

EX3 Computer Vision

ראשית ביצענו את המשימה של חישוב ה fundamental matrix בשתי דרכים :

1. תוך שימוש בפונקציות מתוך ספריית OPENCV

- לצורך חישוב ה-fundamental matrix יש למצוא כמה שיותר נקודות מתאימות בין שתי התמונות, לצורך כך נעשה שימוש ב SIFT descriptors. שימוש בפונקציה detectAndCompute מצאת keypoints and descriptors בכל אחת מהתמונות.
- FLANN based Matcher -נעשה שימוש ב https://docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html
- ונבצע חישוב בעזרת findFundamentalMat

findFundamentalMat() [1/3]

```
Mat cv::findFundamentalMat ( InputArray points1,
                             InputArray points2,
                             int method,
                             double ransacReprojThreshold,
                             double confidence,
                             int maxIters,
                             OutputArray mask = noArray()
                             )
```

Python:

```
retval, mask = cv.findFundamentalMat( points1, points2, method, ransacReprojThreshold, confidence, maxIters[, mask] )
retval, mask = cv.findFundamentalMat( points1, points2[, method[, ransacReprojThreshold[, confidence[, mask]]]] )
```

#include <opencv2/calib3d.hpp>

Calculates a fundamental matrix from the corresponding points in two images.

Parameters

points1	Array of N points from the first image. The point coordinates should be floating-point (single or double precision).
points2	Array of the second image points of the same size and format as points1 .
method	Method for computing a fundamental matrix. <ul style="list-style-type: none">• CV_FM_7POINT for a 7-point algorithm. $N = 7$• CV_FM_8POINT for an 8-point algorithm. $N \geq 8$• CV_FM_RANSAC for the RANSAC algorithm. $N \geq 8$• CV_FM_LMEDS for the LMedS algorithm. $N \geq 8$

מצורפים הסברים על כך שכאשר מריצים את האלגוריתם CV_FM_8POINT אין צורך לנרמל את הנקודות באופן ידני.

the OpenCV implementation [automatically normalizes the input points](#), so there is no need to normalize them manually.

<https://stackoverflow.com/questions/30963199/is-opencv-findfundamentalmat-apply-normalization-before-calculation>

- הערה - ניתן להעביר את F שחישבנו לפונקציה computeCorrespondEpilines שמוצאת קווים אפיפולרים המתאימים לנקודות הנתונות.

הקוד מצוי בקובץ : Draw the epipolar lines.ipynb

2. עשינו שימוש בקוד לחילוץ נקודות מתאימות בשתי תמונות כפי שלמדנו בעיבוד תמונה עם פונקציות :

- **`harris_corner_detector`** - returns an array with shape where `ret[i,:]` are the `[x,y]` coordinates of the `ith` corner points.
- **`sample_descriptor`** that samples descriptors at the given corners.
- **`find_features`** that detects and extracts feature points from a pyramid.
- **`match_features`** that returns indices of matching descriptors.

לאחר מכן חשבנו את ה `fundamental matrix` על ידי :

- נירמול הקורדינטות של התמונות
- בניית מטריצת משוואות על פי הנוסחא
- חישוב SVD, חילוץ העמודה האחרונה של `V` והגדרתה בתור `F`.
- נדאג ש `F` תהיה עם דרגה 2 על ידי איפוס הערכים הסינגולרים האחרונים
- ולסיום `reverse normalization`

הקוד מצוי בקובץ : `Calculate Fundamental Matrix.ipynb`

The two input images:





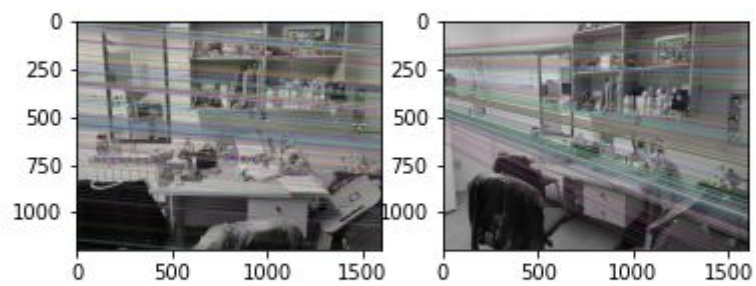
Compute the fundamental matrix F

```

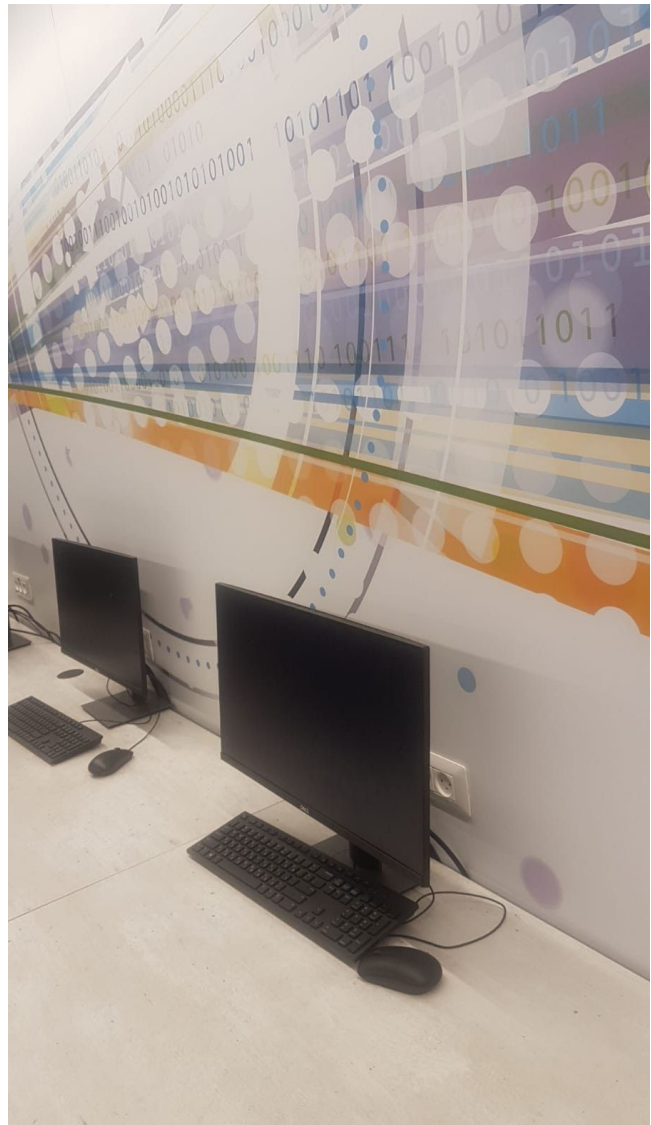
[[-3.59405058e-07  4.49702717e-06 -3.43025285e-04]
 [ 1.32867545e-07 -2.51184006e-06 -5.91519273e-03]
 [-3.15341549e-04  4.12856787e-03  1.00000000e+00]]

