CSCI 5980/8980: Assignment #4

Fundamental Matrix

1 Submission

- Assignment due: March 23
- Individual assignment.
- Write-up submission format: a single PDF up to 5 pages (more than 5 page assignment will be automatically returned.).
- MATLAB code submission in conjunction with visualization.
- Submission through Moodle.

CSCI 5980/8980: Assignment #4

Fundamental Matrix

2 VLFeat Installation

One of key skills to learn in computer vision is the ability to use other open source code, which allow you not to re-invent the wheel. We will use `VLFeat` by A. Vedaldi and B. Fulkerson (2008) for SIFT extraction given your images. Install `VLFeat` from following:
`http://www.vlfeat.org/install-matlab.html`
Run `vl_demo_sift_basic` to double check the installation is completed.

(NOTE) You will use this library only for SIFT feature extraction and its visualization. All following visualizations and algorithms must be done by your code.

3 SIFT Feature Extraction

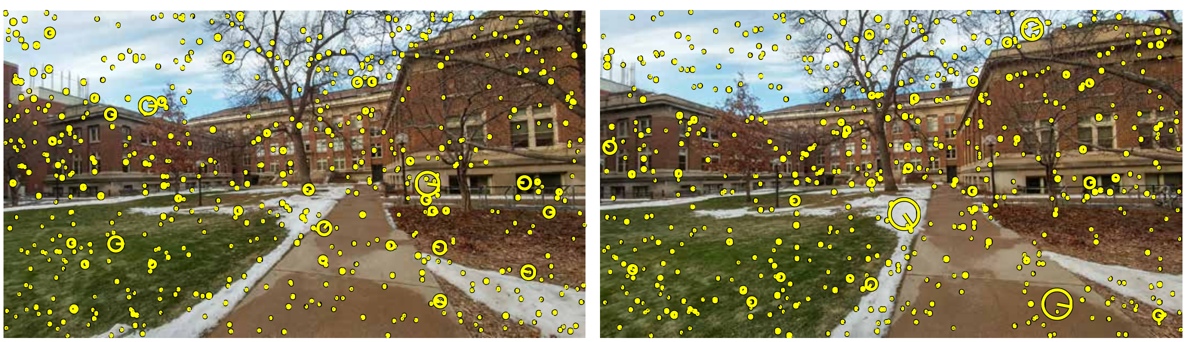


Figure 1: Given your two cellphone images (left and right), you will extract SIFT descriptors and visualize them using `VLFeat`.

You will extract David Lowe's SIFT (Scale Invariant Feature Transform) features from your cellphone images as shown in Figure 1. First, take a pair of pictures with your calibrated camera (intrinsic parameter, $\mathbf{K}$, and radial distortion parameter, k, are pre-calibrated.) as follow:

1. Take the first picture and another one after moving one step right (1m).

2. Common 3D objects across different depths, e,g., buildings, ground plane, and tree, appear in both images.

3. Two images have to be similar enough so that the appearance based image matching can be applied, i.e., SIFT feature does not break, while maintaining the baseline between cameras (at least 1 m), i.e., similar camera orientation and sufficient translation.

4. Avoid a 3D scene dominated by one planar surface, e.g., looking at ground plane.

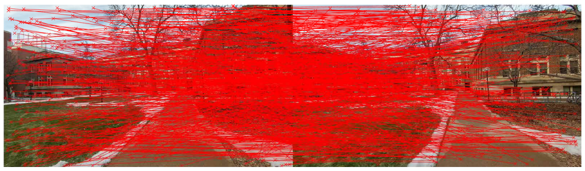
Write-up:

(SIFT visualization) Use `VLFeat` to visualize SIFT features with scale and orientation as shown in Figure 1. You may want to plot up to 500 feature points. You may want to follow the following tutorial:
`http://www.vlfeat.org/overview/sift.html`

