Department of Computer Science, University of Bristol

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

3-D from Stereo: Lab Sheet II

Andrew Calway, andrew@cs.bris.ac.uk
Thanks to Yuhang Ming

This lab sheet builds on Lab Sheet I and concerns epipolar geometry. It involves visualising and computing epipolar lines in views of a scene. It assumes that you are familiar with the 3-D simulator code used in Lab Sheet I and have completed all the tasks in the latter. Follow the tasks set out below. You are strongly encouraged to keep copies of all code and results.

1. Select one of the two VCs created in lab sheet 1. We denote this as the reference VC. Generate code to draw a 3-D line between the camera's centre of projection (COP) and the centre of one of spheres on the plane. Refer to the open3D documentation for drawing 3-D lines. Check the 3-D visualisation to make sure the line has been drawn correctly. You may also want to modify the code so that you can select the reference VC and sphere on the command line so that you can easily switch between configurations.

2. Now look at the image captured from the other VC. We will call this the viewing VC. You should see the 3-D line projected into this image and this should correspond to the epipolar line for the point corresponding to the projection of the sphere centre into the reference VC. Check that the epipolar line is correct and complete, i.e. extends across the whole of the image. If it is not complete, modify the code to show the complete epipolar line. **Extra**: if you created additional VCs for lab I, then check the projected lines in all the camera views.

3. By referring to the lecture slides, draw epipolar lines in each viewing VC image for a point in the reference VC using the relative 3-D pose of the viewing VC (position and orientation) and the intrinsic parameters of the camera, i.e. by computing the essential and fundamental matrices. You will need to compute the relative 3-D pose of each viewing VC with respect to the reference VC. Use the VC poses defined in the code w.r.t world coordinate system to compute the relative pose. You will also find the intrinsic parameters of the cameras in the code, which you will also need to compute the epipolar lines. You can use the openCV function cv2.line() to draw lines in the viewing VC images but you need to generate the code to determine the parameters of the line from the epipolar constraint equation. Do not use any openCV functions to do this.

4. Compare the epipolar lines drawn in task 3 with those visualised in task 2. Check they are the same. They may not be exactly the same. If not, why not?

5. Tidy up your code so that you can easily switch between different reference and viewing VCs and different spheres, checking that the epipolar lines in the captured images match your expectations.