School of Computer Science, University of Bristol

COMS30087: Image Processing and Computer Vision, AY: 2025/26

Assignment Part 2: 3-D from Stereo

Andrew Calway, `andrew@cs.bris.ac.uk`

This part of the assignment assumes that you have completed the two Stereo Lab Sheets. The assignment involves determining corresponding spheres in two views of a scene consisting of spheres on a plane and using the correspondences to reconstruct the spheres and their centre points. You are required to submit your code and a report. Both the code and the report are required to obtain marks.

Important: There are a number of ways in which the assignment can be tackled. However, you must use the methods described in the lectures. If you use any alternative methods, then zero marks will be given. Your submission must be your own work and the use of AI for either the code generation or the report is not allowed.

Assignment Tasks

You should complete the following tasks and then carry experiments as required for the report detailed under the submission requirements:

1. Download and run the assignment version of the 3-D simulator used in lab sessions. Refer to Lab Sheet I for details.
2. You should see 6 spheres of different sizes located on a plane. You will find two images in your current directory corresponding to the images captured by two virtual cameras (VCs). Each time you run the code you will see a different arrangement of the spheres and images captured from different viewpoints. You can change the number, separation and size of the spheres and show the sphere centres via command line arguments. Experiment with the different parameters to see the impact on the VC images.
3. Use the openCV function cv2.HoughCircles() to detect circles in each VC image. Check the results for several examples, noting any dependencies on any parameter settings.
4. Select one of the VCs as the reference view and for each detected circle centre in its image, compute the corresponding epipolar line for the other image. Draw the line and check that it is correct.
5. Use the epipolar line for each detected circle centre in the reference view to find the corresponding circle in the other view.
6. For each circle correspondence, compute the 3-D location of the associated sphere centre using the 3-D reconstruction algorithm described in the lectures.
7. Display the estimated sphere centres alongside the ground-truth centres in the visualisation. Compute the errors in the sphere centre estimates. Note: think carefully about how you should compute the errors.

8. Using the circle detections in the reference and viewing images, estimate the radius of each sphere.

9. Display the estimated spheres alongside the ground truth spheres and compute the error in the radius estimates. Think carefully about how you display the estimated and ground truth spheres so as to allow comparison, since comparison may be difficult if you display the estimates as solid spheres.

10. Investigate the performance of your implementation using different sphere sizes and separations. Note when it does work and when it fails, and find out why in the latter case.

Submission

1. **Code** Your code needs to be clearly commented. You must insert concise comments that clearly explain the implementation of key components and makes explicit reference to equations in the lecture slides (with slide numbers) or your report. Marks will be deducted if these explanations are not included. For example, an acceptable comment might be: "the following code computes the essential matrix defined as .... using the equation on slide ... in lecture .., where the variable ... is the .... and ... is the ....". Your code will be tested and the comments will be checked. It is important to provide a clear explanation as to how to run the code at the top of the file.

2. **Report** Your report should be no more than 4 pages and consist of the following sections:

   (a) **Epipolar lines and correspondence**. (1 page) Give full details of the algorithm you used to find corresponding circles. Justify its use and explain if and when it might fail. Provide an example of successful matching and an example of when it fails, showing the epipolar lines and the correspondence estimates in each case.

   (b) **Sphere centre estimation**. (1 page) Provide one example of estimated sphere centres after successful correspondence matching and one example when mismatches have occurred. Show the ground truth and estimated centres and provide a table listing the ground truth, estimated locations and the error. You must explain clearly the data in the table, including giving the definition of how you computed the error. You should also explain the impact of the mismatched cases.

   (c) **Sphere radius estimation**. (1 page) Give full details of the algorithm you used to compute estimates of the sphere radii. Justify its use, describe any assumptions that are made and explain if and when it might fail. Provide one example showing good estimates and one where the method fails, including the ground truth and estimated values in each case.

   (d) **Noisy pose**. (1 page) Investigate what happens if you use noisy relative pose between the cameras (position and orientation), rather than the ground-truth. Show examples of the impact on the correspondence matching and on estimation of the sphere centres and radii. Compare the errors obtained wrt the level of noise in the pose. You must give details of how you added noise to the pose and justify the method.

Page lengths are a guide - you can use more or less for each as long as the total number of pages does not exceed 4. For some of the above tasks you may wish to change the code so that the number and configuration of spheres gives a specific scenario rather than being random. This is fine, but you should add comments accordingly and comment out the code before submission so that random generation is restored.

Marking Scheme

This part is worth 50% of the coursework mark for the unit. Marks will be awarded based on the code and report **in combination**. Marks will be allocated according to the parts given below. To gain full marks for each part, you need to have completed all of the tasks relevant to the part, provided code that works and has comments as described above, and gives descriptions and examples that provides strong evidence that you have fully understood the tasks and results.

1. Epipolar lines and correspondence - 10%
2. Sphere centre estimation - 15%
3. Sphere radius estimation - 10%
4. Noisy pose - 15%