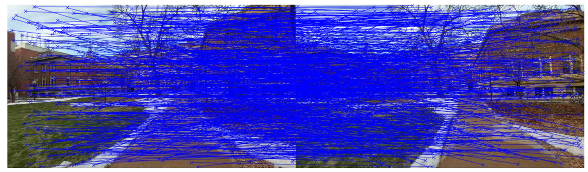
CSCI 5980/8980: Assignment #4

Fundamental Matrix

4 SIFT Feature Matching

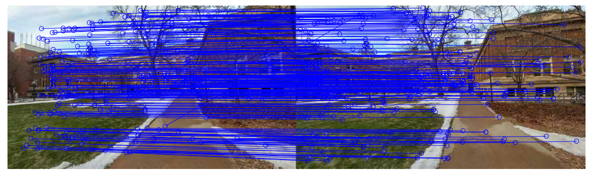


(a) Matching from $\mathcal{I}_1$ to $\mathcal{I}_2$

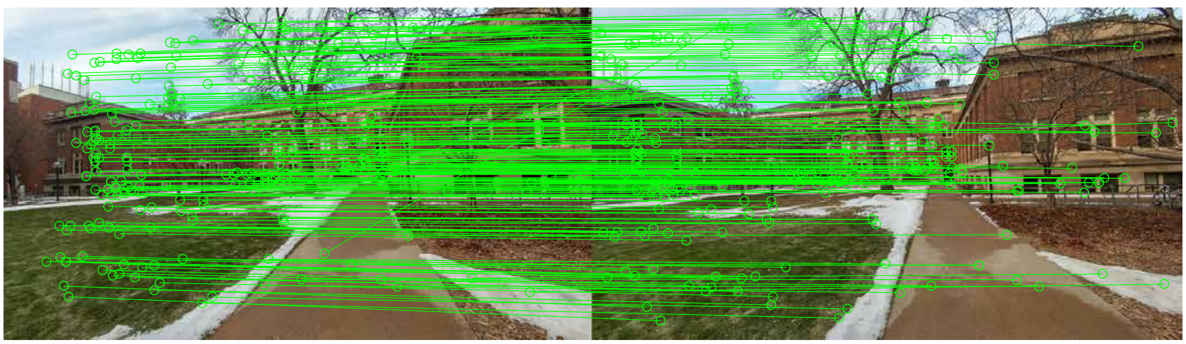


(b) Matching from $\mathcal{I}_2$ to $\mathcal{I}_1$

(c) Matching from $\mathcal{I}_1$ to $\mathcal{I}_2$ after ratio test



(d) Matching from $\mathcal{I}_2$ to $\mathcal{I}_1$ after ratio test



(e) Bidirectional matching between $\mathcal{I}_1$ and $\mathcal{I}_2$

Figure 2: You will match points between $\mathcal{I}_1$ and $\mathcal{I}_2$ using SIFT features.

(NOTE) From this point, you cannot use any function provided by `VLFeat`.

The SIFT is composed of scale, orientation, and 128 dimensional local feature descriptor (integer), $\mathbf{f} \in \mathbb{Z}^{128}$. You will use the SIFT features to match between two images, $\mathcal{I}_1$ and $\mathcal{I}_2$.

Write-up:

(1) (Nearest neighbor search) Let two sets of features be $\{\mathbf{f}_1, \cdots, \mathbf{f}_{N_1}\}$ from $\mathcal{I}_1$ and $\{\mathbf{g}_1, \cdots, \mathbf{g}_{N_2}\}$ from $\mathcal{I}_2$ where N_1 and N_2 are the number of features in image 1 and 2, respectively. Compute nearest neighbor per feature and visualize ($\{\mathbf{f}_1, \cdots, \mathbf{f}_{N_1}\} \rightarrow \{\mathbf{g}_1, \cdots, \mathbf{g}_{N_2}\}$ and $\{\mathbf{g}_1, \cdots, \mathbf{g}_{N_2}\} \rightarrow \{\mathbf{f}_1, \cdots, \mathbf{f}_{N_1}\}$) as shown in Figure 2(a) and Figure 2(b). Note that the distance between two features is defined as $d = \|\mathbf{f} - \mathbf{g}\|$. You may use `knnsearch` function in MATLAB.

