CS364/AM792 : Computer Vision 16 September 2019

Assignment 4: 3D Reconstruction

Please prepare a single, condensed and neatly edited document for submission. For each problem include a short description of what you did, results, and a brief discussion/interpretation. It is not necessary to include any code in your document, unless a snippet helps to explain your method. Upload a zip-file containing your document and code to SUNLearn (under "Assignment 4") **before 17:00** on 30 September. Also hand in a printed copy of the document, at the latest during class on 1 October.

You are free to use any programming language. I recommend Matlab or Python. If you are asked to implement a specific technique, the idea is that you do so from scratch; do not simply use a function from an image processing or computer vision library. Collaboration is restricted to the exchange of a few ideas, and no form of plagiarism will be tolerated. All code, results, and write-up that you submit must be your own work.

1. You are given an image pair (`fountain1.jpg` and `fountain2.jpg`), a number of SIFT feature correspondences between the pair (`matches.txt`, in the same format as those given in Assignment 2), and 3×3 calibration matrices applicable to the images (`K1.txt` and `K2.txt`). The aim is to estimate the spatial relationship between the two views and reconstruct 3D coordinates of the image features.

(a) Display all the given feature correspondences over a greyscale version of image 1 (plot every match $\underline{x}_i \leftrightarrow \underline{x}'_i$ as a clear line segment from $\underline{x}_i$ to $\underline{x}'_i$, and place a marker at $\underline{x}_i$ to distinguish it from $\underline{x}'_i$). Then remove "obvious" outliers by enforcing a maximum distance that a feature is allowed to move in image space, and display the remaining matches. Considering the rest of the reconstruction pipeline in parts (b)–(e) below, why is it better to use a lenient threshold here (one that prefers the retainment of incorrect matches over the removal of correct matches)?

(b) Determine a fundamental matrix from the matches remaining after part (a), by implementing the RANSAC-based procedure from Lecture 17. Display the final set of inlier matches, and remember to re-estimate the fundamental matrix using this entire set.

(c) Use your fundamental matrix and the given calibration matrices to determine an essential matrix E. Calculate the SVD of E, say $E = U\Sigma V^T$, and display the three singular values (theoretically, the first two should be equal and the third should be zero). We will be using U and V as rotation (orthogonal, determinant 1) matrices, but the SVD algorithm may return more general orthogonal matrices (determinant ± 1) which may lead to unwanted reflections. Implement the following fix:

```
[U,S,V] := svd(E)
if (det(U) > 0) and (det(V) < 0) then  E := -E, V := -V
elseif (det(U) < 0) and (det(V) > 0) then  E := -E, U := -U
```

Note that this code may multiply the homogeneous matrix E with a constant, which is perfectly acceptable. We need not worry about the case [$\det(U) < 0$ and $\det(V) < 0$] because we will be working with a product of U and V. If both are reflections they cancel each other out.

(d) Determine two camera matrices from the essential matrix found in part (c) by the method discussed in Lecture 19, and use triangulation (Lecture 17) to pick the second camera matrix that produces 3D points in front of both cameras.

(e) Triangulate all the inlier matches, to create a feature-based 3D reconstruction of the objects in the images. Put some effort into showcasing your reconstruction: scale the three axes equally, colour the 3D points according to their colours in the images, indicate the positions and orientations of the two cameras in the world coordinate system (like in Assignment 3), remove background points far away from the cameras, and show the 3D plot from three or four different viewpoints.

p.t.o.

2. The reconstruction obtained in problem 1 is fairly sparse. Suppose we would like to upgrade it to a more dense representation of the observed scene. We will need to find more matching image points, necessitating **image rectification** (so that matching points have the same vertical coordinates in the two images).

(a) Use the camera matrices obtained in 1(d) to rectify `fountain1.jpg` and `fountain2.jpg`, by the procedure outlined in Lecture 17. Once you have the necessary transformation matrices you may use the "apply homography" function from Assignment 2 to warp the images.

Hint: if you struggle with memory or compute constraints, due to the large images, try to incorporate a downscaling effect in your transformation matrices.

(b) Draw corresponding epipolar lines over the pair of rectified images, which should now be almost as straightforward as drawing horizontal lines. Just note that the top-left corner of a rectified image may no longer coincide with the origin of the image coordinate system (why not?), and you should take that into consideration when identifying corresponding horizontal lines.

Hint: the values of `miny` calculated in the "apply homography" function can be useful here.

Hand in: 30 September 2019