Iterative Closest Points Review

Zhang Yuan

August 2, 2016

Abstract

This note summarizes the ways to compute the rigid transformation that aligns two sets of points, including the registration knowing correspondence between points, and the registration without knowing correspondence.

1 Problem Description

Let $P = \{p_1, p_2, ..., p_n\}$ and $Q = \{q_1, q_2, ..., q_n\}$ be two points set, and assume P can be transformed to Q through rotation and translation. We wish to find this rigid transformation. In other word, we want to minimize the energy term as follows:

$$(R, t) = argmin_{R,t} \sum_{i=1}^{n} w_i ||(Rp_i + t) - q_i||^2$$

In the following sections, we will talk about the algorithm to compute R and t under different assumption.

2 Least-Squares Rigid Motion Using SVD

Assume we know the accurate correspondences between points, we can compute the rotation matrix and translation vector using linear equation, but we can solve this problem in a more clever way.
Denote the energy term, we want to minimize the term:

$$F(t) = \sum_{i=1}^{n} w_i ||(Rp_i + t) - q_i||^2$$

We can find the optimal translation by taking the derivative of F with regrad to t:

$$0 = \frac{\theta F}{\theta t} = \sum_{i=1}^{n} 2w_i(Rp_i + t - q_i) = 2t(\sum_{i=1}^{n} w_i) + 2R(\sum_{i=1}^{n} w_i p_i) - 2\sum_{i=1}^{n} w_i q_i$$

Denote:

$$\bar{p} = \frac{\sum_{i=1}^{n} w_i p_i}{\sum_{i=1}^{n} w_i}, \quad \bar{q} = \frac{\sum_{i=1}^{n} w_i q_i}{\sum_{i=1}^{n} w_i}$$

Then we get:

$$t = \bar{q} - R\bar{p}$$

Let us replace t in object function:

$$\sum_{i=1}^{n} w_i||(Rp_i + t) - q_i||^2 = \sum_{i=1}^{n} w_i||R(p_i - \bar{p}) - (q_i - \bar{q})||^2$$

Denote:

$$x_i := p_i - \bar{p}, \quad y_i := q_i - \bar{q}$$

So now we want to find the optimal R such that:

$$R = argmin_R \sum_{i=1}^{n} w_i||Rx_i - y_i||^2$$

Now we can simplify the expression above:

$$||Rx_i - y_i||^2 = (Rx_i - y_i)^T(Rx_i - y_i) = x_i^T x_i - y_i^T Rx_i - x_i^T R^T y_i + y_i^T y_i$$

The rotation matrix has such property:$R^T R = I$. Denote all terms above is scalar, therefore:

$$x_i^T R^T y_i = (x_i^T R^T y_i)^T = y_i^T Rx_i$$

So we get:

$$||Rx_i - y_i||^2 = x_i^T x_i - 2y_i^T Rx_i + y_i^T y_i$$

Let's look at the minimization and substitute the above expression:

$$argmin_R \sum_{i=1}^{n} w_i||Rx_i - y_i||^2 = argmin_R \sum_{i=1}^{n} w_i(x_i^T x_i - 2y_i^T Rx_i + y_i^T y_i) = argmin_R(-2\sum_{i=1}^{n} w_i y_i^T Rx_i)$$

Denote that:

$$\sum_{i=1}^{n} w_i y_i^T Rx_i = tr(WY^T RX)$$

where $W = diag(w_1, ..., w_n)$ is an n*n diagonal matrix,$tr(A) = \sum_{i=1}^{n} a_{ii}$
Therefore we are looking for a rotation matrix R that maximize $tr(WY^T RX)$, because:

$$tr(AB) = tr(BA)$$

Therefore:

$$tr(WY^T RX) = tr(RXWY^T)$$

Let us denote $S = XWY^T$. Taking SVD of S:

$$S = U\Sigma V^T$$

Now substitute the decomposition into the trace:

$$tr(RXWY^T) = tr(RS) = tr(RU\Sigma V^T) = tr(\Sigma V^T RU)$$

Note that V, R and U are all orthogonal matrices, so $m = V^T RU$ is alos an orthogonal matrix. So, $m_j^T m_j = 1$ for each column in M.

