COMP 558 Assignment 3

Prepared by Prof. Michael Langer
Posted: Sunday, November 15, 2020
Due: Sunday November 29, 2020 (by midnight, 11:59pm)

Instructions

Use the mycourses discussion board for any assignment related questions. This will help us reach the entire class when we respond. However, follow appropriate protocol, e.g., do not reveal the answer to a particular question or ask if your proposed solution is correct.

You are also free to discuss the questions with each other. *However, the solutions that you submit must reflect your own work and must be written by you.* You are not permitted to submit code or text that you have copied from one another or from any third party sources including internet sites.

Submit a single zipped file **A3.zip** to your mycourses Assignment 3 folder containing:

- A single PDF file with figures and text for each question.
- The Matlab code that you wrote so that we can run it if necessary. Submit a separate script for the two questions. The TAs should be able to run the script.
- The images that you used for the figures that you submit.

In order to receive full points, your solution code must be properly commented. Note that the TAs have limited time and will spend at most 20-30 minutes grading each submission.

Late assignment policy: Late assignments will be accepted up to only 3 days late and will be penalized by 10% of the total marks, per day late. For example, an assignment that is submitted 2 days late and that receives a grade of 90/100 would then receive an 18 point penalty. Submitting one minute after midnight on the due date means it is 1 day late (not a fraction of a day), and same for second day.

Question 1   Homographies and RANSAC [50 points]

For this question, you will get some experience working with homographies and real images.   Using your cell phone camera,   take two images of a scene from the same spatial position and for two different camera orientations.   That is, rotate the camera around an axis that passes through the lens (as best you can).   Examples are given with this assignment.

Your task will be to write a script that computes a homography between the two images for the purpose of image stitching, similar to the problem discussed in lecture 17 case 3 except that your homography will be estimated directly since the camera parameters are unknown.

Rather than using SIFT features which I mentioned in class, you will use SURF features (link to paper). The Matlab Computer Vision library provides functions for computing the SURF features from an image, and it also provides a function for finding a match between the features in the two images.   An example is given on this web page.   You will use these built in Matlab functions   (`detectSURFFeatures`, `extractFeatures`, `matchFeatures`) to detect features in a pair of images and find matching points between images.

Note that Matlab has an example of panoramic image stitching on this web page.   However, this example uses functions such as `estimateGeometricTransform2D` which you are not allowed to use.

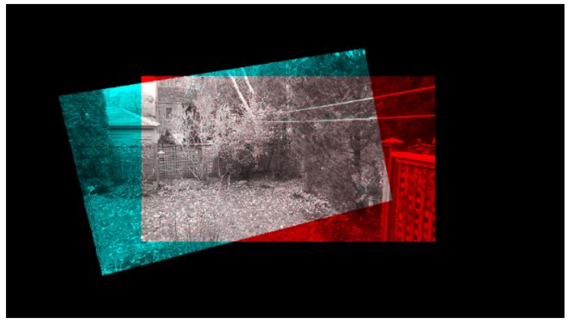
Your task is to (1) implement a RANSAC algorithm for fitting a homography between two given images and (2) to use the homography to stitch the two images together by creating a composite image. Here are some of the elements that your implementation must have:

- the main RANSAC loop must fit an exact homography using four randomly sampled matching pairs of features from the two images;
- after finding the exact homography that has the largest consensus set, fit the matching pairs in the consensus set using least squares (namely used Matlab's `svd` function); you must also normalize the locations of the features as described in the lectures which will give you a more reliable fit
- create a composite image such as shown in the example below;   use the R channel for image 1 and use the B and G channels for the pixels in image 2.   [Update Nov 16:   See the end of lecture 17 for a description of how to do this.   In particular, you are ***not*** allowed to use the Matlab function `imwarp`.

Examples are shown here for the `room` and `yard` images provided with this assignment.   Notice that image 1 is rectangular since it is the original image format, whereas the mapped image 2 is not rectangular.   In particular, opposite sides of the mapped image 2 are not exactly parallel.

