The Trimmed Iterative Closest Point Algorithm

D. Chetverikov, D. Svirko, and D. Stepanov
Computer and Automation Institute
Budapest, Kende u.13-17, H-1111 Hungary
csetverikov@sztaki.hu

Pavel Krsek
Center for Applied Cybernetics
FEE, Czech Technical University
166 27 Prague 6, Technicka 2, Czech Republic

Abstract

The problem of geometric alignment of two roughly pre-registered, partially overlapping, rigid, noisy 3D point sets is considered. A new natural and simple, robustified extension of the popular Iterative Closest Point (ICP) algorithm [1] is presented, called the Trimmed ICP (TrICP). The new algorithm is based on the consistent use of the Least Trimmed Squares (LTS) approach in all phases of the operation. Convergence is proved and an efficient implementation is discussed. TrICP is fast, applicable to overlaps under 50%, robust to erroneous measurements and shape defects, and has easy-to-set parameters. ICP is a special case of TrICP when the overlap parameter is 100%. Results of testing the new algorithm are shown.

1. Introduction

This paper[1] addresses the problem of Euclidean alignment of two roughly pre-registered, partially overlapping 3D point sets in presence of measurement outliers and, possibly, shape defects. This problem has been mainly considered in 3D model acquisition (reverse engineering, scene reconstruction) and motion analysis, including model-based tracking. (See [14] for an overview of recent applications.) Given two 3D point sets, $\mathcal{P}$ and $\mathcal{M}$, the task is to find the Euclidean motion that brings $\mathcal{P}$ into the best possible alignment with $\mathcal{M}$.

The Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [1] is a standard solution to the alignment problem. This iterative algorithm has three basic steps: 1. pair each point of $\mathcal{P}$ to the closest point in $\mathcal{M}$; 2. compute the motion that minimises the mean square error (MSE) between the paired points; 3. apply the motion to $\mathcal{P}$ and update the MSE. The three steps are iterated; the iterations have been proved to converge in terms of the MSE.

Independently, Chen and Medioni [2] published a similar iterative scheme using a different pairing procedure based on surface normal vector. This formulation is only applicable to points on surfaces. In this paper, we prefer the formulation by Besl and McKay which is applicable to volumetric as well as surface measurements.

The idea of ICP proved very fruitful as it was followed by numerous applications, improvements and modifications. A comprehensive survey oriented towards range images is provided in the PhD thesis by Pulli [10]. Rusinkiewicz and Levoy [13] give a fresh update of the variants of the ICP algorithm. They classify the variants according to the way the algorithms: (1) select subsets of $\mathcal{P}$ and $\mathcal{M}$; (2) match (pair) points; (3) weight the pairs; (4) reject some pairs; (5) assign error metric; (6) minimise the error metric.

Selection usually refers to random sampling of points when using a Monte Carlo technique, such as the Least Median of Squares (LMedS) [11, 14]. Pairs can be weighed or rejected based on the distribution of distances [16] or some geometric constraints [8]. Different cost functions and minimisation procedures are applied. For example, a recent paper by Fitzgibbon [6] presents an attempt of direct, rather than iterative, minimisation of the cost function (MSE) using the nonlinear Levenberg-Marquardt algorithm.

Most of the above modifications of ICP seek to improve robustness, convergence (speed) and precision. The most critical issue is probably that of robustness, as the original algorithm assumes outlier-free data and $\mathcal{P}$ being a subset of $\mathcal{M}$, in the sense that each point of $\mathcal{P}$ has a valid correspondence in $\mathcal{M}$. Numerous attempts have been made to robustify ICP by rejecting wrong pairs. In particular, robust statistics have been applied, such as LMedS or the Least Trimmed Squares (LTS) [12, 10]. Pajdla and Van Gool [8] proposed the Iterative Closest Reciprocal Point (ICRP) algorithm that exploits the ϵ-reciprocal correspondence: given a point $\mathbf{p} \in \mathcal{P}$ and the closest point $\mathbf{m} \in \mathcal{M}$, $\mathbf{m}$ is back-projected onto $\mathcal{P}$ by finding the closest point $\mathbf{p}' \in \mathcal{P}$. If $\|\mathbf{p} - \mathbf{p}'\| > \epsilon$, the pair $(\mathbf{p}, \mathbf{m})$ is rejected.

