Term Project – Calibrated Epipolar Geometry

3D Computer Vision – Lab Session Task

(CTU FEE subjects B4M33TDV, BE4M33TDV, XP33VID)

Martin Matoušek and Jaroslav Moravec

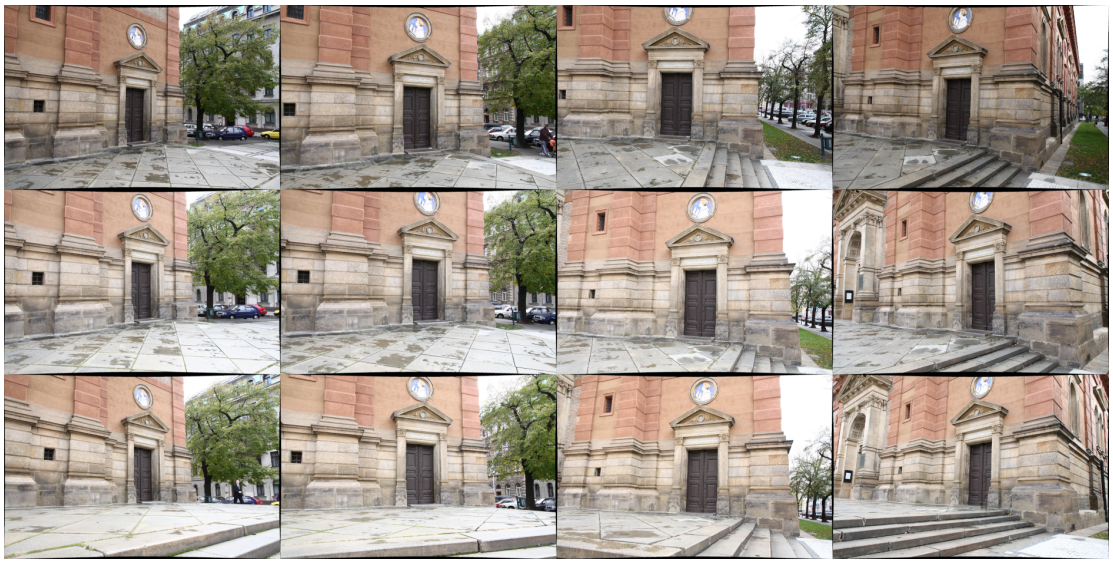
November 2024

CENTER FOR MACHINE
PERCEPTION

Task Outline

Task: **Estimate the calibrated epipolar geometry between two selected views of the scene.**

1. Select one pair of images from the set below and download them from here.
2. Download keypoints and *tentative* correspondences (matches) between the selected pair of views from here.
3. Use robust estimation (RANSAC/MLESAC) to find the essential matrix between the images.
4. Show outliers and inliers of the epipolar geometry (as a needle map).
5. Select a reasonable sub-set of inliers (e.g., every n-th) and show corresponding points and epipolar lines in both images (in the same color).



Sparse Correspondences

The sparse correspondences for provided images has been precomputed and they are available. Note, that the correspondences are tentative, so they may contain mismatches.
The correspondences are stored in several files:

- detected image keypoints (here `u_<id>.txt`)
- 0-based indices of corresponding keypoints (here `m_<i1>_<i2>.txt`)

Example: Working with correspondences

Keypoints in Image 1

```
  5.3 1613.4
  7.0  364.8
  9.5 1522.3
  9.9  585.1
 10.9  571.7  <---
 11.2  578.6
 11.3  666.1
   ...
```

Keypoints in Image 2

```
  6.3 1749.0
  8.4 1753.3
  8.9  497.9
 10.4  540.9
 11.0  683.2
 11.0  687.8
 11.1  589.8
 11.3  583.4  <---
 12.1 1212.6
 12.2  949.3
   ...
```

