ENGN1610 & 2605 Image Understanding
Lab08: 3D Reconstruction

The goal of this lab is to

- Learn how to estimate an Essential matrix under a RANSAC loop.
- Learn how to reconstruct a 3D scene by triangulation.

Problem 1. Scene Reconstruction Given two images, Figure 1, we are interested in reconstructing the scene in a 3D space. This can be achieved by first estimating an Essential matrix of image 2 with respect to image 1 in a RANSAC scheme, followed by a triangulation. The camera intrinsic matrix is provided in a `.mat` file. The necessary steps to follow are:

1. **Resize the images:** The size of the provided image is pretty large which might take a lot of time if we do triangulation on dense points. Resize the images using `imresize` by a factor of 0.25.

2. **Extract and Match SIFT Features:** Use VLFeat to extract and find SIFT feature correspondences as you have done in many previous labs.

3. **Estimate an Essential Matrix in a RANSAC loop:** Create the following function that returns an essential matrix `finalE` supported by the maximal number of inliers and the indices of inliers `inlierIndx`:

   ```
   function [finalE, inlierIndx] = Ransac4Essential(gamma1, gamma2, K)
   ```

   The inputs `gamma1` and `gamma2` are the feature correspondences, and `K` is the camera intrinsic matrix. You have the freedom to change the input names.

   (a) Pick five random pairs of feature correspondences in meters, compute all the valid essential matrices using `fivePointAlgorithmSelf.m` provided in the previous lab.

   (b) For each valid essential matrix, calculate the coefficients of the epipolar line. You have done this in the previous lab.

   (c) Using the epipolar line, count the number of inliers among all feature correspondences. A pair of correspondence is an inlier if the shortest distance between $\bar{\gamma}$ and the epipolar line is smaller than some threshold, *e.g.*, 2 pixels. Notation is borrowed from the previous lab.

   (d) Repeat steps (a) to (c) under an iterative loop with a predefined number of iterations. Set the number of iterations at least 5000.

   (e) Finally, after the iterative loop ends, pick the essential matrix supported by the maximum number of inliers as your output `finalE` (final essential) as well as a list of inlier indices `inlierIndx`.

   (f) Using the final essential matrix, draw at least three epipolar lines. You have done this in the previous lab.

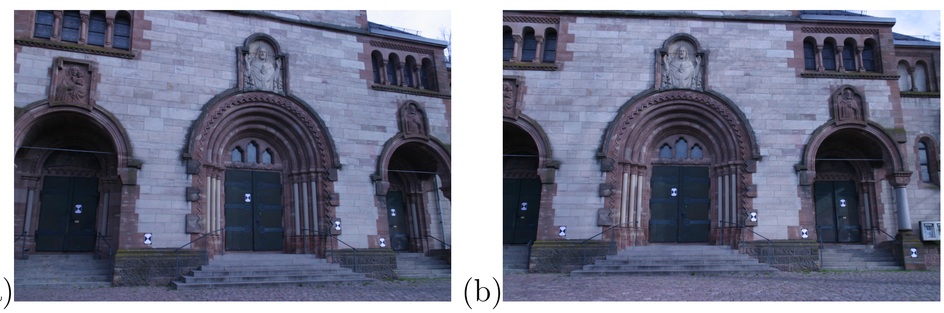
4. **Densify Correspondences:** The list of inlier matches returned by the RANSAC algorithm is usually too sparse for a visually pleasing 3D reconstruction. Thus, densifying the correspondences is necessary. Use the provided matlab function file `Densification.m` and refer to the instructions in that function file to see what the inputs and outputs are.

5. **Find Valid R and T from E:** This step consists of two stages:

   (a) Decompose the essential matrix `finalE` returned from your `Ransac4Essential` function into rotation matrices R and translation vectors T. There are four sets of possible R and T. Refer to the Appendix for more information on how to do find R and T from an essential matrix.