A strong visual indicator that the alignment is correct is that the overlap region appears gray level only. This means that the image 1 and mapped image 2 have the same values at each pixel.

TIP: When debugging your code, we recommend that you try a few simple cases. Take some image and create a second one using the circshift or imrotate commands. Examine your homographies to ensure that do what you expect.



What to submit:

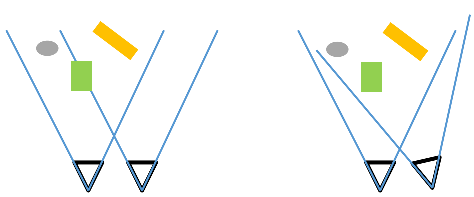
- a single file `homography.m`; this code should assume two images I1 and I2 are *already defined*. This will allow the T.A.s to easily run your code on a new pair of images to confirm that the code is correct.

- a pair of images taken by you, using your cell phone camera (or another student, in which case you should mention which student). Be sure to only rotate the camera, and not to translate it. *Do not submit any images taken from the internet.*

Question 2 Uncalibrated Stereo [50 points]

For this question, you will explore stereo vision in the uncalibrated setting. For expediency, you will make use of several functions in the Matlab Computer Vision library. See the Matlab documentation for how to use these functions. We expect you to rely heavily on the code in those examples, so feel free to copy and modify that code -- but do not copy code from anywhere else.

Create a stereo pair of images e.g. using your cell phone camera. Choose a 3D scene that contains a few objects that are at different depths. Camera 2 must be translated and rotated relative to camera 1 so that the two images cover roughly the same 3D volume as sketched in the figure below.



The objects in your scene need to have surface markings on them so that keypoints will be present which can be matched. It is common to use objects such as books or cereal boxes. Do not use objects with repetitive patterns since this will lead to false matches. If you use a small scene, then the ground surface should have surface markings too.

Make sure you don't swap the order of the images. Image 1 is the left image and image 2 is the right.

With your stereo pair of images, do the following:

- Use Matlab functions `detectSURFFeatures` and `extractFeatures` to find the locations of SURF features and to compute the descriptors, respectively. ("SURF" is a variation on SIFT that runs faster.)
- Use `matchFeatures` to obtain a list of matching features between images 1 and 2.
- Use `estimateFundamentalMatrix` to estimate the fundamental matrix using the matching SURF features. Use the 'RANSAC' method as a parameter for this function.
- Use the `estimateUncalibratedRectification` to estimate the transformations that rectify the two images. Then use `rectifyStereoImages` to perform this rectification.
- Estimate the disparities from the rectified stereo pair using `disparitySGM`.

For each of the above, follow the examples given on the Matlab doc webpages. You'll also need to consider the parameter options for some of these functions. Note that the examples in those web pages include figures that we ask you to submit below. We suggest that you generate those figures for yourself too which will help you in debugging. But only submit the figures that we ask for.

What to submit:

- a figure showing grayscale versions of the original left and right images
- a script `uncalibratedStereo.m` that produces the following figures
- a figure showing grayscale versions of the original left and right images with at least eight pairs of epipolar lines and matching points marked; use only *inlier* matching points. Write the value of the epipole in the title of the image figure.

  To compute the epipolar lines, use the estimated fundamental matrix. The match points should fall on the estimated epipolar lines. Use different color for different pairs of epipolar lines and matching points. This will make it easy to visually confirm the matching points/lines across the two images. You may use the Matlab `line` function to draw the epipolar lines.
- a figure showing the estimated disparity image; closer objects should have positive disparities and farther objects should have negative disparities.
- a roughly 500 word discussion of your results. What worked? What didn't work? Were any of your results different from what you expected? You may add a second example scene if that helps you to tell your story.

Important: We do not expect your fundamental matrix to be exactly correct, and we do not expect your estimated disparities to be "correct".   Uncalibrated stereo is a very challenging problem. *Indeed part of the purpose of this assignment question is for to appreciate the difficulty of the problem.*

Here is an example of what solution figures should look like, namely images showing the epipolar lines and matching points, and a disparity map.   [Update Nov. 16:   These images are provided for you in the zip file.]