# The Iterative Closest Points Algorithm and Affine Transformations

Dmitrii Tihonkih[1], Artyom Makovetskii[1] and Vladislav Kuznetsov[1]

[1]Chelyabinsk State University, Chelyabinsk, Russia

sparsemind@gmail.com, artemmac@mail.ru, k.v.net@rambler.ru

**Abstract.** The problem of consistent aligning of 3D point data is known registration task. The most popular registration algorithm is the Iterative Closest Point (ICP) algorithm. One of the main steps of the ICP algorithm is matching. We find a matching in at first time on the basis of the geometric similarity of individual groups of points. It allows to get a good first approximation of the required transformation, even for big angle rotations, translations, scaling and noisy data. The step of the error minimization is performed for an arbitrary affine transformation.

**Keywords:** Iterative closest points (ICP), surface reconstruction, matching, error minimization

## 1 Introduction

The ICP (Iterative Closest Point) algorithm has become the dominant method for aligning three dimensional models based purely on the geometry. The algorithm is widely used for registering the outputs of 3D scanners, which typically only scan an object from one direction at a time. The standard ICP starts with two point clouds and an initial guess for their relative rigid-body transform, and iteratively refines the transform by repeatedly generating pairs of corresponding points in the clouds and minimizing an error metric. Generating the initial alignment may be done by a variety of methods, such as tracking scanner position, identification and indexing of surface features [1, 2], "spin-image" surface signatures [3], computing principal axes of scans [4], exhaustive search for corresponding points [5, 6], or user input. In this paper, we assume that a rough initial alignment is always available. In addition, we focus only on aligning a single pair of clouds, and do not address the global registration problem [7, 8, 9, 10]. Since the introduction of ICP by Chen and Medioni [11] and Besl and McKay [12], many variants have been introduced on the basic ICP concept. We may classify these variants as affecting one of six stages of the algorithm:

1. Selection of some set of points in one or both clouds.
2. Matching these points to samples in the other cloud.
3. Weighting the corresponding pairs appropriately.

4. Rejecting certain pairs based on looking at each pair individually or considering the entire set of pairs.
5. Assigning an error metric based on the point pairs.
6. Minimizing the error metric (variational subproblem of the ICP).

In this paper, we will look at variants for the categories 2 and 6. Our main focus is on the accuracy of the final answer and the ability of ICP to reach the correct solution for a difficult geometry. We consider affine transformation in $\mathbb{R}^3$ that holds the angles between lines in the cloud of points. An algorithm of matching is inspirited by the recent results of neurophysiology of vision [13]. Also we consider the ICP minimizing the error metric subproblem for the case of an arbitrary affine transformation. The computer simulation section contains results of computational experiments based on our matching and error minimizing approaches.

## 2 The matching procedure for sets X and Y

Let $X = \{x_0, \dots, x_{k-1}\}$ be an set consist of $k$ points in $\mathbb{R}^3$ and $Y = \{y_0, \dots, y_{n-1}\}$ be an set consist of $n$ points in $\mathbb{R}^3$. Here we describe our approach for searching the matching between $X$ and $Y$. Denote by $(x_i, y_j)$, $x_i \in X$, $y_j \in Y$ the pair of corresponding points. The goal of our procedure is representation of points from $X$ and $Y$ as sets of pairs. The first element of a pair belongs to $X$, the second element belongs to $Y$. Note, that each point from $X$ and $Y$ can be included to the set of pairs just one time. At the beginning the set of pairs is empty. Let $m \in \mathbb{N}$ be a number such that:

$$3 \le m \le \min(n, k). \quad (1)$$

Denote by $i$ a natural parameter.

1. Consider the following subset $X_i$ of the $X$:

$$X_i = \{x_{m*(i-1)}, \dots, x_{m*(i-1)+m-1}\}. \quad (2)$$

2. Let $C$ be a closed piecewise linear curve in $\mathbb{R}^3$ that consist of $m$ line segments. The $j$-th segment connects points $x_{m*(i-1)+j}$ and $x_{m*(i-1)+j+1}$. If $j+1 = m$ then we take index $m*(i-1)$ instead $m*(i-1)+j+1$. Denote by $\alpha_j$ a minimal flat angle that is constructed by $j$-th and $(j+1)$-th segments (with the similar agreement for the case $j+1 = m$). Let $V_X$ be a vector

$$V_X = \{\alpha_0, \dots, \alpha_{m-1}\}, \quad (3)$$

where elements $\alpha_j$, $j = 0, \dots, m-1$ are respective angles.

3. Consider all possible combinations of m points in the set Y besides the points that already included to the set of pairs. For an each combination we construct the vector $V$ by the same way as in step 2.

4. We choose a vector from the set of vectors of the step 3 such that distance between them and $V_X$ is minimal relatively the norm $L_1$. Denote this vector as $V_Y$.