Often, different heuristics are combined, making the resulting ICP-variant efficient in cases when the underlying — sometimes, implicit — assumptions are met. Such het-

[1]To appear in Proc.ICPR'02, Québec City, 2002

erogeneous combinations are difficult to analyse; in particular, convergence properties remain unclear.

Computational efficiency is another important issue, since some applications require fast real-time operation for medium-size datasets, such as range images [10]. Various data structures, like k-D tree [7] or spatial bins [16], are used to facilitate search of the closest point. To speed up the convergence, normal vectors are considered, which is mainly helpful in the beginning of the iteration process [10].

In this paper, we concentrate on the issue of robustness. A new robustified extension of ICP is presented, called the Trimmed ICP (TrICP). The new algorithm is based on the consistent use of the Least Trimmed Squares (LTS) approach in all phases of the operation. LTS [12] means sorting the square errors and minimising a certain number of smaller values; LMedS [11] minimises the median, that is, the value in the middle of the sorted sequence.

Previously, LTS has only been used in the context of randomised, Monte-Carlo type initial estimation of the alignment parameters [10], following the guidelines of the classical approach [11] to robust regression and outlier detection. In this approach, model parameters are repeatedly estimated as random samples are drawn whose size is sufficient for the estimation. After the initial estimation, outliers are detected and rejected, and the final least squares solution is obtained for inliers only.

LTS is preferred to LMedS because it has better convergence rate and a smoother objective function [12]. However, as robust statistics in the context of a randomised approach, LTS and LMedS have the same breakdown point of 50%. This means that the overlap between the two point sets has to exceed 50%.

Our basic observation is that LTS fits the original scheme of ICP without any significant modification. At each step of iteration, the optimal motion can be computed for trimmed squares in exactly the same way as it is done in ICP for all squares. (The median of squares does not facilitate this computation, rendering the LMedS variant [14] inapplicable to large point sets.) At the same time, trimming the squares makes the algorithm robust in the original deterministic framework, without randomisation. The resulting algorithm, the Trimmed ICP, is applicable to overlaps under 50%. As no additional heuristics are used, the convergence of the algorithm is easy to prove.

The paper is organised as follows. Section 2 formulates the problem and presents the new algorithm. In section 3, convergence is proved and some implementation details are discussed. Results of tests are shown in section 4.

2. The new algorithm

Following the notation of [14], consider two sets of 3D points to align: the *data* set $\mathcal{P} = \{\mathbf{p}_i\}_1^{N_p}$ and the *model* set $\mathcal{M} = \{\mathbf{m}_i\}_1^{N_m}$. Usually, the numbers of points in the two sets are different: $N_p \neq N_m$. A large portion of the data points may have no correspondence in the model set. Assume the minimum guaranteed rate of the data points that can be paired is known; we will call this rate the *minimum overlap* and denote it by ξ. Then, the number of the data points that can be paired is $N_{po} = \xi N_p$.

If the value of ξ is unknown, one can run TrICP several times and select a result that combines a good MSE with the highest possible overlap. A procedure for automatic setting of ξ is given in section 3.

Like most iterative algorithms, including ICP, our algorithm assumes that $\mathcal{P}$ and $\mathcal{M}$ have been *roughly pre-registered*, either manually or automatically. This can be done, for example, by aligning a few characteristic points or, in a controlled measurement setup, by calculating the sensor motion between the two views. It should be emphasised, however, that the initial alignment can be fairly rough: TrICP has been successfully applied to the initial relative rotations of up to 30°.

