ENGN 2560 Computer Vision: Homework #3
Multiview Geometry: Image Formation & 3D Reconstruction

Due: March 15, 2019 midnight

In this programming assignment you will simulate capturing images from a 3D scene and, conversely, reconstructing the 3D scene captured by a pair of cameras.

An empty MATLAB script, main.m, is provided including few algorithmic parameter suggestions. Make use of helper methods that are called from main.m and make use of comments to improve code readability. Your homework submission should include your source code and a brief PDF report answering the questions with supporting visuals. The supporting visuals should demonstrate the outputs of the steps of your algorithm. Hand in your submission in a ZIP file with filename "FirstName_LastName_Homework3.zip".

NOTE: Vectorize your code as much as possible to achieve feasible run times.

Question 1

Given a dense 3D reconstruction of a scene represented by a point cloud, we are interested in forming the images that could be captured by a pair of cameras placed in this space.

Assume two cameras with a resolution of 1600 x 1200 (column x row) pixels are placed in a scene as shown in Figure 1. Using the camera intrinsic matrix, the point cloud storing the $[X, Y, Z]$ coordinates of each 3D point in world coordinates together with their colors stored in (R, G, B) format, and the R_1, T_1, R_2, T_2 matrices capturing the relative poses of cameras:

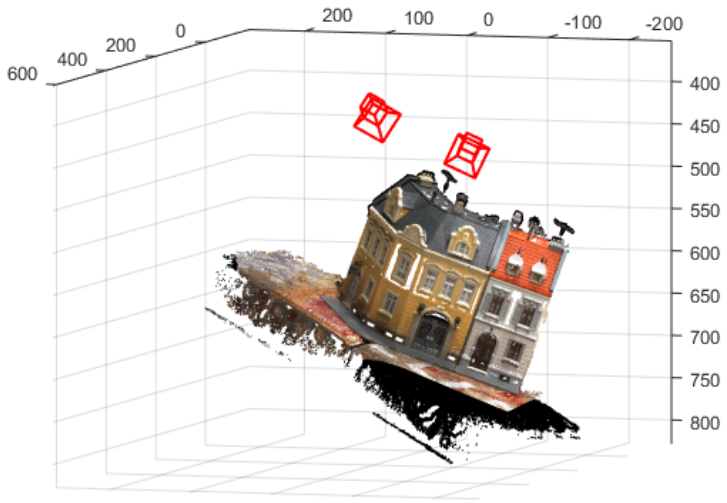

Figure 1: A 3D scene represented by a point cloud is viewed by two cameras.

(a) Write the expressions that project a 3D point in world coordinates onto the image planes of cameras in pixels.

(a) (b)

Figure 2: (a) Image captured by the first camera (b) Image captured by the second camera

(b) Implement an algorithm to construct the images that are captured by these cameras. Figure 2 shows the expected results. You may need to make use of meshgrid, and griddata to properly form the images. What is the principal point in pixels?

Question 2

Given a pair of images captured by a pair of cameras with different viewpoints, we are interested in reconstructing the scene in 3D space. Figure 3 shows the pair of images captured by the cameras. Assuming the camera intrinsic matrices are given, implement a 3D reconstruction algorithm that includes the following steps:

(a) (b)

Figure 3: (a) Image captured by the first camera (b) Image captured by the second camera

- Load the pair of images and the camera intrinsic matrices (the camera intrinsic matrices are the same for both cameras).
- Extract and match SIFT descriptors across the images using the VLFeat library. The function vl_ubcmatch could be used to obtain the pool of candidate matches.
- Implement a RANSAC algorithm to estimate the essential matrix, E. A 5-point algorithm that estimates the essential matrix using 5 correspondences taken at random is given which could be directly called in a RANSAC iteration. As the 5-point algorithm provided finds roots of a polynomial of order 10, at most 10 essential matrix estimates are returned by the solver. Using each estimate of the essential matrix, count the

number of inliers. The inliers are the candidate matches whose reprojection errors (the distance between the candidate match and the epipolar line obtained using the essential matrix) are below a threshold which is given as an algorithmic parameter. Maximize the number of inlier matches across RANSAC iterations. Return the essential matrix that leads to the maximum number of inliers as well as the inlier matches.

- Draw several **corresponding epipolar lines** on both of the images to verify the correctness of the essential matrix estimation. A few **point to epipolar line pairs** are shown in Figure 4.