Correspondences

```
  4 7 <---
  5 18285
 11 27631
  ...
```

Calibrated Epipolar Geometry

- Let us assume that all images were taken by the same perspective camera with known calibration matrix $\mathbf{K}$. Two specific views of the scene are then related by a so-called essential matrix $\mathbf{E}$, defined as:

$$\mathbf{E} = [-\mathbf{t}]_\times \mathbf{R},$$

  where $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and the translation vector from the second view to the first view, respectively.
- Given two corresponding homogeneous image coordinates $\underline{\mathbf{x}} \in I_1$ and $\underline{\mathbf{y}} \in I_2$, we get their normalized coordinates $\underline{\mathbf{x}}'$ and $\underline{\mathbf{y}}'$ as follows:

$$\underline{\mathbf{x}}' = \mathbf{K}^{-1}\underline{\mathbf{x}} \quad \text{and} \quad \underline{\mathbf{y}}' = \mathbf{K}^{-1}\underline{\mathbf{y}}.$$

  Then it must hold that:

$$\underline{\mathbf{y}}'^{\top}\mathbf{E}\,\underline{\mathbf{x}}' = 0 \quad \text{and} \quad \underline{\mathbf{x}}'^{\top}\mathbf{E}^{\top}\,\underline{\mathbf{y}}' = 0.$$

- As all the acquired correspondences are tentative and can contain outliers and noise, we will use robust estimation (RANSAC) that we know from previous assignments. I.e., iteratively repeating hypothesis generation and consensus verification, to find model with more inliers.

Hypothesis Generation

- All images were taken with the same calibrated camera with the calibration matrix

$$\mathbf{K} = \begin{pmatrix} 2080 & 0 & 1421 \\ 0 & 2080 & 957 \\ 0 & 0 & 1 \end{pmatrix}.$$

$\Longrightarrow$ We first need to transform all keypoints from both images by $\mathbf{K}^{-1}$ to get the normalized coordinates, in order to search for the essential matrix.

- The hypothesis generation then has three steps:
  1. Estimating potential essential matrices from five random normalized correspondences.
  2. Decomposing each of the essential matrices into possible rotations and translations (four combinations).
  3. Selecting the pair of rotation $\mathbf{R}$ and translation $\mathbf{t}$, so that all five reconstructed 3D points are in front of both cameras $\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$ and $\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$ (at most one solution $(\mathbf{R}, \mathbf{t})$ for each essential matrix).
     $\Longrightarrow$ From one set of five corresponding points, we get multiple hypothesis $(\mathbf{E}, \mathbf{R}, \mathbf{t})$, each needs to be verified separately

Hypothesis Generation – Essential Matrix Estimation

- The matrix $\mathbf{E}$ (up to scale) has five degrees of freedom and can be estimated from five normalized correspondences using a so-called 5-point algorithm. Download and compile our implementation of the algorithm.

Hint: Compiling the external C++ library p5

- Extract the downloaded .zip archive
- In the directory `p5/src-python` execute:
  ```
  python3 setup.py build
  ```
- Locate the library file (`.so` or `.dll`) in the `build` directory and copy it into `p5/python/p5`
- Test the functionality by running:
  ```
  python3 demo_p5.py
  ```
  in the directory `p5/python`
- To use the function `p5.p5gb(...)` in your code, copy the directory `p5/python/p5` next to the script and import it as usual

- Randomly sample five normalized correspondences $\mathbf{x}'^{(i)} \sim \mathbf{y}'^{(i)}$ and use the 5-point algorithm to estimate the essential matrix. Multiple solutions may be returned.

