CIRCULAR TRAJECTORY CORRESPONDENCES FOR ITERATIVE CLOSEST POINT REGISTRATION

Olli Suominen , Atanas Gotchev

olli.j.suominen@tut.fi, atanas.gotchev@tut.fi
Tampere University of Technology
Department of Signal Processing
P.O. Box 553, FI-33101 Tampere, Finland

ABSTRACT

Iterative closest point (ICP) is a popular algorithm for finding rigid transformations between 3D point clouds. It aims to find the rotation and translation differences between the point clouds. A key component in the algorithm is finding correspondences between the two data sets, which are then used to determine the differences. A method for finding these pairings is described, utilizing the circular trajectory of points when the cloud is being rotated. The proposed method reveals more information per iteration cycle than techniques previously used for 3D data. The method enables the use of an efficient implementation by using a simple data structure, which has the same computational complexity to build and to access as the k-d tree commonly used with nearest neighbor correspondence searches.

The experimental results show that the convergence rate is superior to the original ICP based on point-to-point minimization and compares favorably to more refined and complex approaches, e.g. normal shooting with point to plane minimization. This together with the efficient implementation strategy and low amount of computation per iteration makes the circular trajectory correspondences (CTC) a valid choice for registration tasks, especially in applications where processing power is limited.

Index Terms — Convergence, Geometry, Image registration, Iterative algorithms, Iterative closest point algorithm, Iterative methods

1. INTRODUCTION

Iterative Closet Point is a very popular algorithm for registering two point clouds. On a higher level, the problem can be formulated as follows: Given two point clouds A and B, identify the operation which turns B into A. Assuming a rigid transformation, the difference is explained by translation and rotation components. With 3D point clouds, this brings a total of 6 degrees of freedom:

- Translation t in directions x, y, z
- Rotation R around axis x, y, z

A formal description of the problem is to find

$$\underset{R,t}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{a}_i - (R\boldsymbol{b}_i + \boldsymbol{t})\|^2, \quad (1)$$

where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are *corresponding points* in two N point clouds. In practice, the correspondences between the points are not known, which prompts the usage of algorithms such as ICP. Otherwise, the problem would be a simple set of linear equations with a closed form solution.

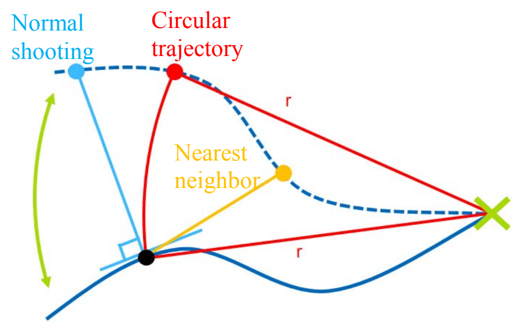

Figure 1. Basic concept of finding the correspondences along the circular trajectory of the reference point when the rotation is happening around the X. For comparison, nearest neighbor and normal shooting are also presented.

This fundamental issue is what prompts the use of an iterative scheme. In the lack of actual knowledge, candidate correspondences are generated, and Eq. 1 is solved for the assumed point pairs. The underlying assumption is that the artificial correspondences will bring the system closer to converging on the true alignment. The method of finding these candidate pairs as it was originally introduced is as the name would suggest: for a given point in the reference point cloud, assume, that the closest point, i.e. nearest neighbor (NN), in the other cloud is the correspondence [1].

The algorithm consists of multiple stages, all of which have been the target of further optimization attempts [2]. Among other things, earlier work on ICP has suggested many different ways of determining point-wise correspondences. The nearest neighbor idea has been found to be a working assumption, but not nearly optimal. A strong contender is dubbed *normal shooting* [3].

2. CIRCULAR TRAJECTORY CORRESPONDENCES

The proposed method is based on the idea that when the rotational difference between clouds dominates, information on the difference can be extracted best from points sharing a common trajectory around the center of rotation. I.e., when looking for corresponding points, preference should be given to point pairs with the same distance r from the current center of rotation (Figure 1). While it may appear that the assumption is only valid in the case of point clouds sharing co-located centroids, a slightly relaxed radius criterion will allow it to handle also the translation component in other cases. If the clouds consist of non-overlapping areas, none of the studied alternatives on their own can handle the registration. This would require e.g. regularized minimization [4].

