Iterative Closest Point Algorithm

Zhouyuan Chen

January 2024

1 Problem Description

This note is referred to from the original paper for ICP[1]. Just imagine we have two mesh A and B. For now, we want to find a transpose T and rotation R, making a move A to B's position as close as possible.

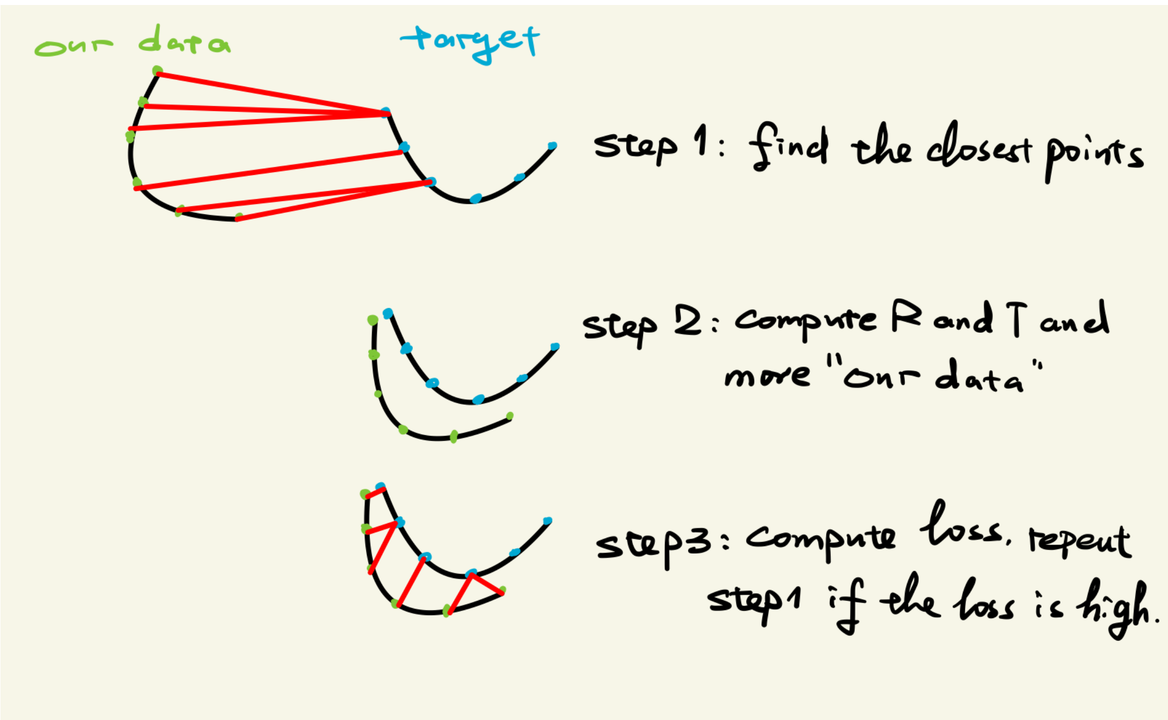

Figure 1: the picture of IPC algorithm's workflow

2 Derivation for the Point to Point Method

Let's recall what we want to minimize in this question, where x_i is "our data" and y_i stands for the "target":

$E = argmin\Sigma\|y_i - (Rx_i + t)\|_F^2$

$E = argmin\Sigma\|(y_i - \bar{y}) - (Rx_i + t - \bar{y})\|_F^2$ **(move the model to the reference coordinate)**

$E = argmin\Sigma\|(y_i - \bar{y}) - R(x_i + R^T(t - \bar{y}))\|_F^2$

Let $\tilde{x} = R^T(t - \bar{y})$, then $t = R\tilde{x} + \bar{y}$. Mention: As you can see, I chose $\bar{y}$ as the origin point.