Hint: Python

```
import p5
...
# Sample five 2D correspondences xs~ys
Es=p5.p5gb(K_inv @ e2p(xs), K_inv @ e2p(ys))
```

Hypothesis Generation – Essential Matrix Decomposition

- Each of the returned estential matrices yields one of four possible combinations of rotation and translation. Here, we will refresh the steps of the decomposition method from the lecture, please see the slides for more details.
- Given an essential matrix $\mathbf{E}$:
  1. Compute SVD of $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ and verify that $\mathbf{D} = \lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \lambda \neq 0.$
  2. Ensure that $\mathbf{U}$ and $\mathbf{V}$ are rotation matrices, set:
  $$\mathbf{U} = \det(\mathbf{U}) \cdot \mathbf{U} \quad \text{and} \quad \mathbf{V} = \det(\mathbf{V}) \cdot \mathbf{V}$$
  3. Compute the possible rotation matrix and translation vector as:
  $$\mathbf{R}(\alpha) = \mathbf{U} \begin{pmatrix} 0 & \alpha & 0 \\ -\alpha & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{V}^\top, \quad \mathbf{t}(\beta) = -\beta \mathbf{U}_3,$$
  where $|\alpha| = 1$, $\beta \neq 0$ and $\mathbf{U}_3$ is the third column of $\mathbf{U}$.
- This decomposition yields four possible combinations of $\mathbf{R}$ and $\mathbf{t}$ for combinations of $\alpha = \pm 1$ and $\beta = \pm 1$.

- Let us assume that the first camera is canonical, i.e., $\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$, and the second one is defined as $\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R}(\alpha) & \mathbf{t}(\beta)\end{bmatrix}$. Given the (same) five sampled correspondences $\mathbf{x}^{(i)} \sim \mathbf{y}^{(i)}$ $(i = 1, ..., 5)$, we perform triangulation with numerical conditioning:
  1. For the correspondence $\mathbf{x}^{(i)} \sim \mathbf{y}^{(i)}$ construct the system of linear equations:

$$\mathbf{D} = \begin{bmatrix} x_1^{(i)}(\mathbf{p}_3^1)^\top - (\mathbf{p}_1^1)^\top \\ x_2^{(i)}(\mathbf{p}_3^1)^\top - (\mathbf{p}_2^1)^\top \\ y_1^{(i)}(\mathbf{p}_3^2)^\top - (\mathbf{p}_1^2)^\top \\ y_2^{(i)}(\mathbf{p}_3^2)^\top - (\mathbf{p}_2^2)^\top \end{bmatrix}, \text{ where } (\mathbf{p}_i^j)^\top \text{ is the } i\text{-th row of } \mathbf{P}_j.$$

  2. Re-scale the problem by a regular diagonal conditioning matrix $\mathbf{S} \in \mathbb{R}^{4\times4}$, then:

$$\mathbf{0} = \mathbf{D}\mathbf{X} = \mathbf{D}\mathbf{S}\mathbf{S}^{-1}\underline{\mathbf{X}}.$$

  3. Solve $\mathbf{D}\mathbf{S}\underline{\mathbf{X}}' = 0$ for $\underline{\mathbf{X}}'$.
  4. Compute the 3D point $\underline{\mathbf{X}} = \mathbf{S}\underline{\mathbf{X}}'$.

  (see slides from the lecture for more details)

Hint: Python

```
S=np.diag(1/np.max(np.abs(D), axis=0))
u,_,_=np.linalg.svd((D@S).T@(D@S))
X = (S@u[:, -1]) / (S[-1, :]@u[:, -1])
```

- We then select such a combination of $\mathbf{R}$ and $\mathbf{t}$, so that all points are in front of both cameras. I.e.: $(\mathbf{P}_1\underline{\mathbf{X}}^{(i)})_3 > 0 \ \wedge \ (\mathbf{P}_2\underline{\mathbf{X}}^{(i)})_3 > 0, \forall i = 1, ..., 5.$