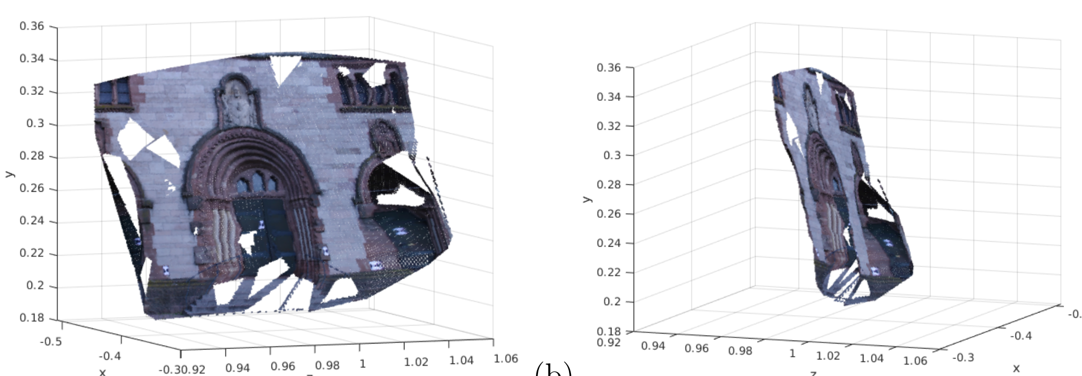
   (b) Only one out of four sets of R and T is valid. To find the valid set, pick one single random pair from the reliable dense correspondences and compute their respective depths ρ and $\bar{\rho}$. The rotation and translation is valid only if both ρ and $\bar{\rho}$ are positive. Also refer to the Appendix for more information on how to find the depths.

6. **Triangulation:** Using the valid R and T obtained from the previous step, loop over all reliable dense correspondences and compute the depths ρ and $\bar{\rho}$. The triangulated 3D points are thus $\Gamma = \rho\gamma$ and $\bar{\Gamma} = \bar{\rho}\bar{\gamma}$. Because of noise, Γ and $\bar{\Gamma}$ may not be necessarily identical. A typical way to do is to use the mid-point, *i.e.*, the average of Γ and $\bar{\Gamma}$, as the reconstructed 3D points. Store all 3D points and its color. Use `scatter3` to show your 3D reconstruction results. An example is given in Figure 2



(a) (b)

Figure 1: 3D scene reconstruction from different views.



(a) (b)

Figure 2: 3D scene reconstruction from different views.

Problem 2. Answer the Following Questions (This part is mandatory for ENGN2605 students but optional for ENGN1610 students who will get 5 extra points if all the questions are answered correctly.)

Question 1: Take a look at the `Densification.m` function file. How does the function work to make dense correspondences? Also, briefly explain the possible causes of the gaps of regions obtained in the 3D reconstruction.

Question 2: Can we estimate a meaningful essential matrix if there is no translation between the two cameras, *i.e.*, a pure-rotation case? Why or why not?

Appendix

1. **Find R and T from E**

   (a) **Method 1: Singular Value Decomposition (SVD)**
   Decompose an essential matrix E by SVD, *i.e.*, `[U,S,V]=svd(E)`, and define the matrix W as

   $$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

   then two rotations are

   $$\begin{aligned} R_1 &= UWV^T \\ R_2 &= UW^TV^T, \end{aligned}$$

   and two translations are the last column of U with two different signs. Two translations and two rotations compose four set of solutions.

   (b) **Method 2: Co-factor Matrix**
   From the constraint of an essential matrix,

   $$TT^T = \frac{1}{2}trace(EE^T)I - EE^T, \tag{2}$$

   the translation can be obtained by taking any row from the left hand side of the equation and normalize it, *i.e.*,

   $$\hat{T} = \frac{T}{||T||}. \tag{3}$$

   The two translations are $\hat{T}$ and $-\hat{T}$. The rotations can be subsequently computed from

   $$R = \frac{1}{||T||^2}cofactor(E)^T \pm \frac{1}{||T||^2}[T]_\times E, \tag{4}$$

   where the sign $\pm$ gives you two rotations. Two translations and two rotations compose four set of solutions.

2. **Depths Computation:** Let γ and $\bar{\gamma}$ be a pair of correspondences in meters, and their corresponding 3D points which project to γ and $\bar{\gamma}$ be Γ and $\bar{\Gamma}$, respectively. The relative geometry between two cameras is represented by a rotation matrix R and a translation vector T such that a 3D point whose representation in two cameras is related as

$$\bar{\Gamma} = R\Gamma + T, \tag{5}$$

or

$$\bar{\rho}\bar{\gamma} = R\rho\gamma + T, \tag{6}$$

where ρ and $\bar{\rho}$ are the depths of γ and $\bar{\gamma}$. Rewriting Equation 6 in a form of a linear system,

$$\begin{bmatrix} -R\gamma & \bar{\gamma} \end{bmatrix} \begin{bmatrix} \rho \\ \bar{\rho} \end{bmatrix} = T, \tag{7}$$

we can solve ρ and $\bar{\rho}$ given R and T. Note that $\begin{bmatrix} -R\gamma & \bar{\gamma} \end{bmatrix}$ is a 3×2 non-square matrix, to solve Equation 7 a pseudo-inverse method or a SVD method can be used. Be aware that the computed depths are up to a scale because of metric ambiguity in the geometry of relative pose estimation.