$$E = argmin\Sigma\|(y_i - \bar{y}) - R(x_i + R^T(R\tilde{x} + \bar{y} - \bar{y}))\|_F^2$$
$$E = argmin\Sigma\|(y_i - \bar{y}) - R(x_i + \tilde{x}))\|_F^2$$
$$E = argmin\Sigma(tr((y_i - \bar{y})^T(y_i - \bar{y})) - 2(y_i - \bar{y})^T R(x_i + \tilde{x}) + (x_i + \tilde{x})^T R^T R(x_i + \tilde{x}))$$

The first term in the trace is constant, so we throw it.

$$\tilde{x}^*, R^* = \underset{\tilde{x}^*, R^*}{argmin}\Sigma(-tr(2(y_i - \bar{y})^T R(x_i + \tilde{x})) + tr((x_i + \tilde{x})^T(x_i + \tilde{x})))$$

Let's say:

$$F(\tilde{x}, R) = \Sigma(-tr(2(y_i - \bar{y})^T R(x_i + \tilde{x})) + tr((x_i + \tilde{x})^T(x_i + \tilde{x})))$$

Then we first compute the $\tilde{x}^*$:

$$\frac{\partial F}{\partial \tilde{x}} = \Sigma(-2(y_i - \bar{y})^T R + 2(x_i + \tilde{x})) = 0$$

We have:

$$\Sigma(x_i + \tilde{x}) = \Sigma(y_i - \bar{y})^T R$$
$$\Sigma(x_i + \tilde{x}) = 0$$
$$\tilde{x} = -\bar{x}_i$$

Therefore:

$$t^* = R^*\bar{x} + \bar{y}$$

We have:

$$R^* = \underset{\tilde{R}^*}{argmin}\Sigma(-tr(2(y_i - \bar{y})^T R(x_i + \bar{x})) + tr((x_i - \bar{x})^T(x_i - \bar{x})))$$
$$R^* = \underset{\tilde{R}^*}{argmax}\Sigma(tr((y_i - \bar{y})^T R(x_i - \bar{x})))$$
$$R^* = \underset{\tilde{R}^*}{argmax}\Sigma(tr(Y_i^T R X_i))$$
$$R^* = \underset{\tilde{R}^*}{argmax}\Sigma(tr(R X_i Y_i^T))$$
$$R^* = \underset{\tilde{R}^*}{argmax}(tr(RXY^T))$$
$$R^* = \underset{\tilde{R}^*}{argmax}(tr(RM))$$
$$R^* = \underset{\tilde{R}^*}{argmax}(tr(RUDV^T))$$

If we choose $R = VU^T$, then it becomes:

$$R^* = \underset{\tilde{R}^*}{argmax}(tr(VDV^T))$$
$$R^* = \underset{\tilde{R}^*}{argmax}(tr(VD^{\frac{1}{2}}D^{\frac{1}{2}}V^T))$$

For now, let's prove another inequality of the equation: $tr(X^T X) \geq tr(AX^T X)$, where A is a rotation matrix.

$$tr(AX^T X) = tr(XAX^T)$$

On the other hand, when X is a matrix belonging to $\Re^{N\times N}$:

$$XAX^T \leq \sqrt{X^TX}\sqrt{XA^TAX^T} = X^TX$$

Since $tr(X^TX) \geq tr(AX^TX)$, where A is a rotation matrix, the optimal R^* should be VU^T. Therefore, $R^* = VU^T$, $t^* = R^*\bar{x} + \bar{y}$, where $UDV^T = \Sigma((x_i - \bar{x})(y_i - \bar{y}))$.

References

[1] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 5, pp. 698-700, Sept. 1987, doi: 10.1109/TPAMI.1987.4767965. keywords: Economic indicators;Matrix decomposition;Singular value decomposition;Iterative algorithms;Motion estimation;Quaternions;Computer vision;Application software;Parameter estimation;Position measurement;Computer vision;least-squares;motion estimation;quaternion;singular value decomposition,