ICP (Iterative Closest Point)

Geometry Processing (GPR)

1 ICP

In order to obtain the rotation matrix for the simplified alignment problem, we will need to follow these steps:

1. First compute the centroid corrected versions of the point sets:

$$\tilde{\mathbf{p}}_\mathbf{i} = \mathbf{p}_\mathbf{i} - \tilde{\mathbf{p}} \qquad \tilde{\mathbf{q}}_\mathbf{i} = \mathbf{q}_\mathbf{i} - \tilde{\mathbf{q}}$$

2. Compute the covariance matrix $\mathbf{S}$ of both point sets:

$$\mathbf{S} = \mathbf{Q}\mathbf{P}^T = \begin{pmatrix} | & | & & | \\ \tilde{\mathbf{q}}_\mathbf{1} & \tilde{\mathbf{q}}_\mathbf{2} & \cdots & \tilde{\mathbf{q}}_\mathbf{n} \\ | & | & & | \end{pmatrix} \cdot \begin{pmatrix} — & \tilde{\mathbf{p}}_\mathbf{1} & — \\ — & \tilde{\mathbf{p}}_\mathbf{2} & — \\ & \vdots & \\ — & \tilde{\mathbf{p}}_\mathbf{n} & — \end{pmatrix}$$

3. Compute the singular value decomposition of $\mathbf{S}$:

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

4. Compute the optimal rotation matrix using $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T$$

5. If there is a reflection ($det(\mathbf{S}) = -1$), we need to cancel it. For that we multiply $\mathbf{R}$ by a matrix that results from changing the last element in the diagonal of an identity matrix from $+1$ to -1:

$$\mathbf{R} \leftarrow \mathbf{R} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 \end{pmatrix}$$

6. Compute the translation vector:

$$\mathbf{t} = \tilde{\mathbf{p}} - \mathbf{R}\tilde{\mathbf{q}}$$

The ICP algorithm then is:

1. For every point $\mathbf{q_i}$ in Q we find the closest point $\mathbf{p_j}$ in P.
2. The correspondence found on the previous step is used to find an optimal rigid transformation (rotation $\mathbf{R}$ plus translation $\mathbf{t}$). This is done using the SVD method we have already explained.
3. We apply the transformation $\mathbf{R}, \mathbf{t}$ to the points in Q.
4. We repeat this process from step 1, until there is convergence.

There are several ways to detect convergence:

- When the norm of $\mathbf{t}$ and the Frobenius norm of $\mathbf{R} - \mathbf{I}$ are smaller than a threshold.
- When the correspondence computed in step 1 is the same as it was in the previous iteration.

2 Detecting border points

There are several algorithms to detect if a point in a cloud is a border point or not. The one described here analyses the distribution of the neighbors of each point to determine if it is a border point or not.

Given a point cloud $P = \{\mathbf{p_i}\}$, the algorithm works as follows. For every point p_i in P:

1. Compute its k-nearest neighbors $\{\mathbf{p'_j}\}_{1 \leq j \leq k}$.
2. Use PCA to compute a local reference frame for point $\mathbf{p_i}$.
   (a) Compute the covariance matrix of the k-nearest neighbors. Remember to use the centroid adjusted points, not the points themselves.
   (b) The eigenvectors $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}$ are the axes of the reference frame, while point $\mathbf{p_i}$ is its center.
3. Transform the k-nearest neighbors to this frame.

$$(x_j, y_j, z_j) = ((\mathbf{p'_j} - \mathbf{p_i}) \cdot \mathbf{v_1}, (\mathbf{p'_j} - \mathbf{p_i}) \cdot \mathbf{v_2}, (\mathbf{p'_j} - \mathbf{p_i}) \cdot \mathbf{v_3})$$

4. Assuming that $\mathbf{v_3}$ is the eigenvector with the smallest eigenvalue (a good approximation of the surface's normal at point $\mathbf{p_i}$), we project on the XY-plane of the local frame.

$$(x_j, y_j, 0)$$

5. Now that the k-nearest neighbors are on the XY-plane, we compute the angles of their polar representation on that plane.

$$\alpha_j = atan2(y_j, x_j)$$

6. We order the angles and look for the largest difference between adjacent ones in the sorted sequence. We call this value $maxDeltaAlpha_i$.

7. If $maxDeltaAlpha_i$ is larger than a given threshold, $\mathbf{p_i}$ is a border point.