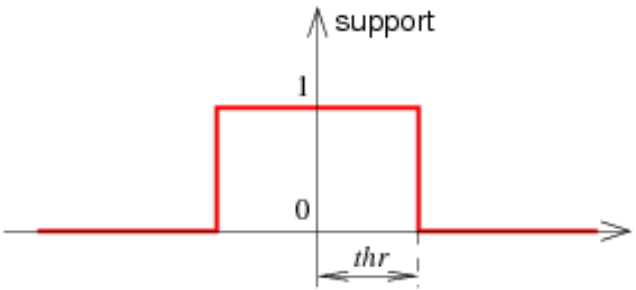
- Compute the fundamental matrix $\mathbf{F}$ from the found essential matrix $\mathbf{E}$ (to estimate errors in pixels):

$$\mathbf{F} = (\mathbf{K}^\top)^{-1}\,\mathbf{E}\,\mathbf{K}^{-1}.$$

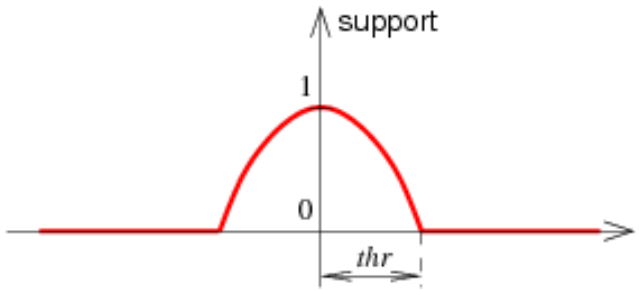
- To verify the quality of each hypothesis, we will use the Sampson error. For a 2D correspondence $\mathbf{x} \sim \mathbf{y}$, its squared Sampson error is defined as:

$$\varepsilon_{\mathbf{F}}(\mathbf{x},\mathbf{y})^2 = \frac{(\mathbf{y}^\top \mathbf{F}\mathbf{x})^2}{||\mathbf{S}\mathbf{F}\underline{\mathbf{x}}||^2 + ||\mathbf{S}\mathbf{F}^\top\underline{\mathbf{y}}||^2}, \text{ where } \quad \mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \qquad (1)$$

- Find an essential matrix $\mathbf{E}^*$ with the best support over all tentative correspondences using either RANSAC (left) or MLESAC (right).



$$s_i = \begin{cases} 1 & \text{if } \varepsilon_{\mathbf{F}}(x^{(i)}, y^{(i)}) \le \theta \\ 0 & \text{otherwise} \end{cases}$$



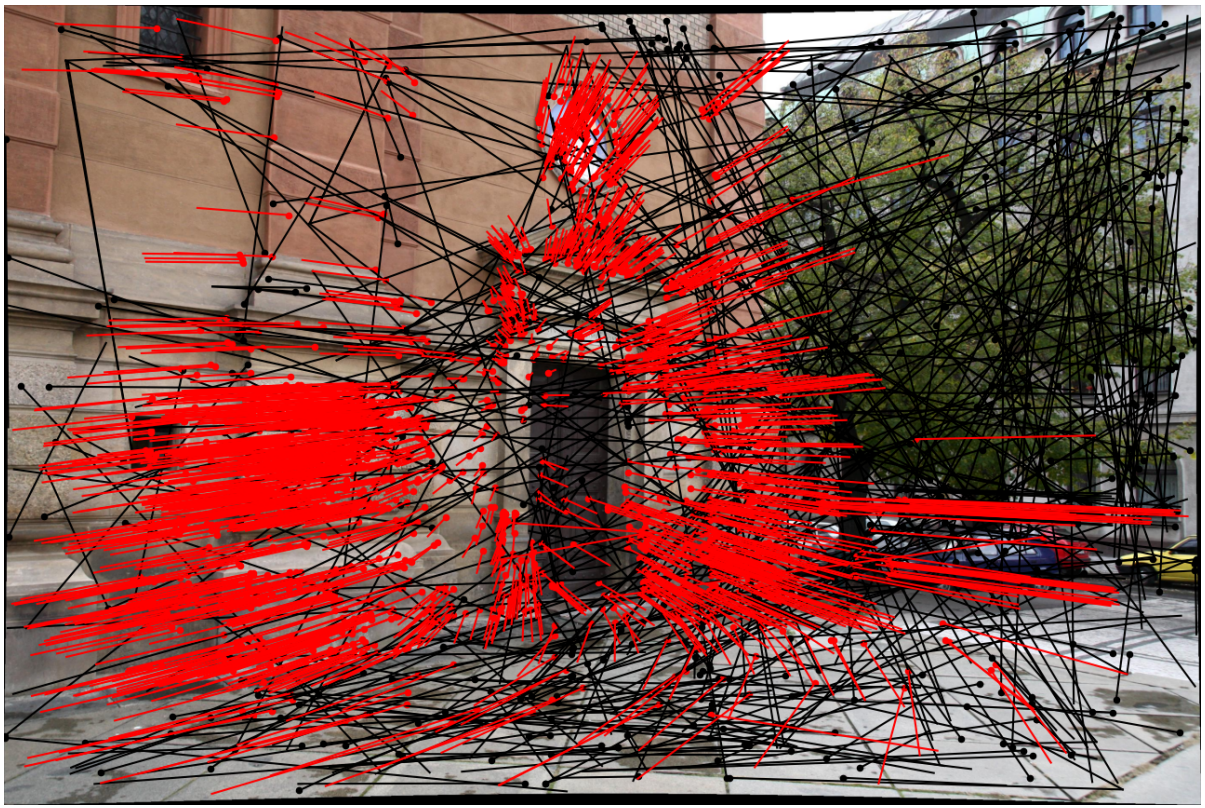
$$s_i = \begin{cases} 1 - \frac{\varepsilon_{\mathbf{F}}(x^{(i)}, y^{(i)})^2}{\theta^2} & \text{if } \varepsilon_{\mathbf{F}}(x^{(i)}, y^{(i)}) \le \theta \\ 0 & \text{otherwise} \end{cases}$$