$$1 = m_j^T m_j = \sum_{i=1}^{d} m_{ij}^2 \rightarrow |m_{ij}| \leq 1$$

Remember that Σ is a diagonal matrix with non-negative values on the diagonal. Therefore:

$$tr(\Sigma M) = \sum_{i=1}^{d} \sigma_i m_{ii} \leq \sum_{i=1}^{d} \sigma_i$$

Therefore the trace is maximized if $m_{ii} = 1$. Since M is an orthogonal matrix, this means that M would have to be an identity matrix:

$$I = M = V^T R U \rightarrow R = V U^T$$

In addition, this method not only compute the best rotation and translation, but also reflection. So in the case including reflection, this method can be applied. By calculating Det(R), we can know whether the answer includes reflection. If $det(VU^T) = -1$ it contains reflection.
If we only want the rotation without reflection, we can modify as follows:

$$tr(\Sigma M) = \sigma_1 m_{11} + ... + \sigma_d m_{dd} = \sigma_1 + \sigma_2 + ... + \sigma_d$$

Then we can find the second maximum.

$$tr(\Sigma M) = \sigma_1 + \sigma_2 + ... + \sigma_{d-1} - \sigma_d$$

So we can get rotation:

$$R = V[\begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{matrix}]U^T$$

Let us summarize the steps to compute the optimal translation vector t and rotation matrix R that minimize:

$$\sum_{i=1}^{n} w_i ||(Rp_i + t) - q_i||^2$$

(1) Compute the weighted centroids of both points sets.
(2) Compute the centerd vectors.
(3) Compute the matrix $S = XWY^T$.
(4) Compute the singular value decomposition $S = U\Sigma V^T$, then the rotation matrix R is VU^T.
(5) Compute the optimal translation as:

$$t = \bar{q} - R\bar{p}$$

3 Iterative Closest Points Algorithm

Above method bases on the known correspondence condition, what if we don't know the correspondence? ICP is an iterative algorithm for matching point sets. Consider 2 point sets, $A, B \in R^d$ where $\|A\| = n, \|B\| = m$. We are interested in a one-to-one matching function $\mu : A \rightarrow B$ that minimizes the root mean squared distance (RMSD) between A and B. Mathematically, we want to minimize the following energy:

$$RMSD(A, B, \mu) = \sqrt{\frac{1}{n} \sum_{a \in A} ||a - \mu(a)||^2}$$

Incorporating rotation and translation into matching, we can find out:

$$min \sum_{a \in A} ||Ra - t - \mu(a)||^2$$

The ICP algorithm seeks to minimize the RMSD, by alternating between a matching step and a transformation step. In the matching step, given a certain rotation and translation, the optimal matching is calculated by minimizing the RMSD. In the transformation step, given a matching, the optimal rotation and translation are computed. This alternating process terminates when the matching remains unchanged in sucessive iterations. The ICP procedure as follows:
ICP(A,B)
1. Initialize R = I(the identity matrix), t = 0.
2. Matching Step: Given R and t, compute optimal μ by finding $min_\mu RMSD(A, B, \mu)$.
3. Transformstion Step: Given μ, compute optimal R and t by finding $min_{R,t} RMSD(RA - t, B, \mu)$.
4. Go to step 2 unless μ is unchanged.

3.1 Matching Step

$\forall a \in A$, find closest $b \in B$ using KD-Tree or Voronoi diagram. The matching step is usually the slowest part of the algorithm.

3.2 Transformation Step

Assume $\mu(a_i) = b_i$, where $a_i \in A$ and $b_i \in B$. We want to find the matrix R and vector t that minimize:

$$\sum_{i=1}^{n} ||Ra_i - t - b_i||^2$$

We can use the result in above section:

$$R = VU^T$$

4 ICP Convergence

The ICP algorithm is guaranteed to converge to a local minimum. The algorithm terminates when the matching step does not change any of the previous matchings.