(2) (Ratio test) Filter out matches using the ratio test, i.e., keep the match if $d_{ij_1}/d_{ij_2} < 0.7$ and discard otherwise, where d_{ij_1} and d_{ij_2} are the first and second nearest neighbors for the i^{th} feature, respectively. Visualize the matches after the ratio test as shown Figure 2(d) and Figure 2(c).

(3) (Bidirectional match) Visualize bidirectionally consistent matches as shown Figure 2(e). Compare the number of matches from (1) to (3).

5 Fundamental matrix

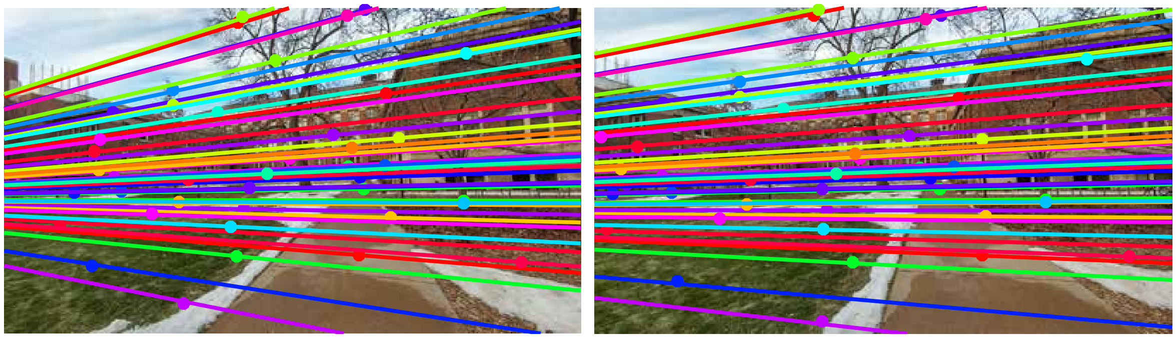


Figure 3: You will visualize epipole and epipolar lines for each image.

Compute a fundamental matrix between $\mathcal{I}_1$ and $\mathcal{I}_2$.

Write-up:

(1) (Fundamental matrix) Complete the following function to compute a fundamental matrix, linearly:
$\mathbf{F}$ = `ComputeFundamentalMatrix(`$\mathbf{u}$`, `$\mathbf{v}$`)`
Input: $\mathbf{u}$ and $\mathbf{v}$ are $N_f \times 2$ matrices of 2D correspondences where the N_f is the number of 2D correspondences, $\mathbf{u} \leftrightarrow \mathbf{v}$.
Output: $\mathbf{F} \in \mathbb{R}^{3\times3}$ is a rank 2 fundamental matrix.

(2) (Epipole and epipolar line) Pick 8 random correspondences, $\mathbf{u}_r \leftrightarrow \mathbf{v}_r$, compute the fundamental matrix, $\mathbf{F}_r$ and visualize epipole and epipolar lines for the rest of feature points in both images as shown in Figure 3. Pick different sets of correspondences and visualize different epipolar lines.

CSCI 5980/8980: Assignment #4

Fundamental Matrix

6 Robust Fundamental Matrix Estimation

Estimate the fundamental matrix using RANSAC.

