FAST ICP ALGORITHMS FOR THE REGISTRATION OF 3D DATA

ALGORITHMES ICP RAPIDES DE RECALAGE DE DONNEES 3D

Abstract

The iterative closest point (ICP) algorithm is widely used for the registration of geometric data and it applies to a wide field of activities that range from 3D object modeling to object recognition. One of the its main drawbacks is its quadratic time complexity $O(N^2)$ with the shape size N, which implies heavy computations. Consequently, there is a need to speed up the ICP algorithm and several methods have been proposed. The most effective ones focus on reducing the closest point computation time and complexity like the k-D tree search or projection methods. This paper proposes a review of the existing fast ICP methods and places emphasis on a recently proposed solution that combines the neighbor search algorithm with a multiresolution scheme to create a very fast and robust ICP. Confirming the success of the latter, the results show that it is possible to gain speed up to a factor 1600 over the standard, non-accelerated ICP algorithm, while avoiding the tradeoff with matching quality that is imposed by many existing solutions.

Timothée Jost and Heinz Hügli

Pattern Recognition Group, Institute of Microtechnology,
University of Neuchâtel,
Breguet 2, CH-2000 Neuchâtel, Switzerland.

tel: +41 32 7183459, fax: +41 32 7183402

e-mail: timothee.jost@unine.ch, heinz.hugli@unine.ch

Fast ICP algorithms for the registration of 3d data

1. Introduction

Shape registration consists in finding the correct alignment of two or more sets of data and plays an important role in today's computer vision. Advanced techniques for 3D object modeling often rely on registration to match the set of 3D views that is needed to cover the whole surface and create a complete model. Quality inspection and 3D object recognition are other examples of applications of the registration.

In 1992, Besl introduced the iterative closest point (ICP) algorithm [BES], which is one of the best-known and widely used geometric matching algorithms. Starting from an initial rough alignment of the data, the ICP processes iteratively. At each iteration, it first creates closest point correspondences between two sets of points (or more generally geometric data) and then minimizes the average distance of the previously found correspondences by a rigid transformation - a translation and a rotation.

The ICP algorithm has a complexity of $O(N_p N_x)$, where N_p and N_x basically represents the number of points of the datasets. Matching high-resolution data generally requires heavy computations and, consequently, there is a real need for ways to reduce ICP computation time. Several solutions to speed up the algorithm have been proposed and a review of the different techniques is presented in the next chapter. The main trouble encountered is that existing solutions often create a tradeoff between the speeding up and the quality of the registration.

A new solution to accelerate the ICP is presented in this paper. In fact, it consists in the combination of two recently proposed methods to speed up the ICP. First of all, the neighbor search algorithm [JOS], which relies on neighborhood relationships in the data to restrict the search of the closest point to a local subset. Then, a multi-resolution scheme [JOSb] that proceeds from coarse to fine and successively improves a previous solution at the finer representation level. This solution for the speeding up of the ICP has been developed in a perspective to avoid the tradeoff with the registration quality that was mentioned above.

This document is organized as follows. The next section presents the basic ICP algorithm and its principal variants and reviews the associated acceleration methods. Section 3 briefly describes the key points of the neighbor search algorithm and the multiresolution scheme ICP. Typical results obtained with the proposed algorithm are presented in section 4. Finally, conclusions can be found in section 5.

2. Fast registration with ICP

2.1. Basic ICP algorithm and its variants

The ICP algorithm registers two sets of points, P and X composed respectively of N_p and N_x points, starting from an initial pose estimation. The algorithm proceeds iteratively. It first pairs every point of P with its closest point of X. These pairs are used to compute the rigid transformation (**R**, **t**) which, when applied to P, minimizes the coupling error *e* of the two data sets. The resulting transformation is then applied to set P and the iteration continues until a defined stopping criterion is fulfilled.

Chen and Medioni [CHE] proposed a similar algorithm where couplings are made between points of one surface and plans parallel to the other surface. Several authors proposed to weight the point couplings [ZHA] [TUR] in order to make the ICP algorithm robust to outliers that typically appear when registering shapes of different sizes and when data sets overlap each other only partially.

Others authors also suggested to use additional features, such as surface normals [BRE] or surface colors [SCH], to define point closeness or distance, in order to improve the quality of the registration. The ensuing ICP algorithm then requires less iterations for converging to a better position and it also improves the range of successful initial configurations [SCH].

