$).
7. Store these inliers.
8. Until maximum number of iterations k is reached.
9. The set with the maximum number of inliers is chosen as solution to the problem.

$$k = \frac{\log(1-p)}{\log(1-w^s)}. \tag{2.5}$$

In order to implement RANSAC for Structure From Motion (SFM), we need three key ingredients

a) What's the **model** in SFM? $\rightarrow$ the Essential Matrix (for calibrated cameras) or the Fundamental Matrix (for uncalibrated cameras). Alternatively, R and T.

b) What's the **minimum number of points** to estimate the model? $\rightarrow$ We know that 5 points is the theoretical minimum number of points. However, 8-point algorithm is the state of the art, then 8 is the minimum.

c) How do we compute the **distance** of a point from the model. $\rightarrow$ We can use the epipolar constraint to measure how well a point correspondence verifies the model E or F, respectively. However, the **Directional error**, the **Epipolar line distance**, or the **Reprojection error (even better)** are used.

1. Randomly select 8 point correspondences.
2. Fit the model to all other points and count the inliers.
3. Repeat from 1 for k times.

With s points one gets:

$$k = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{2.6}$$

Remark. No 6 DOF estimation for the 2-point RANSAC. k increases exponentially with the fraction of outliers ε.

- As observed, k is exponential in the number of points s necessary to estimate the model. We can see that k increases exponentially with the fraction of outliers ε.

- The 8-point algorithm is extremely simple and was very successful; however it requires more than 1177 iterations.
- The 5-point algorithm only requires 145 iterations, but can return up to 10 solutions of E.

Can we use less than 5 points? Yes, with planar motion.

2.2 Planar Motion

Planar motion is described by three parameters ϑ, φ, ρ, defined in Figure 5. It holds

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad T = \begin{pmatrix} \rho\cos(\varphi) \\ \rho\sin(\varphi) \\ 0 \end{pmatrix} \tag{2.7}$$

Let's compute the Epipolar Geometry

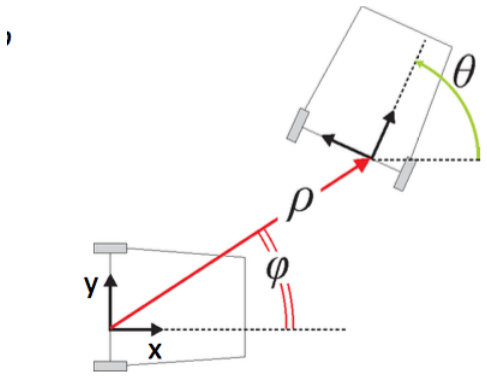


Figure 5: Planar motion.

$$\begin{aligned} E &= [T]_x \cdot R \\ &= \begin{pmatrix} 0 & 0 & \rho\sin(\varphi) \\ 0 & 0 & -\rho\cos(\varphi) \\ -\rho\sin(\varphi) & \rho\cos(\varphi) & 0 \end{pmatrix} \cdot \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & \rho\sin(\varphi) \\ 0 & 0 & -\rho\cos(\varphi) \\ -\rho\sin(\varphi-\theta) & \rho\cos(\varphi-\theta) & 0 \end{pmatrix}. \end{aligned} \tag{2.8}$$

E has 2 DoF (θ, φ), because ρ is the scale factor. Thus, 2 correspondences are sufficient to estimate them.
But: can we use **less** than 2 point correspondences? Yes, if we exploid wheeled vehicles with **non-holonomic** constraints. Wheeled vehicles like cars, follow a locally-planar circular motion about the instantaneous Center of Rotation (ICR). Since $\varphi = \theta/2$, we have only 1 DoF. Only 1 point correspondence is needed. **This is the smallest parametrization possible and results in the most efficient algorithm for removing outliers**

(Scaramuzza). This updates the problem to be

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad T = \begin{pmatrix} \rho\cos(\frac{\theta}{2}) \\ \rho\sin(\frac{\theta}{2}) \\ 0 \end{pmatrix} \tag{2.9}$$

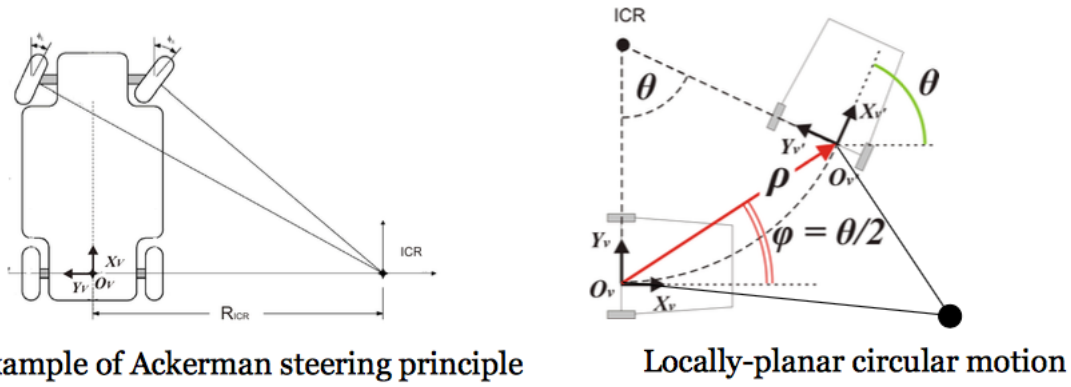
and

$$\begin{aligned} E &= [T]_x \cdot R \\ &= \begin{pmatrix} 0 & 0 & \rho\sin(\frac{\theta}{2}) \\ 0 & 0 & -\rho\cos(\frac{\theta}{2}) \\ \rho\sin(\frac{\theta}{2}) & -\rho\cos(\frac{\theta}{2}) & 0 \end{pmatrix}. \end{aligned} \tag{2.10}$$

With the Epipolar Geometry constraint this leads to

$$\theta = -2\tan^{-1}\left(\frac{v_2 - v_1}{u_2 + u_1}\right). \tag{2.11}$$

Only one iteration: compute θ for every point correspondence. Up to 1000 Hz, 1-point RANSAC is then only used to find the inliers. Motion is then estimated from them in 6DOF.



Example of Ackerman steering principle

Locally-planar circular motion

Figure 6: Non-holonomic.

2.3 Understanding Check

Are you able to answer the following questions?

- *What's the minimum number of correspondences required for calibrated SFM and why?*
- *Are you able to derive the epipolar constraint?*
- *Are you able to define the essential matrix?*
- *Are you able to derive the 8-point algorithm?*
- *How many rotation-translation combinations can be the essential matrix decomposed in?*
- *Are you able to provide a geometrical interpretation of the epipolar constraint?*
- *Are you able to describe the relation between the essential and the fundamental matrix?*
- *Why is it important to normalize the point coordinates in the 8-point algorithm? Describe one or more possible ways to achieve this normalization.*
- *Are you able to describe the normalized 8-point algorithm?*
- *Are you able to provide quality metrics for the fundamental matrix estimation?*
- *Why do we need RANSAC?*
- *What is the theoretical maximum number of combinations to explore?*
- *After how many iterations can RANSAC be stopped to guarantee a given success probability?*
- *What is the trend of RANSAC iterations k vs. the fraction of outliers $\epsilon = 1 - w$ vs. the number of points to estimate the model?*
- *How do we apply RANSAC to the 8-point algorithm vs DLT?*
- *How can we reduce the number of RANSAC iterations for the SFM problem?*
- *In practice, can you fully rely on the formula that predicts the optimal number of iterations?*