978-1-4799-1369-5/13/$31.00 ©2013 IEEE

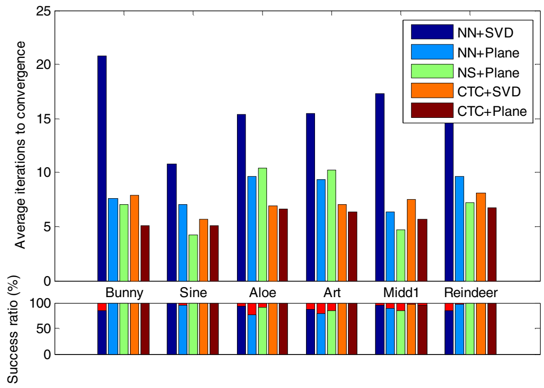

Figure 2. Average number of iterations needed for the method to converge to the correct position over 50 initial random rotations. The lower section shows the success rate, i.e. how many of the 50 tests converged to the correct end result in 30 iterations.

A similar idea [5] has been suggested for registering 2D range scans, but the concept has been overlooked in 3D point clouds registration. The proposed method can thus be considered as an extension of this to cover 3D point clouds. In the ideal, noiseless case where the rotation is the only difference between the two clouds, the true correspondence for a given point a_i satisfies

$$r_i = \|c_A - a_i\| = \|c_B - b_i\|, \tag{2}$$

where c_A and c_B are the centroids, i.e. centers of mass for the clouds A and B. However, in practice, because of the effects of noise, it is required that the point pair satisfies

$$\big|\|c_A - a_i\| - \|c_B - b_i\|\big| < \Delta r, \tag{3}$$

Δr being the adjustable parameter, which tunes the noise tolerance of the correspondence search. The tighter the parameter, the better matches it will find, but intuitively, the parameter should be proportional to the amount of noise. Noteworthy is that with a high enough Δr, the matching procedure is exactly equivalent to nearest neighbor. Should the choice of the parameter be done adaptively, this gives the performance a well-known lower limit.

From those candidate points that fulfill the criterion in Eq. 3, a reasonable assumption is to select the one with the shortest Euclidian distance. Strictly speaking, it could be argued that the distance should be measured along the trajectory, but this entails quite a lot of work, and will result in the same relative ordering of points as using the Euclidian.

3. DATA STRUCTURE

The simplest way to implement CTC matching is to pose it as a rejection mechanism for point pairs found via simple nearest neighbor search. While functionally equivalent to a proper implementation, the overhead is a very high as massive amounts of unnecessary computations of point-to-point distances will be performed. The method also invalidates the use of k-d tree structures, as points on the circular trajectory do not follow the same spatial partitioning scheme.

A straightforward, efficient implementation is to represent the point clouds as lists of elements $a_i = (r_i, x_i, y_i, z_i)$, sorted by the radius r. When querying the potential correspondence points, it suffices to find the correct range of radii, $[r_i - \Delta r, r_i + \Delta r]$. The sorting can be done in $O(n \log n)$, which corresponds to the complexity of building a k-d tree commonly used in nearest neighbor searches. The cost of accessing an element in a k-d tree is $O(\log n)$, which is the same as in making a binary search in the radius sorted list. Furthermore, it is possible to optimize this as consecutive accesses will be directed towards consecutive elements in the list, although similar tricks may be possible also for a k-d tree.

In comparison to normal shooting, the complexity of the operations is also the same. However, the amount of actual work needed to find the intersections between normals and a reconstructed surface is significantly more than the simple comparisons required for either NN or CTC, not to mention the need for an actual surface, which in turn should be reconstructed explicitly. In case of point-to-plane minimization, it may become necessary to recompute and re-sort the radius lists if the translation component exhibits the kinds of changes sometimes associated with the minimization scheme. Still, if the changes in translation stay relatively small, relaxing Δr may be sufficient.

4. EXPERIMENTAL RESULTS

Tests on convergence rate of the proposed algorithms were performed by comparing against the following methods:

- Nearest neighbor correspondences, SVD minimization
- Nearest neighbor correspondences, point-to-plane minim.
- Normal shooting correspondences, point-to-plane minim.

According to [2], normal shooting coupled with point-to-plane minimization exhibits the fastest convergence out of the well-known combinations. The circular correspondence search is tested both with SVD and point-to-plane minimization. The error metric is used is the root-mean-square (RMS) error between the true ground truth correspondences. The result is considered as converged when the error between true correspondences stabilizes under 110% of the RMS of the added noise, which acts as a practical lower bound of the achievable registration error.

For all methods, initial alignment of the point clouds is done by centering centroids on zero. This starting point makes the comparison between methods more reasonable. While SVD-based variants do this on the first iteration as it is a part of the algorithm, point-to-plane takes several rounds to minimize this original displacement. An additional culling step is tried by removing 10% of the pairs with the longest distance. Without the removal of these pairs, normal shooting does not converge at all. The other methods are on average performing better without it.