Also, it is assumed that the overlapping part of the two sets is characteristic enough to allow for unambiguous matching. In particular, this part should not be symmetric and 'featureless'. This assumption is typical for most point set registration algorithms. A possible way to cope with this problem is discussed in section 3.

Under these assumptions, the problem is to find the Euclidean transformation that brings an N_{po}-point subset of $\mathcal{P}$ into the best possible alignment with $\mathcal{M}$. For an Euclidean motion with rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, denote the transformed points of the data set by

$$\mathbf{p}_i(\mathbf{R},\mathbf{t}) = \mathbf{R}\mathbf{p}_i + \mathbf{t}, \quad \mathcal{P}(\mathbf{R},\mathbf{t}) = \{\mathbf{p}_i(\mathbf{R},\mathbf{t})\}_1^{N_p} \qquad (1)$$

Define the *individual distance* from a data point $\mathbf{p}_i(\mathbf{R},\mathbf{t})$ to the model set $\mathcal{M}$ as the distance to the closest point of $\mathcal{M}$:

$$\mathbf{m}_{cl}(i,\mathbf{R},\mathbf{t}) = \arg\min_{\mathbf{m}\in\mathcal{M}} \|\mathbf{m} - \mathbf{p}_i(\mathbf{R},\mathbf{t})\| \qquad (2)$$

$$d_i(\mathbf{R},\mathbf{t}) = \|\mathbf{m}_{cl}(i,\mathbf{R},\mathbf{t}) - \mathbf{p}_i(\mathbf{R},\mathbf{t})\| \qquad (3)$$

We wish to find the motion $(\mathbf{R},\mathbf{t})$ that minimises the sum of the *least* N_{po} squared individual distances $d_i^2(\mathbf{R},\mathbf{t})$.

The conventional ICP algorithm assumes that all data points can be paired: $\xi = 1$ and $N_{po} = N_p$. TrICP provides a smooth transition to ICP as $\xi \to 1$.

The structure of TrICP is similar to that of ICP. The basic idea of is to consistently use the least trimmed squares (LTS) in all major aspects of operation: to cope with outliers, shape defects, or just partial overlap; to estimate the optimal transformation at each iteration step; and to form the global cost function which is minimised. The main steps of TrICP are as follows. These steps are iterated until any of the stopping conditions described below is satisfied. The iterations are started with $S_{LTS} = huge\_number$.

Algorithm 1: Trimmed Iterative Closest Point

1. For each point of $\mathcal{P}$, find the closest point in $\mathcal{M}$ and compute the individual distances d_i^2 (eq.(3)).
2. Sort d_i^2 in ascending order, select the N_{po} least values and calculate their sum S'_{LTS}.
3. If any of the stopping conditions is satisfied, exit; otherwise, set $S_{LTS} = S'_{LTS}$ and continue.
4. Compute for the N_{po} selected pairs the optimal motion $(\mathbf{R}, \mathbf{t})$ that minimises S'_{LTS}.
5. Transform $\mathcal{P}$ according to $(\mathbf{R}, \mathbf{t})$ (eq.(1)) and go to 1.

We use the standard stopping conditions [14] related to the number of iterations N_{iter} and the MSE for the N_{po} selected pairs: (1) the maximum allowed N_{iter} has been reached, or (2) the *trimmed MSE* $e = S'_{LTS}/N_{po}$ is sufficiently small, or (3) the relative change of the trimmed MSE $|e - e'|/e$ is sufficiently small. Note that in [14] absolute rather than relative change is tested.

3. Implementation and convergence

Like any variant of the ICP, a fast *implementation* of the TrICP needs an efficient data structure supporting the closest point search. In step 1, we use a simple boxing structure [3] that partitions the space into uniform boxes, cubes. Given a point in space, only the box containing this point and the adjacent boxes are to be considered during the search. The box size is updated as the two sets get closer.