$$\text{support} = \textstyle\sum_i s_i$$

Expected Results: Inlier Correspondences

Visualize the inliers and outliers (in different colors) of the estimated essential matrix $\mathbf{E}^*$ as a needle map. Given one selected correspondence $\mathbf{x} \sim \mathbf{y}$, its needle visualization on the first image can be done as follows:

```
plt.scatter(x[0], x[1], 20, 'r')
plt.plot([x[0], x[0]+(y[0]-x[0])], [x[1], x[1]+(y[1]-x[1])], 'r-', linewidth=2)
```

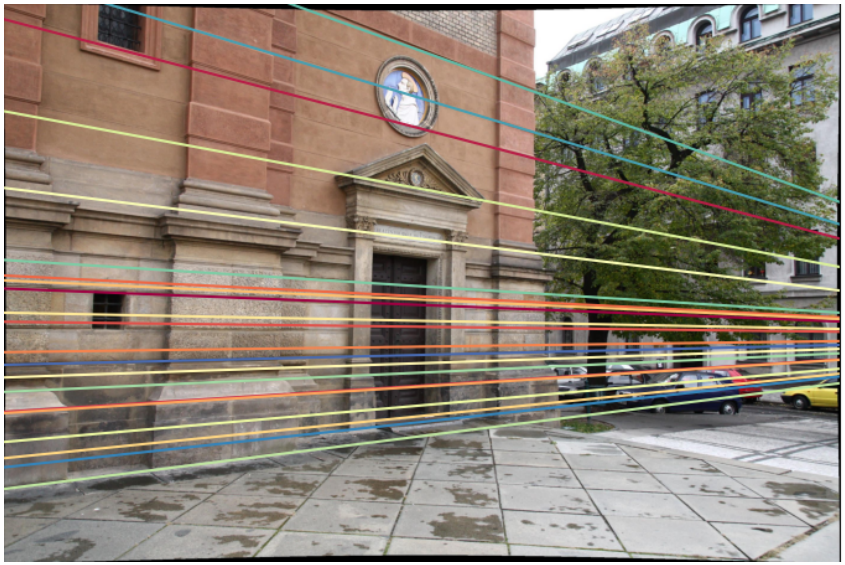


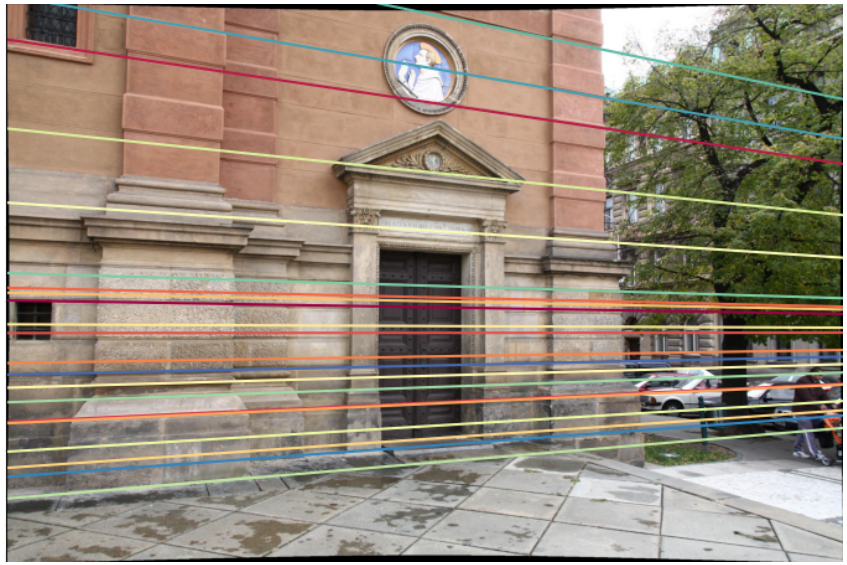
Expected Results: Epipolar Lines

Visualize a subset of corresponding epipolar lines of the estimated fundamental matrix $\mathbf{F}^* = (\mathbf{K}^\top)^{-1}\,\mathbf{E}^*\,\mathbf{K}^{-1}$ (use the same color for corresponding epipolar lines). Given one selected inlier 2D correspondence $\mathbf{x} \sim \mathbf{y}$, we get:

$$\mathbf{e}_1 = \mathbf{F}^{*\top}\mathbf{y} \quad \text{and} \quad \mathbf{e}_2 = \mathbf{F}^*\mathbf{x},$$

where $\mathbf{e}_1$ is the epipolar line in the first image and $\mathbf{e}_2$ is the corresponding epipolar line in the second image.





Toolbox Functions

- For this assignment, you should implement three toolbox functions, which will be needed throughout the term project:
  1. Linear triangulation with numerical conditioning
