ICP Derivation

1 Introduction

Given 2 point clouds: destination $P = \{p_i\}_{i=1}^{n}$ and source $Q = \{q_j\}_{j=1}^{m}$, where p_i and q_j are three-dimensional vectors denoting a vertex in the point cloud. We need to find the rigid transformation—rotation and translation matrices (R, t) to transform the source point cloud Q to align it with the destination point cloud P

We can find the transformation matrices using several methods. In this report, we will discuss point-to-point in detail and mention the point-to-plane method.

1.1 Point-to-point

We find the transformation matrix by minimizing the Euclidean distances of the corresponding vertices from P and Q according to equation 1

$$\min_{(R,t)} E(R,t) = \sum ||p_i - Rq_i - t||^2, \tag{1}$$

where the pair of points p_i and q_i are correspondences. More about correspondences are in section 2

1.2 Point-to-plane

We find the transformation matrix by minimizing the Euclidean distances of the corresponding vertices from Q and the nearest plane of P according to equation 2

$$\min_{(R,t)} E(R,t) = \sum ||(p_i - Rq_i - t).\hat{n}||^2, \tag{2}$$

where $\hat{n}$ is the unit normal vector of the plane in P closest to vertex q_i.

2 Finding Correspondences

We can find the pair of points by getting the indices vertices in the source point cloud Q that are closest to each vertex in the destination point cloud P.

This can be done by finding the k-nearest neighbour from Q of each vertex in P

```
pt = NearestNeighbors(n_neighbors=1,
algorithm='kd_tree')
pt.fit(p)

dist, i = pt.kneighbors(q,
return_distance=True)
```

3 Point-to-point optimization function

To minimize equation 1, we differentiate E by R and t and set them to 0.

$$\frac{dE(R,t)}{dt} = \sum_{i=1}^{n}(2(p_i - Rq_i - t) = 0, \tag{3}$$

$$= \frac{\sum p_i}{n} - \frac{\sum Rq_i}{n} - \frac{\sum t}{n} = 0, \tag{4}$$

$$t = \bar{p} - R\bar{q}, \tag{5}$$

where $\bar(p)$ and $\bar(q)$ are the means of p and q.
To find R, we substitute Equation 5 in Equation 1.

$$\frac{dE(R,t)}{dR} = \sum_{i=1}^{n} ||(p_i - Rq_i + R\bar{q} - \bar{p})||^2, \tag{6}$$

$$= \sum_{i=1}^{n} ||(p_i - \bar{p}) + R(q_i - \bar{q})||^2, \tag{7}$$

let $\tilde{p_i} = (p_i - \bar{p})$ and $\tilde{q_i} = (q_i + \bar{q})$ $\forall i$. minimize $E(R)$ (Equation 8) such that $RR^T = I$ and $det(R) = 1$ (the conditions of rigid transformations).

$$E(R) = \sum_{i=1}^{n} ||(\tilde{p_i} - R\tilde{q_i}||^2, \tag{8}$$

We expand the equation and factor out the terms that are independent of R [1]

$$E(R) = \sum_i (\tilde{p_i} - R\tilde{q_i})^T(\tilde{p_i} - R\tilde{q}), \tag{9}$$

$$= \sum_i \tilde{p_i}^T\tilde{p_i} - \sum_i \tilde{p_i}^T R\tilde{q_i} - \sum_i \tilde{q_i}^T R^T\tilde{p_i} + \sum_i \tilde{q_i}^T R^T R\tilde{q_i}, \tag{10}$$

simplifying the equation by using the property. [2]
simplifying terms [3]

$$= \sum_i \tilde{p_i}^T\tilde{p_i} + \tilde{q_i}^T\tilde{q_i} - 2\sum_i \tilde{p_i}^T R\tilde{q_i} \tag{11}$$