The heap sort [9] is used to efficiently sort the distances in step 2. The optimal motion in step 4 is computed by the unit quaternion method due to Horn [5]. The same method was used in the original version of the ICP [1].

When the value of the overlap parameter ξ is unknown, we set it automatically by minimising the objective function $\psi(\xi) = e(\xi)\xi^{-(1+\lambda)}$, where $\lambda \geq 0$ is a preset parameter. (In the tests described in section 4, we used $\lambda = 2$.) $\psi(\xi)$ minimises the trimmed MSE $e(\xi)$ while trying to use as many points as possible. Increasing λ, one can attempt to avoid undesirable alignments of symmetric and/or 'featureless' parts of the two sets.

The minimum of $\psi(\xi)$ is searched in the range $[0.4, 1.0]$, which is a typical range of overlaps. The objective function is quite smooth. Usually, 5–8 iterations are sufficient to locate the minimum with the necessary precision.

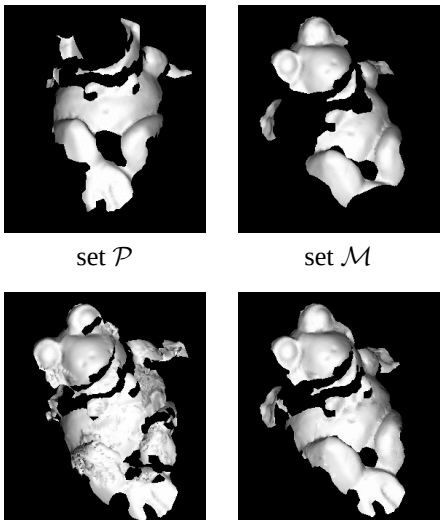
Many previous attempts to robustify the ICP used some additional geometric or statistical heuristics which were not mathematically coherent with the original idea. The TrICP incorporates the robust LTS statistics in a way compatible with the philosophy and data structure of the ICP. An important advantage of this natural extension is that the *convergence* of the TrICP can be proved. The following theorem is valid, whose proof is given in [4].

Theorem: The Trimmed Iterative Closest Point algorithm always converges monotonically to a local minimum with respect to the trimmed MSE objective function.

Convergence to global minimum depends on the starting point. To avoid local minima, the ICP is usually run several times at different conditions. Varying ξ one can also run the TrICP at different conditions and select the best result.

4. Tests and discussion

Figure 1 compares the ICP and the TrICP in aligning two partially overlapping and differently rotated measurements of the Frog. Each of the two sets has about 3000 points. Some numerical results are shown in table 1, including number of iterations and the execution time on a 1.6 GHz PC. The TrICP alignment is better and faster. (Note: As the ICP is a special case of the TrICP, the same program is run with different values of ξ.)



set $\mathcal{P}$ set $\mathcal{M}$

result of ICP result of TrICP

Figure 1. Aligning two measurements of Frog.

Table 1. Numerical results for Frog data

Method	N_{iter}	MSE	Exec.time
ICP (100%)	45	5.83	7 sec
TrICP 70%	88	0.10	2 sec

A systematic study is in progress, aimed to quantitatively compare the TrICP to the ICP, the ICRP [8] and other meth-

ods for large sets of 2D and 3D shapes. Some results are presented in tables 2 and 3. They were obtained for the SQUID fish contour database available at the web site [15] of the University of Surrey, UK. The database contains two-dimensional shapes of 1100 different fishes.

To form $\mathcal{P}$, the original shape is rotated by a known angle. $\mathcal{M}$ preserves the original orientation. Then, different non-overlapping parts of $\mathcal{M}$ and $\mathcal{P}$ are deleted so as to provide a desired overlap. Finally, noise is added to both shapes.

The fully automatic version of TrICP is applied to align $\mathcal{M}$ and $\mathcal{P}$. This means that the known overlap ξ_a is *not* passed to the algorithm: ξ is set automatically using the procedure described in section 3. In most cases, the obtained ξ and the actual ξ_a are very close, which does not necessarily mean that ξ_a is always optimal for alignment.