4.1 Test data

The datasets used in the tests are a number of ground truth depth maps from the Middlebury database [6] and an artificial depth map constructed from sinusoidal waves. The main intended use case is registration of data originating from range scanners such as time-of-flight cameras, consequently represented as depth maps. Additionally, to see how the method fares in a more traditional setting where the data is a 3D model mesh, the iconic Stanford bunny [7] is included.

To simulate noise in acquiring the depth maps, Gaussian zero mean noise to the depth component of the point clouds is added. Given a specific devise, better assumptions on the type of noise could be made, but for the purposes of these tests, this can be considered sufficient. To test the convergence rate in number of iterations, each data set is transformed to a point cloud, rotated and translated randomly and then registered using the listed methods. This is repeated 50 times for random rotations in the range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ around each of the three axes. Due to the preprocessing step of centering the centroids of the point clouds, translation component does not play a significant role.

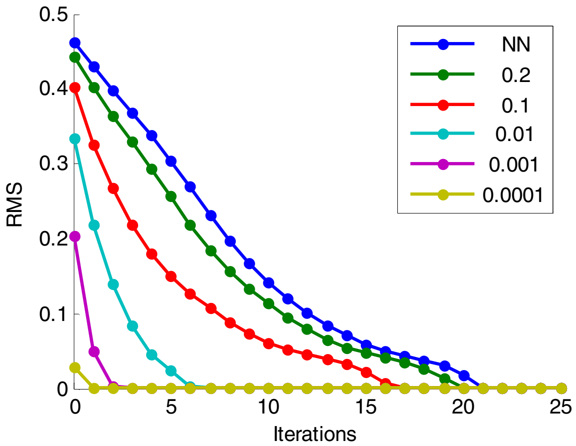

Figure 3. The effect of the correspondence search parameter Δr is demonstrated on an ideal case of registration between two identical point clouds using SVD minimization. Nearest neighbor (NN) correspondence is shown as a reference.

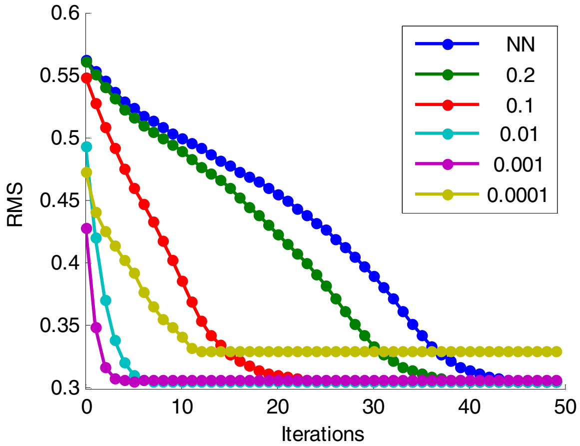

Figure 5. SVD/CTC registration, where 10% of the coordinates of both input clouds have respectively been replaced with statistically independent random outliers. (Y-axis starting from 0.3 due to RMS sensitivity.)

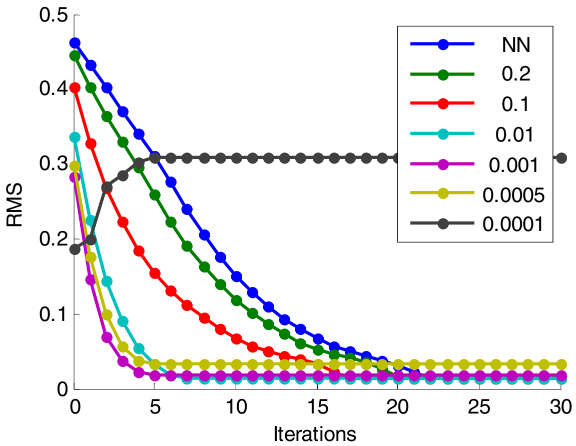

Figure 4. Convergence of point-to-point minimization via SVD and circular correspondences with different values of Δr. The clouds have respectively independent noise ($\sigma = 0.01$) added to them, which invalidates the use of the tightest threshold parameters. A good choice of Δr still converges to the minimum

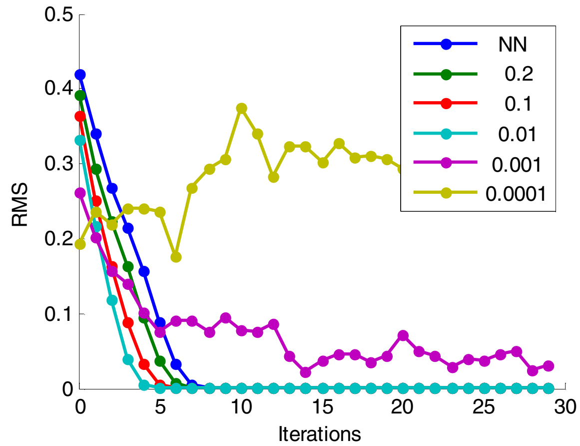

Figure 6. Similarly to Figure 3, the effect of the parameter Δr is presented, but now in the case of point-to-plane minimization. Despite the clouds having no noise, the tighter thresholds do not allow the small changes in the translation component induced by the characteristics of the minimization method. Potentially, an adaptive, decreasing threshold would alleviate the issue and yield faster convergence.