Write-up:

(1) (RANSAC with fundamental matrix) Write a RANSAC algorithm for the fundamental matrix estimation given N matches from Section 4 using the following pseudo code:

Algorithm 1 GetInliersRANSAC

$n \leftarrow 0$
for $i = 1 : M$ **do**
    Choose 8 correspondences, $\mathbf{u}_r$ and $\mathbf{v}_r$, randomly from $\mathbf{u}$ and $\mathbf{v}$.
    $\mathbf{F}_r$ `= ComputeFundamentalMatrix(`$\mathbf{u}_r$`,` $\mathbf{v}_r$`)`
    Compute the number of inliers, n_r, with respect to `F`.
    if $n_r > n$ **then**
        $n \leftarrow n_r$
        $\mathbf{F} = \mathbf{F}_r$
    end if
end for

(2) (Epipole and epipolar line) Using the fundamental matrix, visualize epipole and epipolar lines.

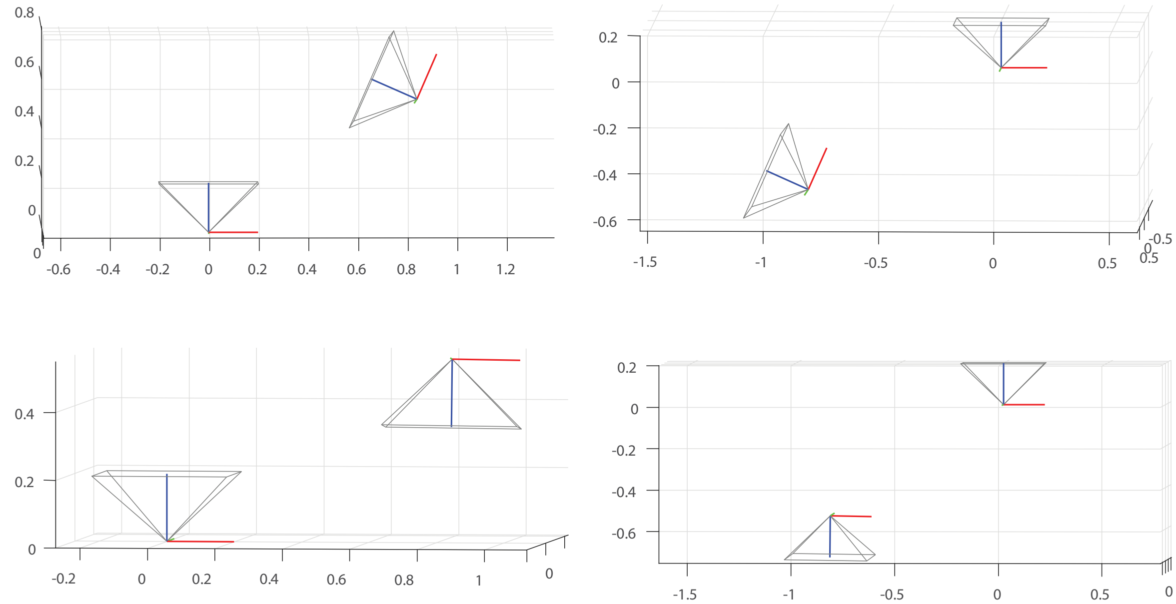


Figure 4: Four configurations of camera pose from a fundamental matrix.

(3) (Camera pose estimation) Compute 4 configurations of relative camera poses:
`[R1 C1 R2 C2 R3 C3 R4 C4] = CameraPose(`**F**`, `**K**`)`
Input: **F** is the fundamental matrix and **K** is the intrinsic parameter.
Output: `R1 C1` ⋯ `R4 C4` are rotation and camera center (represented in the world coordinate system).

(4) Visualize the 4 configuration in 3D as shown in Figure 4.

7 Triangulation

Given four configurations of relative camera pose, you will find the best camera pose by verifying through 3D point triangulation.

