University of Salzburg — Lecturer: Roland Kwitt

Imaging Beyond Consumer Cameras – Proseminar (911.422)

Exercise sheet **D**

Exercise 1. 8 P.

In the ICP algorithm from the lecture, we used the *point-to-point* error metric. Here, we are going to use the *point-to-plane* error metric and aim to register (wrt. rotation and translation only) two point clouds following a least-squares approach. In particular, we have two sets of points in $\mathbb{R}^3$, i.e., a source point cloud $\{p_i\}_{i=1}^N$ and a target point cloud $\{q_i\}_{i=1}^N$. The **error function** (without scaling) is given by

$$\sum_{i=1}^{N} \left((\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i)^\top \mathbf{n}_i\right)^2$$

and we want to minimize this sum wrt. $\mathbf{R}$ (rotation) and $\mathbf{t}$ (translation). In this setting, the normal vectors $\{\mathbf{n}_i\}_{i=1}^N$ in the *target* point cloud are also given. Assuming we have only *incremental rotations*, we can *linearize* the rotation matrix. For instance, in case of rotation in x, we linearize via

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{pmatrix}$$

This means, we approximate $\cos\alpha$ by 1 and $\sin\alpha$ by α. The overall rotation matrix thus becomes (approximately)

$$\mathbf{R} \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} .$$

The **linearized error function** thus simplifies to

$$\sum_{i=1}^{N} \left[\left((\mathbf{p}_i - \mathbf{q}_i)^\top \mathbf{n}_i\right) + \mathbf{r}^\top(\mathbf{p}_i \times \mathbf{n}_i) + \mathbf{t}^\top \mathbf{n}_i \right]^2 \tag{1}$$

with $\mathbf{r} = (\alpha, \beta, \gamma)^\top$ and $\mathbf{t} = (t_x, t_y, t_z)^\top$ denoting the translation vector.

Your task is find a **least-squares** solution to the problem of minimizing the **linearized error function** wrt. $\mathbf{R}$ and $\mathbf{t}$. The idea is to formulate Eq. (1) as

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

and use, e.g., numpy's least squares solver `np.linalg.lstsq`. In particular, this requires to appropriately construct the matrix $\mathbf{A} \in \mathbb{R}^{N\times 6}$ and the vector $\mathbf{b} \in \mathbb{R}^N$. Calling `np.linalg.lstsq` will then give you the least-squares solution $\mathbf{x} \in \mathbb{R}^6$ holding $(\alpha, \beta, \gamma, t_x, t_y, t_z)$ (the order depends on how you set up the matrices).

Once you have found the least-squares solution, transform the source point cloud according to the found $\mathbf{R}$ and $\mathbf{t}$ and visualize the result.

The provided Jupyter notebook, `Exercise-Sheet-D-Notebook-2021.ipynb` *contains additional details and helper code, e.g., code to obtain* $\mathbf{R}$ *from the found* α, β, γ.

Exercise 2. 2 P.

Consider the following list of set of seven points: $\{(12,8), (18,11), (15,15), (8,12), (3,18), (10,1), (17,5)\}$. Create a *median* k-D tree using *coordinate cycling* as the cutting dimension per level. Draw the k-D tree.

Exercise 3. 2 P.

Consider the point set from the previous question. Given a query point $\mathbf{q} = (4,5)$, (1) what is the closest point and (2) which subtrees of the k-D tree are not explored? Mark the *not-explored subtrees* in your drawing from the previous question.

Total #points: 12 P.