Robotic 3D Vision
Prof. Dr. Jörg Stückler, Rui Wang
Winter Semester 2017/2018

Computer Vision Group
Department of Informatics
Technical University of Munich

Exercise Sheet 2

Topics: Graphical Models, Indirect Visual Odometry

Submission deadline: Wednesday, 29.11.2017, 23:59

Hand-in via email to rob3dvis-ws17@vision.in.tum.de

General Notice

All exercises can be done in teams of up to three students. Please hand-in your solution before the submission deadline, indicating names and matriculation numbers of your team members. Teams are encouraged to present their submitted solution during the exercise sessions.

Exercise 2.1: Bayesian Networks and Factor Graphs

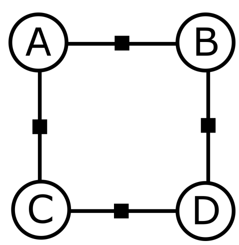

Figure 1: Factor graph for exercise 2.1.

(a) List the statistical conditional independence and dependence relations expressed by the factor graph in Fig. 1.

(b) Show that no directed graphical model (Bayesian network) can represent all the conditional independence and dependence relations of the factor graph in Fig. 1 over the same variable set.

Exercise 2.2: 2D-to-2D Motion Estimation

In this exercise, you will implement the eight-point algorithm to estimate camera motion between two images. You will also implement point triangulation to reconstruct the 3D position of corresponding points between the images.

(a) Extract the image files `0005.png` and `0007.png` from folder data/fountain in the exercise archive. The intrinsic camera calibration parameters for both

images are provided in the file
`camera_calibration.txt` as the camera intrinsics matrix,

$$\mathbf{C} = \begin{pmatrix} 2759.48 & 0 & 1520.69 \\ 0 & 2764.16 & 1006.81 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

(b) Find the 2D pixel coordinates of 8 corresponding point pairs in the images. Display the first image and select 8 points. You can use the command
`[x,y] = ginput(gcf)`
to obtain the pixel coordinates of mouse clicks. Select the corresponding 8 points in the second image in the same way and the same order. Transform the pixel coordinates to normalized image coordinates.

(c) Implement the eight-point algorithm from the lecture and run it with the point pairs.

(d) The eight-point algorithm provides 4 solutions. Choose a correct one by reconstructing the 3D point coordinates for each of the resulting relative camera motion using the triangulation method from the lecture.

Exercise 2.3: 2D-to-3D Motion Estimation

In this exercise, you will implement the direct linear transform algorithm to estimate camera motion between two images from 2D to 3D point correspondences.

(a) Extract the exercise archive to obtain the provided data files. The archive contains RGB and depth images in the data folders rgbd/rgb and rgbd/depth. The file names of the images specify the recording timestamps in seconds. In the following, associate the RGB with depth images by the closest timestamp. The file formats are described here: https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats

The RGB image timestamps of each subsequent pair are

$$\begin{aligned} P_1 &= (1305031102.175304, 1305031102.275326) \\ P_2 &= (1341847980.722988, 1341847982.998783) \end{aligned} \tag{2}$$

The corresponding camera intrinsics matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ of the RGB image pairs are:

$$\mathbf{C}_1 = \begin{pmatrix} 517.3 & 0 & 318.6 \\ 0 & 516.5 & 255.3 \\ 0 & 0 & 1 \end{pmatrix}, \ \mathbf{C}_2 = \begin{pmatrix} 535.4 & 0 & 320.1 \\ 0 & 539.2 & 247.6 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

(b) Find the 2D pixel coordinates of 6 corresponding point pairs in the images. Display the first image and select 6 points. You can use the command

`[x,y] = ginput(gcf)`
to obtain the pixel coordinates of mouse clicks. Select the corresponding 6 points in the second image in the same way and the same order. Transform the pixel coordinates to normalized image coordinates. Retrieve the depth of the selected pixels from the corresponding depth images in the first frame. Select a new pixel, if there is no valid depth at the selected pixel (invalid depth is indicated by zeros in the depth images).

(c) Implement the direct linear transform (DLT) algorithm from the lecture and run it with the point pairs to estimate the SE(3) transformation which transforms coordinates from the second to the first frame. Report the resulting transformation matrices.

(d) Use 2D-to-2D motion estimation to determine the relative transformations from second to first frame using the points from the previous task and two additional points. Rescale the estimates to match the measured depth from the first depth images. Report and compare your results with the transformations estimated with the DLT algorithm.

Submission instructions

A complete submission consists both of a PDF file with the solutions/answers to the questions on the exercise sheet and a ZIP file containing the source code that you used to solve the given problems. Note all names and matriculation numbers of your team members in the PDF file. Make sure that your ZIP file contains all files necessary to compile and run your code, but it should not contain any build files or binaries. Please submit your solution via email to `rob3dvis-ws17@vision.in.tum.de`.