Given $d(\mathbf{p}_k, \mathbf{x})$ the function that measures the distance between a pair of points from the two sets, $\mathbf{p}_k \in P$, $\mathbf{x} \in X$, one iteration of the algorithm can be summarized as follows:

1) Compute closest points: $\forall \mathbf{p}_k \in P$, $\mathbf{x}_k = \{\mathbf{x} \in X \mid d(\mathbf{p}_k, \mathbf{x}) = \min\}$

2) Weight the couplings: define a w_k for each couple

3) Compute the best transformation, i.e. the rotation **R** and translation **t** that minimize

$$e(\mathbf{R},\mathbf{t}) = \frac{1}{W}\sum_{N_p} w_k \left\| \mathbf{R}\mathbf{p}_\mathbf{k} + \mathbf{t} - \mathbf{x}_\mathbf{k} \right\|^2, \; W = \sum_{N_p} w_k$$

4) Apply transformation (**R**, **t**) to P

These steps are iterated and the algorithm stops when a defined criterion is reached, for example when the change in the coupling error e_i falls below a threshold: $e_{i-1} - e_i < \tau$ or when the resulting best transformation is closer to identity than a threshold.

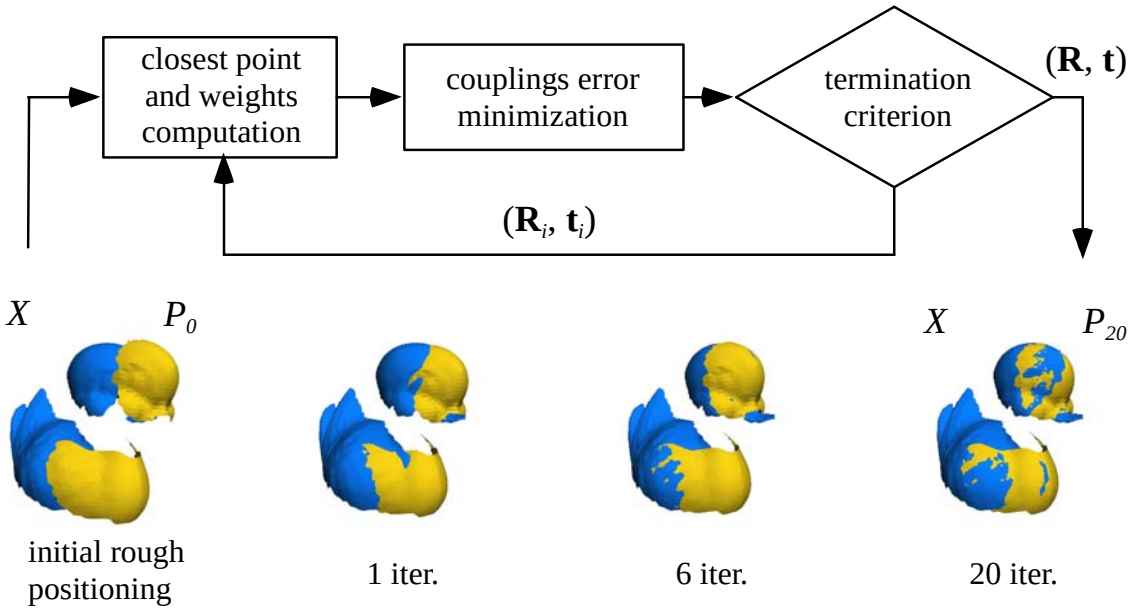

Figure 1. ICP algorithm principle

2.2. Fast ICP algorithms

The first step of the algorithm, closest point computation, has a complexity of $O(N_pN_x)$, while steps 2 to 4 possess a complexity of $O(N_p)$. Consequently, the complexity of the ICP algorithm is $O(N_pN_x)$ and, for large data sets, most of the time is spent for closest point computation.

Several authors have proposed solutions to hasten the algorithm. Langis [LAN] recently presented a parallel implementation of the ICP and showed that a nearly linear performance improvement with the number of processors can be obtained with up to 16 processors. Beside this hardware-oriented solution, one can separate the different methods into three main classes: reduction of the number of iteration, reduction of the number of data points and acceleration of the closest points search. A review of the different methods and their results are given in the next paragraphs.

Reducing the number of iteration n

In his original publication, Besl [BES] proposed a variation named *"accelerated ICP"*. It uses a linear or quadratic extrapolation of the registration parameters to reduce the number of iterations. Simon [SIM] later proposed to *decouple rotation and translation* in the accelerated ICP to reduce the number of iterations further more. Typical results from these authors showed *"accelerated ICP"* and *"decoupled"* version to reduce computation time by a factor of 3 and 4.5 respectively.
Rusinkiewicz [RUS] recently proposed a review of the influence of many variants of the ICP (as presented in 2.1) on the number of iterations. Most of them were shown to only have a marginal influence on it.

Reducing the number of data points N

Another way to reduce computation time is to reduce the number of points involved in the computation of closest points and best transformation. Some authors proposed to use a *coarse to fine strategy*. They execute the first iterations using a lower resolution, like 1/4 or 1/5 of the points, and finish with fine matching using full resolution [ZHA][TUR]. In this case, the acceleration is greatly dependent on the number of iterations performed at the different resolutions. So far, few results have been published concerning multi resolution strategy. Zhang [ZHA] found a reduction factor of about 2 to 4.
Chen and Medioni [CHE] proposed to use only subsets of the data named *control points*. As such, they suggested using points sitting in smooth areas, because normals and line plan intersections are more reliable in that case. This argument is valid when using point to plan distances but is of less importance for the ICP algorithm where point to point distances are calculated. Brett [BRE] applies an *alternate mesh reduction* algorithm to triangulated surfaces that keeps significant features (high curvature) and iteratively matches a reduced P mesh with X then a reduced X mesh with P. More ways of choosing control points exist, such as random choice or even distribution of the normals [RUS].