```

The images with the epipolar lines drawn in it



The two input images:



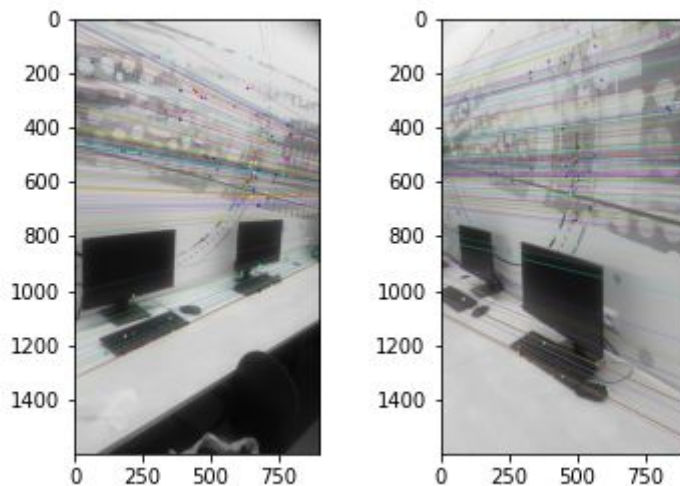
Compute the fundamental matrix F

Compute the fundamental matrix F from a pair of **normalized** images with at least 8 corresponding points

CV_FM_8POINT

```
[[ 2.73432880e-06 -2.38474161e-07  1.86874165e-04]
 [ 2.09717539e-06  1.09534106e-06 -1.12629774e-03]
 [-3.05515266e-03 -9.50552220e-04  1.00000000e+00]]
```

The images with the epipolar lines drawn in it



תיאור האלגוריתם כפי שמומש:

בהינתן שתי נקודות תואמות :

1. נרמול הקורדינטות של התמונה

- חישוב המרכזים והמרחקים הממוצעים עבור כל שתי קבוצות נקודות
- חישוב הטרנספורמציות המנורמלות לכל קבוצת נקודות - לאחר הטרנספורמציה בכל קבוצה המרכז יהיה ממוקם בראשית והמרחק הממוצע מהראשית הוא

$\sim \sqrt{2}$

2. חישוב ה Fundamental Matrix עם קואורדינטות מנורמלות:

- נשתמש במשוואה $Ax = 0$ כך שעבור זוג נקודות m_1, m_2 שורה ב A מייצגת את המקדמים של המשוואה

$$(m_2, 1) \cdot F \cdot (m_1, 1) = 0$$

על מנת לייעל את זמן הריצה נחשב את $A.TA$ ונפתור את המשוואה

$$(A.T \cdot A)x = 0$$

- נקח את העמודה האחרונה ב V כפתרון של $AF = 0$ (נבצע פירוק SVD). כמו כן נוודא ש F סינגולרית כלומר מדרגה 2 על ידי איפוס האלמנט האחרון באלכסון של מטריצה אלכסונית S , ונרכיב את המטריצה F בחזרה.
- נפעיל את הטרנספורמציה ההפוכה למה שהשתמשנו לנרמול קואורדינטות של הנקודות.
- נהפוך את $F[3,3]$ להיות 1
- נחזיר את F

ההסבר המפורט מהספר ע"מ 282:

Objective

Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix F such that $\mathbf{x}'_i{}^T F \mathbf{x}_i = 0$.

Algorithm

- (i) **Normalization:** Transform the image coordinates according to $\hat{\mathbf{x}}_i = T\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = T'\mathbf{x}'_i$, where T and T' are normalizing transformations consisting of a translation and scaling.
- (ii) Find the fundamental matrix \hat{F} corresponding to the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ by
 - (a) **Linear solution:** Determine \hat{F} from the singular vector corresponding to the smallest singular value of \hat{A} , where \hat{A} is composed from the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ as defined in (11.3).
 - (b) **Constraint enforcement:** Replace \hat{F} by \hat{F}' such that $\det \hat{F}' = 0$ using the SVD (see section 11.1.1).
- (iii) **Denormalization:** Set $F = T'^T \hat{F}' T$. Matrix F is the fundamental matrix corresponding to the original data $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

Algorithm 11.1. *The normalized 8-point algorithm for F .*