4.2 Convergence measures

In the tested data sets (Figure 2), CTC matched correspondences consistently lead to better convergence rate compared to nearest neighbor matches with the same minimization methods. In most cases, CTC gives a better estimate regardless of the minimization. Normal shooting is more competitive, winning on data sets *Middl* and the artificial *Sine*. A likely cause for this is that those datasets exhibit a very even distribution of normal vectors, which is very beneficial for normal shooting. Notable is also that CTC matching always leads to convergence, whereas the others sometimes get stuck on local minima. While achieving the global minimum is not a guaranteed property, it does appear to happen with a higher certainty than the competition.

The effect of the parameter Δr is demonstrated in Figure 3. SVD based minimization solves the translation component quickly. As expected, by increasing the threshold, the convergence of the method starts approaching the simple Nearest Neighbor search. With a tight threshold, near-instant convergence is achieved. When noise is added (Figure 4), parameters tighter than the amount of added noise become useless.

However, by adjusting the parameter Δr such that it corresponds to the noise level, the benefits of the method are retained. Empirically it was found that in the case of Gaussian noise and unit magnitude data, a suitable relation between the noise level and threshold is $\Delta r \approx \frac{1}{10}\sigma$. Furthermore, CTC still exhibits robust behavior (Figure 5.) when random coordinates of the clouds are replaced with random values. This is due to the outliers affecting a relatively small percentage of the candidate matches. The same cannot be said for pure nearest neighbor, which gets distracted by the one-to-many mappings any individual outlier can create. Normal shooting, on the other hand, gets completely thrown off by the effect of these outliers on the estimated surfaces.

In contrast, point-to-plane minimization (Figure 6) takes a different approach to the translation component. This renders setting a very tight threshold useless. As the minimization procedure creates some fluctuation in the translation while heading towards convergence, an exact match along the circular trajectory is not in all cases valid and leads to erratic behavior. However, with a suitable choice of Δr (in this case, 0.01 when cloud data is normalized to $[-1, 1]$), an improvement can still be achieved.

5. CONCLUSIONS

A method for determining corresponding points in Iterative Closest Point algorithms is proposed, together with an efficient implementation strategy. Based on the concept of tracing the trajectories of points as the point cloud rotates, it reveals significant information of the rotational component of the misalignment. Circular Trajectory Correspondences has the same computational complexity as earlier methods based on nearest neighbor matching, whose convergence rate it greatly surpasses. In terms of convergence rate in number of iterations, it has a noticeable gain against normal shooting, but is much less complicated to both implement and to compute, thus yielding comparable results with less computation. Unlike normal shooting, it lacks the need for constructing explicit surfaces and finding intersections between surface elements.

CTC is directly applicable to most existing ICP implementations as an interchangeable component to e.g. nearest neighbor or normal shooting. The functionality is demonstrated with point-to-point (SVD) and point-to-plane minimization, but it is expected that it will work with any minimization strategy. In terms of practical use of the method, a data structure based on sorting and binary searches is both effective and easy to implement by utilizing commodity algorithmic tools. Further refinements of the method can be foreseen by designing adaptive thresholding strategies and interleaving CTC with other methods to improve the robustness of recovering the translation component in more challenging use cases.

REFERENCES

[1] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL Tentative,* 1992, pp. 586-606.

[2] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on,* 2001, pp. 145-152.

[3] C. Yang and G. Medioni, "Object modelling by registration of multiple range images," *Image Vision Comput.,* vol. 10, pp. 145-155, 1992.

[4] D. Chetverikov, D. Svirko, D. Stepanov and P. Krsek, "The trimmed iterative closest point algorithm," in Pattern Recognition, 2002. Proceedings. 16th International Conference on, 2002, pp. 545-548.

[5] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans," *Journal of Intelligent and Robotic Systems,* vol. 18, pp. 249-275, 1997.

[6] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on,* 2007, pp. 1-8.

[7] Stanford University Computer Graphics Laboratory (2011, Sept. 6). *The Stanford 3D Scanning Repository*. [Online] Available: http://graphics.stanford.edu/data/3Dscanrep/