[1] $||X||^2 = X^TX$
[2] If x^Ty is a scalar then $X^Ty = y^Tx$.
Therefore, $\tilde{q_i}^T R^T \tilde{p_i} = R\tilde{q_i}^T\tilde{p_i} = \tilde{p_i}^T R\tilde{q_i}$
[3] since $R^TR = 1$, $\tilde{q_i}^T R^T R\tilde{q_i} = \tilde{q_i}^T\tilde{q_i}$

removing terms independent of R [4]

$$= -2\sum_i \tilde{p_i}^T R\tilde{q_i} \tag{12}$$

The equation we got is negative, therefore we can define the optimization function as the negative of itself and maximize it instead of minimizing it, i.e., $\min_R E(R) = -\max_R E(R)$. After vectorizing everything we get equation 13, where $\tilde{P}^T$ and $\tilde{Q} \in \mathbb{R}^{n\times 3}$ and $R \in \mathbb{R}^{3\times n}$

$$\max_R E(R) = trace\ (\tilde{P}^T R\tilde{Q}) \tag{13}$$

4 Solving point-to-point optimization function

From equation 13, we need to find the final solution to find the transformation matrix R. The idea is that we need to find R in terms of P and Q. To do so, we compute the singular value decomposition of $\tilde{Q}\tilde{P}^T$

$$tr\ (\tilde{P}^T R\tilde{Q}) = tr(R\tilde{Q}\tilde{P}^T) = tr(RU\Sigma V^T) \tag{14}$$

We know that $M = V^T RU$ is orthogonal because $V^T V = U^T U = R^T R = I$.

Let M be defined as $M = \begin{bmatrix} | & | & | \\ m_1 & m_2 & m_3 \\ | & | & | \end{bmatrix}$

Therefore,

$$\Sigma M = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{15}$$

From this its clear that $tr(\Sigma M) = \sum_{i=1}^{3} \sigma_i m_{ii}$ as σ is a diagonal matrix.

Let's go back to equation 13. It transforms to

$$\max_R E(R) = tr(\Sigma M) = \sum_{i=1}^{3} \sigma_i m_{ii} \tag{16}$$

We know that each $\sigma_i \geq 0$, as the singular values cannot be negative. Also, from [5], we see that $|m_{ij}| \leq 1$ Therefore,

$$\max_R E(R) \leq \sum_{i=1}^{3} \sigma_i \tag{17}$$

To maximize this equation, $m_i i$ must be 1. Therefore we get

$$M = I \tag{18}$$

$$V^T RU = I \tag{19}$$

$$VV^T RUU^T = VIU^T \tag{20}$$

$$R = VU^T \tag{21}$$

but, determinant of $VU^T = \pm 1$, therefore, we need to adjust for it. Therefore.

$$R_{final} = V det(R) U^T \tag{22}$$

and

$$t = \bar{p} - R\bar{(q)} \tag{23}$$

5 discussion

We iteratively refine the transformation applied to Q to minimize the discrepancy between the P and Q. Usually, it is quite difficult to establish perfect correspondences because usually we want to align 2 parts of a single mesh together, which by definition means that the vertices in mesh P are different from vertices in mesh Q. However, the idea is that we find the closest match and get the best correspondences. Due to this, ICP works better for meshes with a higher number of matched vertices. Therefore, aligning a $0°$ mesh to $180°$ mesh is harder than aligning a $0°$ mesh to $45°$.

It's important to note that the transformation parameters R and t obtained at each iteration are not necessarily the final transformations but rather intermediate refinements. We can get the final transformation by performing point-to-point optimization on the final Q's vertex locations with the initial Q's vertex locations.

Other optimizations can be performed to get better results—

5.1 Sampling

You can uniformly sample points from P or Q or both to reduce the number of vertices the operations are performed on. This is especially useful for dense meshes.

5.2 filter matched correspondences

After finding the correspondences using 1-nearest neighbour, we can further filter the correspondences using angles of the normals of each vertex. This method is only possible if the normals and mesh faces are known. This helps us choose points that match well in both distance and direction and ensures that we are using the most trustworthy pairs of vertices for ICP.

Point-to-plane optimization algorithm is better than point-to-point for a similar reasoning. The derivation of point-to-plane optimization can be found in @see [Low, Kok-Lim. (2004)]

[4] $\tilde{p_i}^T\tilde{p_i} + \tilde{q_i}^T\tilde{q_i}$ is independent of R

[5] $M^T M = I, m_i^T m_i = 1, \sum_{i=1}^{3} = 1$ and $|m_{ij}| \leq 1$