  2. Essential matrix decomposition with chirality constraint
  3. Sampson error estimation
- You can check your implementations of these toolbox functions through automatic evaluation in BRUTE. In case of a mistake, you will be prompted with the anticipated result for a given input.

Toolbox

`X = Pu2X(P1, P2, u1, u2)`
Reconstructs n 3D points `X` from n corresponding 2D points `u1` and `u2` observed by two cameras `P1` and `P2`, respectively. See slide 8. Inputs: `P1, P2` $\in \mathbb{R}^{3\times 4}$; `u1, u2` $\in \mathbb{R}^{3\times n}$. Output: `X` $\in \mathbb{R}^{4\times n}$.

`[R, t] = EutoRt(E, u1, u2)`
Decomposes the given essential matrix `E` into rotation `R` and translation `t` (see slide 7). It returns the combination, which fulfills the chirality constraint for n points `u1` and `u2` (see slide 8). If the chirality fails, it returns `R = []`.
Inputs: `E` $\in \mathbb{R}^{3\times 3}$; `u1, u2` $\in \mathbb{R}^{3\times n}$. Outputs: `R` $\in \mathbb{R}^{3\times 3}$; `t` $\in \mathbb{R}^{3\times 1}$.

`err = err_F_sampson(F, u1, u2)`
Returns the **squared** (less computationally intensive) Sampson error for n homogeneous corresponding coordinates `u1`$\sim$`u2`. See (1). Inputs: `F` $\in \mathbb{R}^{3\times 3}$; `u1, u2` $\in \mathbb{R}^{3\times n}$. Output: `err` $\in \mathbb{R}^{1\times n}$.