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 5

Topics: Surfel-Pair Feature Matching, Iterative Closest Points Algorithm

Submission deadline: Wednesday, 24.01.2018, 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 5.1: Surfel-Pair Feature Matching

In this exercise, you will implement surfel-pair feature matching for detection and 6-DoF pose estimation of an arrangement of surfels in a point cloud.

(a) Load the surfels contained in the file `surfel_data.txt`. The data format is $p_x, p_y, p_z, n_x, n_y, n_z$ where the first three values are the 3D point coordinates and the last three values are specifying the normal of each point.

(b) Use the code in `object_model.m` to load the object model described as surfels.

(c) Implement a function that determines the surfel-pair features of a pair of surfels.

(d) Build a hash map of all surfel-pairs in the model based on a discretization of the surfel-pair features and a hash key on the discretized values. Hint: use a map container. The key type of the map will be integer while the value will be a list of pairs of surfel indices into the model.

(e) For each reference surfel in the model, determine a transformation $T_{m\to g}$ that translates the surfel point to the origin and aligns the normal of the surfel with the x-axis. For each surfel-pair, determine α_m which rotates the second surfel point into the half-plane spanned by the x- and positive y-axis.

(f) Implement surfel-pair voting: For each reference surfel in the scene, compute its transformation $T_{s\to g}$ analogously to $T_{m\to g}$. For each surfel pair, determine the angle α_s that aligns the second surfel point with the half-plane spanned by the x- and positive y-axis. Look up the matching model surfel-pairs in

the hash map according to the surfel-pair feature of the scene surfel pair. For each matched reference surfel in the model vote for the rotation angle $\alpha = \alpha_m - \alpha_s$ that aligns the surfel-pairs after transforming them into the global frame. Determine the model surfel and angle which receives the most number of votes.

(g) Cluster the pose votes of the scene reference surfels and determine the cluster with the most votes. Report the found object poses.

Exercise 5.2: Iterative Closest Points Algorithm

In this exercise, you will implement the iterative closest points algorithm to align depth images and to estimate the camera motion between the images.

(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

$$\begin{aligned} P_1 &= (1305031102.175304, 1305031102.275326) \\ P_2 &= (1341847980.722988, 1341847982.998783) \end{aligned} \tag{1}$$

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

$$\mathbf{C}_1 = \begin{pmatrix} 517.3 & 0 & 318.6 \\ 0 & 516.5 & 255.3 \\ 0 & 0 & 1 \end{pmatrix}, \ \mathbf{C}_2 = \begin{pmatrix} 535.4 & 0 & 320.1 \\ 0 & 539.2 & 247.6 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

Note: Convert the RGB images to floating point grayscale images before processing them. The depth images represent depth values by 16-bit integer values and need to be scaled by a factor of 1/5000 to obtain metric depth. Convert the depth images to floating point metric values before further processing them.

(b) Use the provided `downscale` function to downsample the images to 80×60 resolution. Convert the images to 3D point clouds.

(c) Implement the ICP algorithm by alternating the following steps: a) Establish point correspondences by finding the nearest neighbor for each point in one image within the other image based on the Euclidean distance. b) Find the transformation between the point sets using Arun's method.

(d) Iterate the ICP steps until convergence, i.e. the relative error falls below a threshold or a maximum number of iterations is reached. Report your results for the 2 image pairs and compare with the results from Ex 3.2.

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`.