

## Exercise Sheet 2

Topic: Graphical Models, Indirect Visual Odometry

### Exercise 2.1: Bayesian Networks and Factor Graphs

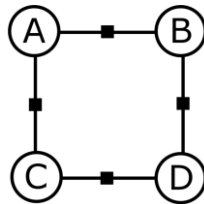


Figure 1: Factor graph.

- List the statistical conditional independence and dependence relations expressed by the factor graph in Figure 1.
- Show that no directed graphical model (Bayesian network) can represent all the conditional independence and dependence relations of the factor graph in Figure 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 the relative transformation between two images. You will also implement point triangulation to reconstruct the 3D position of corresponding points between the images.

- 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,

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

- 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` to obtain the pixel coordinates of mouse clicks (see `help ginput` for details). 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.
- Implement the eight-point algorithm and run it using the point pairs.

- d) The eight-point algorithm provides 4 solutions. Find the correct solution by reconstructing the 3D point coordinates from the 2D point correspondences and the resulting relative camera transformations.

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

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

- a) The exercise 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](https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats)

The RGB image timestamps of each subsequent pair are

$$\begin{aligned} P1 &= (1305031102.175304; 1305031102.275326) \\ P2 &= (1341847980.722988; 1341847982.998783) \end{aligned}$$

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

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

- b) Find the 2D pixel coordinates of 6 corresponding point pairs in the images. Display the first image and select 6 points. You can again use the command `[x,y] = ginput` 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.
- a) 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. Compare your results with the transformations estimated with the DLT algorithm.