

Name _____

SUNet ID _____

1. Image Generation

Consider the scenario depicted in Fig. 1, where an object O is moving while observed by a robot through an RGB-D camera C .

A point cloud from the scene is given in the file `point_cloud1.npy` (in the format of $(N \times 6)$ ndarray representing (x, y, z, R, G, B)). Some of the points are on the surface of the moving object, some of them are on the background. The coordinates of these points are given with respect to an absolute reference frame (world frame, WF).

The pose of the robot wrt. the same frame is given by the following translation and rotation (given in quaternions):

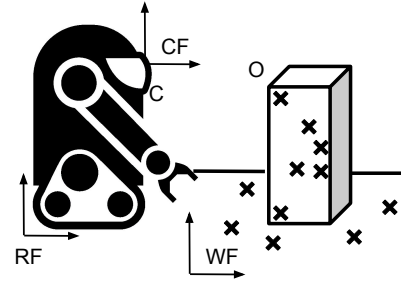


Figure 1

$$\mathbf{t}_{WF}^{RF} = (0.5, -9.2, 2.4) \quad (1)$$

$$\mathbf{q}_{WF}^{RF} = (w = 0.14, 0.28, 0.56, 0.76) \quad (2)$$

The camera is attached to the robot with the following translation and rotation (given in angle-axis notations) wrt. the robot frame:

$$\mathbf{t}_{RF}^{CF} = (0.1, -0.1, 1.2) \quad (3)$$

$$\hat{\mathbf{u}}_{RF}^{CF} = (0.707, 0.707, 0.0) \quad (4)$$

$$\theta = \pi/2.0 \quad (5)$$

The camera can be approximated by a pin-hole model with the following internal camera parameters:

$$P = \begin{bmatrix} 486 & 0 & 318.5 & 0 \\ 0 & 491 & 237 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

Let's assume that the robot moves with the following constant 6D velocity wrt. the world frame:

$$\xi_{WF}^{RF} = (v, \omega) = (0.1, -0.02, 0, 0, 0.09, 0.1) \quad (7)$$

The robot moves its head creating a constant 6D velocity of the camera wrt. the robot frame:

$$\xi_{RF}^{CF} = (0, 0, 0, 0, 0, 0.2) \quad (8)$$

The object is also moving with an unknown velocity. A second point cloud from the RGB-D camera after 2 s is given in the file `point_cloud2.npy`.

- a. (3 points) Compute the pose of the camera wrt. the robot frame, the pose of the robot wrt. the world frame, and the pose of the camera wrt. the world frame at $t = 0$ s, represented as 4x4 homogeneous transformation matrices.
- b. (3 points) Represent the relative motion of the robot wrt. the world frame using matrix exponential of a twist
- c. (1 points) Represent the relative motion of the camera wrt. the robot frame using matrix exponential of a twist
- d. (3 points) Compute the pose of the camera wrt. the world frame at $t = 2$ s.
- e. (4 points) Using `pointcloud1.npy`, calculate the 3D positions of the points in camera frame at $t = 0$ s.
- f. (8 points) Now calculate the positions of the points in 2D pixel coordinates. The resolution of the images is 640×480 pixels. Pixels projecting outside of the image plane should be excluded.
- g. (8 points) From the positions of the points in 2D pixel coordinates, generate two RGB images ($t = 0$ s and $t = 2$ s) as view from the camera using scatterplot.
- h. (10 points) The files `pointcloud1_ids.npy` and `pointcloud2_ids.npy` contain the point correspondences between timesteps. Take the points with id 4,9,20, and find the 6D transformation between the pairs of points. Give in homogeneous transformation matrix.
- i. (10 points) Apply RANSAC to estimate the 6D transformation of the object using all point correspondences.
- j. (5 points) From the transformation, calculate the 6D twist of the object. Provide this velocity with respect to the world frame.

2. Multi-View Geometry

Now, imagine the same scenario as in the previous exercise (Fig. 1) but now the object O is being interacting by a person, while the robot observes the interaction with its RGB-D camera C . Files `rgb1.png`, `rgb2.png` and `rgb3.png` contain three RGB frames of the interaction as observed by the robot. Files `depth1.txt`, `depth2.txt` and `depth3.txt` contain the registered depth values for the same frames.

- a. (20 points) Detect and track the $N = 200$ best point features using Luca-Kanade point feature optical flow algorithm on the consecutive images, initialized with Tomasi-Shi corner features. Use the opencv implementation (look at this tutorial for more information: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html). For the termination criteria use the number of iterations and the computed pixel residual from *Bouget*, 2001. Use only 2 pyramid levels and use minimum eigen values as an error measure. What is the lowest quality level of the features detected in the first frame? Provide a visualization of the flow for block sizes 15×15 and 75×75 . Explain the differences observed. Is it always better a larger or a smaller value? What is the trade-off?
- b. (25 points) Use the registered depth maps to estimate the sparse scene flow between frames for the features obtained in the previous steps. Assume that the camera is the one of the exercise 1. Provide the point clouds at the three frames if all values are not NaN.