Write-up:

(1) (Linear triangulation) Write a code that computes the 3D point given the correspondence, $\mathbf{u} \leftrightarrow \mathbf{v}$, and two camera projection matrices:

`[X] = LinearTriangulation(P1,u,P2,v)`

Input: $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{3\times 4}$ are two camera projection matrices, and $\mathbf{u} \leftrightarrow \mathbf{v} \in \mathbb{R}^2$ are their 2D correspondence.
Output: $\mathbf{X} \in \mathbb{R}^3$ is the triangulated 3D point.
Hint: Use the triangulation method by linear solve, i.e.,

$$\begin{bmatrix} \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \\ \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

(2) (Cheirality) Write a code that computes the number of 3D points in front of two cameras. The condition of a 3D point being in front of camera is called *cheirality*:

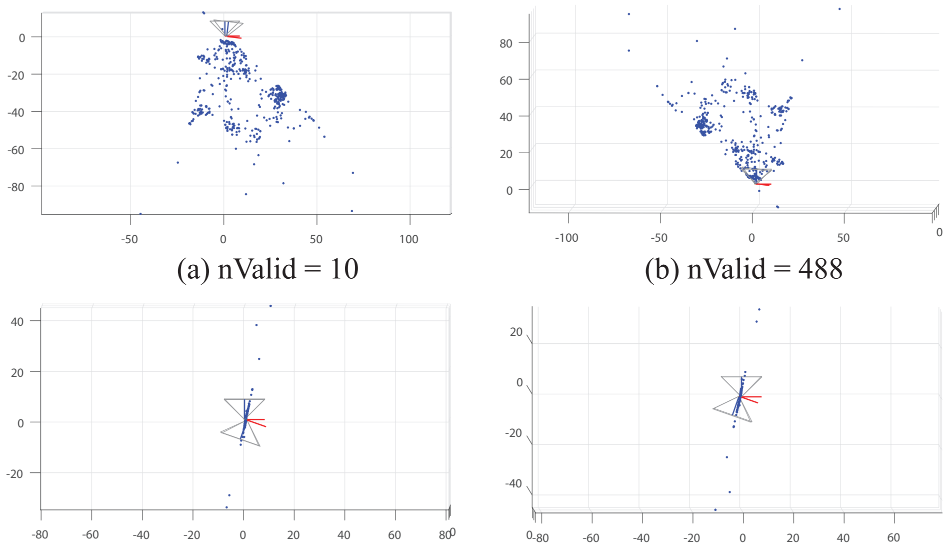
`idx = CheckCheirality(Y,C1,R1,C2,R2)`

Input: $\mathbf{Y}$ is a $n \times 3$ matrix that includes n 3D points, and $\mathbf{C}_1$,$\mathbf{R}_1$/$\mathbf{C}_2$,$\mathbf{R}_2$ are the first and second camera poses (camera center and rotation).
Output: `idx` is a set of indices of 3D points that satisfy cheirality.
Hint: the point must satisfy $\mathbf{r}_3(\mathbf{X} - \mathbf{C}) > 0$ where $\mathbf{r}_3$ is the 3rd row of the rotation matrix (z-axis of the camera), $\mathbf{C}$ is the camera position and $\mathbf{X}$ is the 3D point.

CSCI 5980/8980: Assignment #4

Fundamental Matrix

(3) (Camera pose disambiguation) Based on cheirality, find the correct camera pose. Visualize 3D camera pose and 3D points together as shown in Figure 5.
Hint: Use `plot3` and `DisplayCamera.m` to visualize them.



(a) nValid = 10

(b) nValid = 488

(c) nValid = 0

(d) nValid = 0

Figure 5: You will visualize four camera pose configurations with point cloud.

(4) (Reprojection) Project 3D points to each camera and visualize reprojection, $\widehat{\mathbf{u}}$ and measurement, $\mathbf{u}$ onto the image as shown in Figure 6, i.e.,

$$\lambda \begin{bmatrix} \widehat{\mathbf{u}} \\ 1 \end{bmatrix} = \mathbf{KR} \begin{bmatrix} \mathbf{I} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

where $\mathbf{X}$ is the reconstructed point.

Figure 6: Visualization of measurements and reprojection.