5. We construct $m$ pairs of the points from $V_X$ and $V_Y$. Add this m pairs to the set of pairs.

6. If the number of remaining points in $X$ or $Y$ less that $m$ then procedure terminates. Else $i \coloneqq i + 1$ and go to step 1.

We use this procedure only as first iteration on the ICP algorithm. Obtained after the first iteration the transformation matrix and the translation vector are used for a second iteration. In the next iterations we use the standard nearest neighbor approach to find a match between the points.

In the practical using of the ICP algorithm very often a set $Y$ obtained from a set $X$ by a some geometrical transformation. The described above approach can good work not for rigid transformation only but for sufficiently wide subset of the affine transformations.

## 3 The ICP variational subproblem for an arbitrary affine transformation

Let $X = \{x_0, \ldots, x_{k-1}\}$ be a source point cloud and $Y = \{y_0, \ldots, y_{n-1}\}$ be a destination point cloud in $\mathbb{R}^3$. Suppose that the relationship between points in $X$ and $Y$ is done by such a way that for each point $x_i$ is calculated corresponding point $y_i$. In many works [11, 12, 14] the ICP algorithm is considered as a geometrical transformation for rigid objects mapping $X$ to $Y$:

$$Rx_i + t, \tag{4}$$

where $R$ is a rotation matrix, $T$ is a translation vector, $i = 0, \ldots, n-1$. The S-ICP algorithm [14] is a slightly different geometrical transformation given by

$$RSx_i + t, \tag{5}$$

where $S$ is a scaling matrix.

The group $E(3)$ of affine transformations in the dimension three has 12 generators. It means that affine transformation in dimension three is a function of 12 variables. Let us consider ICP variational problem for the case of an arbitrary affine transformation. Let $J(A, T)$ be the following function:

$$J(A,T) = \sum_{i=0}^{n-1} \| \; A\,x_i + t - \; y_i \; \|^2. \tag{6}$$

The ICP variational problem can be stated as follows:

$$\underset{\text{A,t}}{\arg min} \quad J(A,t), \tag{7}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}, \quad x_i = \begin{pmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{pmatrix}, \quad y_i = \begin{pmatrix} y_{1i} \\ y_{2i} \\ y_{3i} \end{pmatrix}. \tag{8}$$

One can be seen that

$$\begin{aligned} J(A,t) = \textstyle\sum_{i=0}^{n-1} \quad & ( a_{11}x_{1i} + a_{12}x_{2i} + a_{13}x_{3i} + t_1 - y_{1i})^2 + \\ & + ( a_{21}x_{1i} + a_{22}x_{2i} + a_{23}x_{3i} + t_2 - y_{2i})^2 + \\ & +( a_{31}x_{1i} + a_{32}x_{2i} + a_{33}x_{3i} + t_3 - y_{3i})^2. \end{aligned} \tag{9}$$

Let new coordinates $x_{ki}$ be expressed through old coordinates $x_{ki}$ as follows:

$$x_{ki} = x_{ki} - \frac{1}{n}\textstyle\sum_{j=0}^{n-1} x_{kj} \;, k = 1, \dots, 3, \; i = 1, \dots, n. \tag{10}$$

Also for the points of the second cloud we can write

$$y_{ki} = y_{ki} - \frac{1}{n}\textstyle\sum_{j=0}^{n-1} y_{kj} \;, k = 1, \dots, 3, \; i = 1, \dots, n. \tag{11}$$

Let us define coefficients $\alpha_i, \beta_i, \gamma_i, \varphi_i$ and $\psi_i$ for $i = 1, \dots, n$ as

$$\alpha_i = x_{2i} - \frac{x_{1i}}{\sum_{j=1}^{n} x_{1j}^{\;2}} - \textstyle\sum_{j=0}^{n-1} x_{2j}\, x_{1j} \,, \tag{12}$$

$$\beta_i = x_{3i} - \frac{x_{1i}}{\sum_{j=1}^{n} x_{1j}^{\;2}} - \textstyle\sum_{j=0}^{n-1} x_{3j}\, x_{1j} \,, \tag{13}$$

$$\gamma_i = y_{1i} - \frac{x_{1i}}{\sum_{j=1}^{n} x_{1j}^{\;2}} \textstyle\sum_{j=0}^{n-1} y_{1j}\, x_{1j}, \tag{14}$$

$$\varphi_i = \beta_i - \frac{\alpha_i}{\sum_{j=1}^{n} \alpha_j^{\;2}} \textstyle\sum_{j=0}^{n-1} \beta_j \;\; \alpha_j, \tag{15}$$

$$\psi_i = \gamma_i - \frac{\alpha_i}{\sum_{j=1}^{n} \alpha_j^{\;2}} \textstyle\sum_{j=0}^{n-1} \gamma_j \;\; \alpha_j. \tag{16}$$