Tables 2 and 3 present mean absolute differences between the ground-truth rotation and the rotations obtained by the TrICP and ICRP [8] for the 1100 shapes at various rotations (degrees) and overlaps (per cents). To obtain the best possible result, for each alignment the ICPR is run a few times. Each time, the input is the output of the previous run, and the parameters of the ϵ-reciprocal correspondence are modified accordingly. Note that iterations of the ICRP may not converge, although in practice this rarely happens.

Table 2. TrICP errors for SQUID data, degrees

	100%	90%	80%	70%	60%
1°	0.0512	0.0829	0.0701	0.0984	0.1879
5°	0.0509	0.0858	0.0797	0.1216	0.3411
10°	0.0517	0.0917	0.0984	0.1915	0.5800
15°	0.0509	0.1091	0.1646	0.3380	1.1430
20°	0.0502	0.0953	0.2025	0.6942	1.7949

Table 3. ICRP errors for SQUID data, degrees

	100%	90%	80%	70%	60%
1°	0.0531	0.0612	0.0762	0.1226	0.2259
5°	0.0541	0.0652	0.1110	0.1763	0.3079
10°	0.0542	0.0655	0.1826	0.5988	1.7008
15°	0.0609	0.1108	0.3625	1.0878	2.5363
20°	0.1076	0.1614	0.4871	1.5114	3.0254

At the current state of the experimental study, we experience that the execution times of the two algorithms are comparable, which is not surprising. For the SQUID dataset, the TrICP usually more accurate than the ICRP. At small rotations, the difference is not significant. However, the TrICP is more robust to rotation and incomplete, noisy data. Its proved convergence is also an advantage. The results need more thorough analysis. Additional tests are under way to systematically assess the algorithms in 3D.

Acknowledgement. This work was supported by the EU grant ICA1-CT-2000-70002: MIRACLE - Centre of Excellence and by the Hungarian Scientific Research Fund under grant OTKA M28078. Pavel Krsek was supported by the Czech Ministry of Education under project LN00B096.

References

[1] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[2] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10:145–155, 1992.

[3] D. Chetverikov. Fast neighborhood search in planar point set. *Pattern Recognition Letters*, 12:409–412, 1991.

[4] D. Chetverikov and D. Stepanov. Robustifying the Iterative Closest Point Algorithm. In *Third Hungarian Conference on Image Processing and Pattern Recognition*, pages 1–9, 2002.

[5] D. Eggert, A. Lorusso, and R. Fisher. Estimating 3-D Rigid Body Transfromations: A Comparison of Four Major Algorithms. *International Journal of Machine Vision and Applications*, 9:272–290, 1997.

[6] A. Fitzgibbon. Robust Registration of 2D and 3D Point Sets. In *British Machine Vision Conference*, 2001.

[7] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. on Mathematical Software*, 3:209–226, 1977.

[8] T. Pajdla and L. V. Gool. Matching of 3-D Curves using Semi-differential Invariants. In 5^{th} *International Conference on Computer Vision*, pages 390–395, 1995.

[9] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[10] K. Pulli. *Surface Reconstruction and Display from Range and Color Data*. PhD thesis, University of Washington, Seattle, 1997.

[11] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley Series in Probability and Mathematical Statistics, 1987.

[12] P. Rousseeuw and B. van Zomeren. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85:633–651, 1990.

[13] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *Third International Conference on 3D Digital Imaging and Modeling*, 2001.

[14] E. Trucco, A. Fusiello, and V. Roberto. Robust motion and correspondence of noisy 3-D point sets with missing data. *Pattern Recognition Letters*, 20:889–898, 1999.

[15] Univ. of Surrey, UK. Shape Queries Using Image Databases. www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html.

[16] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13:119–152, 1994.