(a)

(b)

(c)

Figure 4: Some point-epipolar line pairs obtained using the essential matrix estimate returned by the RANSAC algorithm.

- Decompose E into R and T as discussed in class. There are 4 possible (R, T) decompositions due to metric ambiguity. Given R, T, γ_1, γ_2 and K, write an expression that triangulates the matching point pairs (γ_1, γ_2) so that the depths (ρ_1, ρ_2) and the 3D points (Γ_1, Γ_2) could be recovered. Implement the triangulation algorithm. Triangulate all the inlier matches returned by the previous step using the 4 different (R, T) decompositions and pick the correct camera pose that returns the maximum number of 3D points for which the depths (ρ_1, ρ_2) are simultaneously positive.

- The list of inlier matches returned by the RANSAC algorithm is usually too sparse for a visually pleasing 3D reconstruction. We need to densify the number of correspondences before reconstructing the 3D scene. To do so, we can make use of the griddata function to interpolate and densify the correspondences. Specifically, given the sparse set of inlier correspondences between image 1 and image 2, the griddata function could be used to query for a correspondence at a pixel for which the correspondence is not known. As the images are quite big

in size, it is a good idea to interpolate the correspondences for every 4 pixels. Densify the correspondences from image 1 to image 2 as well as in the other direction, namely from image 2 to image 1, and filter out the correspondences that are not bidirectionally consistent. Bidirectionally consistent correspondences are the ones that almost form a closed loop. Starting at a pixel in image 1 we read off the corresponding pixel in image 2 using the densified image 1 to image 2 correspondences. We then traverse back to image 1 by reading the corresponding pixel in image 1 using the densified image 2 to image 1 correspondences. If the norm of the displacement vector from the starting pixel is less than some algorithmic threshold, the correspondence is deemed bidirectionally consistent.

- The densified bidirectionally consistent matches obtained from the previous step might violate the epipolar constraints as the densification step was solely based on interpolation being unaware of the epipolar constraints. From this dense set, filter out the correspondences that violate the reprojection error threshold previously defined above.
- Finally, triangulate all the remaining pairs of points and show the 3D point cloud using the triangulated points of positive depth. The point cloud could be colored by reading off the colors from either of the images.

Figure 5 shows the expected 3D reconstruction result using the pair of images shown in Figure 3.

(a) Briefly explain the possible causes of the gaps of regions obtained in the 3D reconstruction.

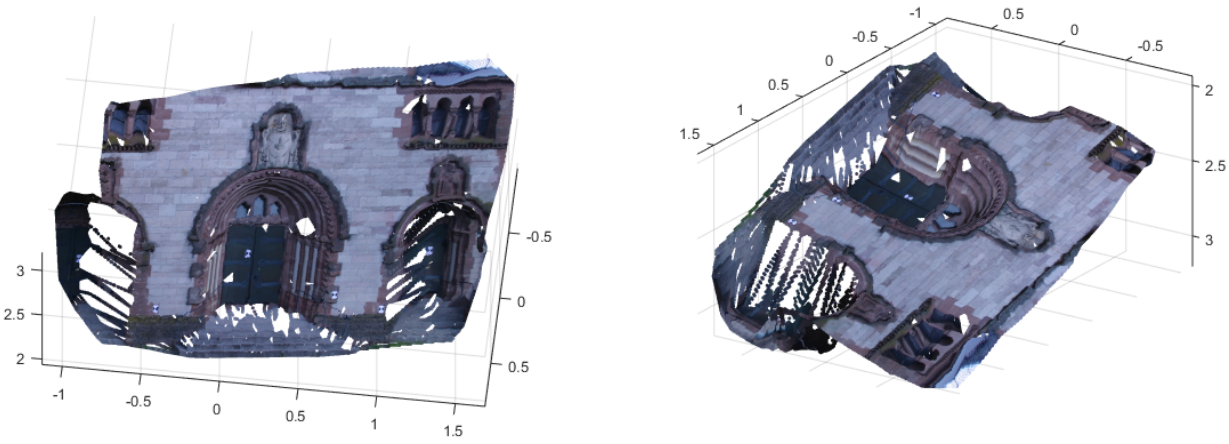

Figure 5: 3D reconstruction of the scene captured by a pair of cameras. Note how the depth of the structures are properly recovered.