**Proposition.** The elements of the first row of the matrix $A_*$ that minimizes $J$ are computed as

$$a_{11} = \frac{\sum_{i=0}^{n-1}( y_{1i} - a_{12}x_{2i} - a_{13}x_{3i} )\, x_{1i}}{\sum_{i=0}^{n-1} x_{1i}^{\;2}}, \tag{17}$$

$$a_{12} = \frac{\sum_{j=0}^{n-1} \gamma_j \;\; \alpha_j - a_{13} \sum_{j=0}^{n-1} \beta_j \;\; \alpha_j}{\sum_{j=0}^{n-1} \alpha_j^{\;2}}, \tag{18}$$

$$a_{13} = \frac{\sum_{k=0}^{n-1} \varphi_k\, \psi_k}{\sum_{k=0}^{n-1} \varphi_k^{\;2}}. \tag{19}$$

For the second and third rows of the matrix $A$ similar formulas can be easily derived.

## 4 Computer simulation

Let $X$ be the set consists of 80 points. The coordinates of points are randomly generated (by the uniform distribution). The values of all coordinates belong to the range $[0,\ldots,100]$. The set $Y$ is obtained from the set $X$ by the geometrical transformation $Y = R * X + t$, where $R$ and $t$ are described below:

$$R = \begin{pmatrix} 0.5 & 0 & 0.866025 \\ 0.866025 & 0 & -0.5 \\ 0 & 1 & 0 \end{pmatrix}, \tag{20}$$

$$t^{\mathrm{T}} = (5 \quad 6 \quad 7). \tag{21}$$

And each component of every point from the set $Y$ is noised by the following way.

$$y_j^i := y_j^i + n_j^i, \tag{22}$$

Here $n_j^i$ is uniformly distributed real number in the closed interval $[0, 1]$, j is the number of point, $i = \{1, 2, 3\}$.

Estimated using the our algorithm matrix $\tilde{R}$ and vector $\tilde{t}$:

$$\tilde{R} = \begin{pmatrix} 0.49987 & 0.00001 & 0.86612 \\ 0.865925 & 0.00009 & -0.500072 \\ -0.000027 & 0.999896 & -0.000021 \end{pmatrix}, \tag{23}$$

$$\tilde{t}^{\mathrm{T}} = (5.00281 \quad 6.0127 \quad 7.00388). \tag{24}$$

The standard approach based on nearest neighbor method implemented in the open source library LIBICP (C++ Library for Iterative Closest Points Matching) gives the following results:

$$\tilde{R} = \begin{pmatrix} 0.4281353 & 0.0278281 & 0.9032861 \\ -0.0109876 & 0.9996122 & -0.0255878 \\ 0.9036479 & -0.0010301 & -0.4282750 \end{pmatrix}, \tag{25}$$

$$\tilde{t}^{\mathrm{T}} = (-36.6521776 \quad 30.6680470 \quad 9.0428475). \tag{26}$$

Fig. 1 shows initial sets $X$ and $Y$.

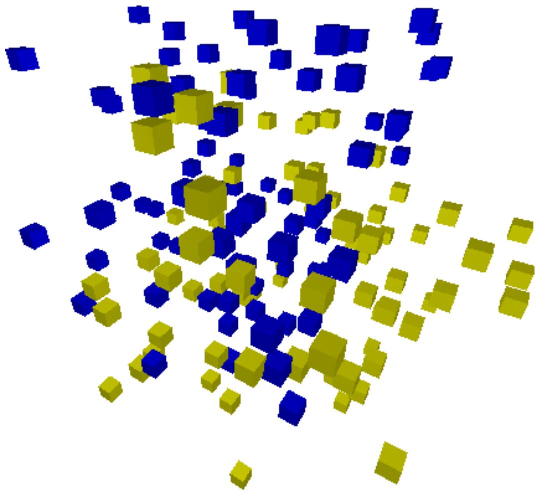

**Fig. 1.** Sets $X$ (yellow) and $Y$ (blue).

Fig. 2 shows a set $X$ and a set $\tilde{Y} = X * \tilde{R} + \tilde{t}$. Where $\tilde{R}$ and $\tilde{t}$ is the result of the standard approach.

