1 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.

2 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, 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 \texttt{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:

\url{http://www.vlfeat.org/overview/sift.html}
3 SIFT Feature Matching

Figure 2: You will match points between $I_1$ and $I_2$ 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), $f \in \mathbb{Z}^{128}$. You will use the SIFT features to match between two images, $I_1$ and $I_2$.

Write-up:

(1) (Nearest neighbor search) Let two sets of features be $\{f_1, \ldots, f_{N_1}\}$ from $I_1$ and $\{g_1, \ldots, g_{N_2}\}$ from $I_2$ where $N_1$ and $N_2$ are the number of features in image 1 and 2, respectively. Compute nearest neighbor per feature and visualize ($\{f_1, \ldots, f_{N_1}\} \rightarrow \{g_1, \ldots, g_{N_2}\}$ and $\{g_1, \ldots, g_{N_2}\} \rightarrow \{f_1, \ldots, f_{N_1}\}$) as shown in Figure 2(a) and Figure 2(b). Note that the distance between two features is defined as $d = \|f - g\|$. 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.
CSCI 5980/8980: Assignment #4

Fundamental Matrix

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).
4 Fundamental matrix

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

Compute a fundamental matrix between $I_1$ and $I_2$.

Write-up:

(1) (Fundamental matrix) Complete the following function to compute a fundamental matrix, linearly:

\[
F = \text{ComputeFundamentalMatrix}(u, v)
\]

Input: $u$ and $v$ are $N_f \times 2$ matrices of 2D correspondences where the $N_f$ is the number of 2D correspondences, $u \leftrightarrow v$.

Output: $F \in \mathbb{R}^{3 \times 3}$ is a rank 2 fundamental matrix.

(2) (Epipole and epipolar line) Pick 8 random correspondences, $u_r \leftrightarrow v_r$, compute the fundamental matrix, $F_r$ 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.
5 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 3 using the following pseudo code:

\begin{algorithm}
\caption{GetInliersRANSAC}
\begin{algorithmic}[1]
\State $n \leftarrow 0$
\For{$i = 1 : M$} \Comment{Choose 8 correspondences, $u_r$ and $v_r$, randomly from $u$ and $v$.}
\State $F_r = \text{ComputeFundamentalMatrix}(u_r, v_r)$
\State Compute the number of inliers, $n_r$, with respect to $F$.
\If{$n_r > n$} \Comment{Compute the number of inliers, $n_r$, with respect to $F$.}
\State $n \leftarrow n_r$
\State $F = F_r$
\EndIf
\EndFor
\end{algorithmic}
\end{algorithm}

(2) (Epipole and epipolar line) Using the fundamental matrix, visualize epipole and epipolar lines.
(3) (Camera pose estimation) Compute 4 configurations of relative camera poses:
\[ [R1 \ C1 \ R2 \ C2 \ R3 \ C3 \ R4 \ C4] = \text{CameraPose}(F, K) \]
Input: \(F\) is the fundamental matrix and \(K\) is the intrinsic parameter.
Output: \(R1\ C1\ \cdots\ R4\ C4\) are rotation and camera center (represented in the world coordinate system).

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

Figure 4: Four configurations of camera pose from a fundamental matrix.
CSCI 5980/8980: Assignment #4

Fundamental Matrix

6 Submission

- Assignment due: April 4
- Individual assignment.
- Problem 4, 5, and 6: MATLAB code submission in conjunction with visualization.
- Submission through Moodle.