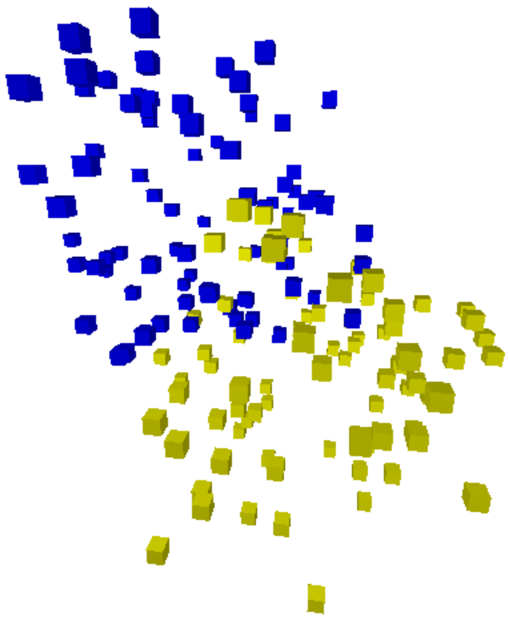

**Fig. 2.** Sets $X$ (yellow) and $\tilde{Y}$ (blue).

Fig. 3 shows a set $X$ and a set $\tilde{Y} = X * \tilde{R} + \tilde{t}$. Where $\tilde{R}$ and $\tilde{t}$ estimated by our algorithm after the first iteration.

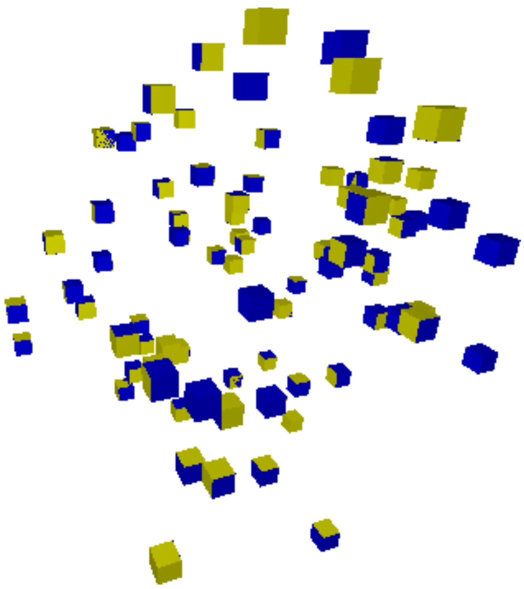

**Fig. 3.** Sets $X$ (yellow) and $\tilde{Y}$ (blue).

## 5 Conclusion

In this paper we considered matching and error minimizing steps of the ICP algorithm. On the base of the obtained results, a new efficient algorithm for the sets alignment was designed. The obtained results are illustrated with the help of computer simulation.

## Acknowledgments

The work was supported by Russian Science Foundation grant №15-19-10010.

## References


1. Faugeras, O. and Hebert, M., "The Representation, Recognition, and Locating of 3-D Objects," Int. J. Robotic Res., Vol. 5, No. 3 (1986).
2. Stein, F. and Medioni, G., "Structural Indexing: Efficient 3-D Object Recognition," Trans. PAMI, Vol. 14, No. 2 (1992).
3. Johnson, A. and Hebert, M., "Surface Registration by Matching Oriented Points," Proc. 3DIM (1997).
4. Dorai, C., Weng, J. and Jain, A., "Optimal Registration of Object Views Using Range Data," Trans. PAMI, Vol. 19, No. 10 (1997).
5. Chen, C., Hung, Y. and Cheng, J., "A Fast Automatic Method for Registration of Partially-Overlapping Range Images," Proc. ICCV (1998).
6. Chen, C., Hung, Y. and Cheng, J., "RANSAC-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images," Trans. PAMI, Vol. 21, No. 11 (1999).
7. Bergevin, R., Soucy, M., Gagnon, H. and Laurendeau, D., "Towards a General Multi-View Registration Technique," Trans. PAMI, Vol. 18, No. 5 (1996).
8. Stoddart, A. and Hilton, A., "Registration of Multiple Point Sets," Proc. CVPR (1996).
9. Pulli, K., "Surface Reconstruction and Display from Range and Color Data," Ph. D. Dissertation, University of Washington (1997).
10. Pulli, K., "Multiview Registration for Large Data Sets," Proc. 3DIM (1999).
11. Chen, Y. and Medioni, G., "Object Modeling by Registration of Multiple Range Images," Proc. IEEE Conf. on Robotics and Automation (1991).
12. Besl, P. and McKay, N., "A Method for Registration of 3-D Shapes," Trans. PAMI, Vol. 14, No. 2 (1992).
13. Petitot, J., "The neurogeometry of pinwheels as a sub-Riemannian contact structure," J. Physiology – Paris, 97, 265–309 (2003).
14. Langis, C., Greenspan, M. and Godin, G., "The parallel Iterative closest point algorithm," Proc. IEEE Third International Conference 3-D Digital Imaging and Modeling, 195-204 (2001).
15. Lowe, D. G., "Object recognition from local scale invariant features," Proc. 7th International conference on Computer Vision, 2, 1150–1157 (1999).