Speeding up the closest point computation

The acceleration of the closest point search can be done using either search structures or projection methods. Search structures, like the *k-D tree* [ZHA] or *the spherical triangle constraint nearest neighbor (STCNN)* [GRE], permit to accelerate the search by restricting it to a subpart of the data. This allows to reduce the complexity of the closest point search – and of the ICP - to $O(N_p \log N_x)$ with a k-D tree and up to $O(N_p)$ with the STCNN.
The goal of projection methods is to speed up the closest point search by projecting points into one or more planes, reducing the problem to a 2D search. If scanner parameters are known, the *reverse calibration* [BLA] consists in projecting the points of one dataset into the range image of the second one, in the direction of the range camera. Projection in *multiple Z-buffers* [BENb] is another solution. Both these methods permits to reduce the complexity up to $O(N_p)$.

2.3. Discussion

An in-depth review and comparison of the different methods can be found in [JOSc] but here are the key points that we retained from it.
Using the extrapolation of parameters, a reduction of the computation time by a factor of 3 or 4 can be expected but at the risk of overshoot. The latter could at best eliminate the beneficial effect of the method but it could also cause the algorithm to converge toward a bad local minimum, which would be annoying.

Using control points imply a reduction of the computation time linked with the number of control points. The less control points used, the better the acceleration of the ICP. On the other hand, less control points also means a bigger registration error. In this perspective, a coarse to fine approach would be preferred. We should also remember that these solutions don't change the overall cost of the ICP algorithm, which remains quadratic and, thus, is still subject to long computation time for large data sets.
Search structure and projection method allow to reduce the complexity – theoretically up to $O(N_p)$ - and consequently have the best impact on the computation time of the ICP algorithm. For example, Zhang [ZHA] obtained a time reduction factor of about 15 for meshes containing about 2500 points when using a k-D tree and it should increases with the number of points. The main problem with search structures is that they lose most of their advantage when datasets are far from each other, which is generally the case in the first iterations of the ICP, and when additional features are used to define the closest point, which is useful to make the ICP more robust.
Finally, projection methods permit a very good speeding up of the closest point search. On the other hand, projection methods only give approximations of the closest points, especially when datasets are only coarsely aligned, and are not very adapted to the use additional features. This can lead to bad matching and, consequently, the range of successful initial configurations tends to be much smaller when using these methods.
One can note here that all three types of acceleration methods are quite independent and consequently can be combined together. For example, Simon [SIM] mixed accelerated ICP with k-D tree and Zhang [ZHA] used both coarse to fine strategy and k-D tree. Our solution, presented in the next section, also combines a coarse to fine strategy with a fast closest point search.
To conclude, we can see that most of the existing solutions lead to a tradeoff between the speeding up and the quality of the registration – i.e. registration error and range of successful initial configurations [HUG].

3. Multiresolution neighbor search algorithm

This section briefly describes the key points of the neighbor search algorithm and the multiresolution scheme ICP. Both methods are combined in order to create a fast ICP algorithm that also avoids any tradeoff with the quality of the registration. In-depth presentation and analyses of both algorithms are found in [JOS][JOSb][JOSc].

3.1 Neighbor search

Recently, Jost [JOS] proposed the “neighbor search” closest point algorithm. Given that there exist neighborhoods V and V’ defined in respectively datasets *P* and *X*, it assumes that two neighbors on a surface possess closest points on the other surface that are neighbors as well. This neighborhood relationship assumption is illustrated in Figure 2. Points $\mathbf{p}_i$ and $\mathbf{p}_k$ of set *P* are within the same neighborhood $V(\mathbf{p}_i)$, so their corresponding closest points in set *X*, $\mathbf{x}_i$ and $\mathbf{x}_k$ are also found within neighborhood $V'(\mathbf{x}_k)$.
The neighbor search algorithm uses this property to obtain a first approximation of the closest point and then refines the result with a local search. In Figure 2, if $\mathbf{p}_i$ possesses a neighbor $\mathbf{p}_k$ in data set *P*, with a know closest point $\mathbf{x}_k$ in data set *X*, finding the closest point of $\mathbf{p}_i$ can be reduced to searching the closest point in the neighborhood V’ of $\mathbf{x}_k$, $V'(\mathbf{x}_k)$. This method permits to avoid a global search for most points and leads to a closest points search (and therefore ICP) algorithm that provides a complexity of $O(N_p)$.

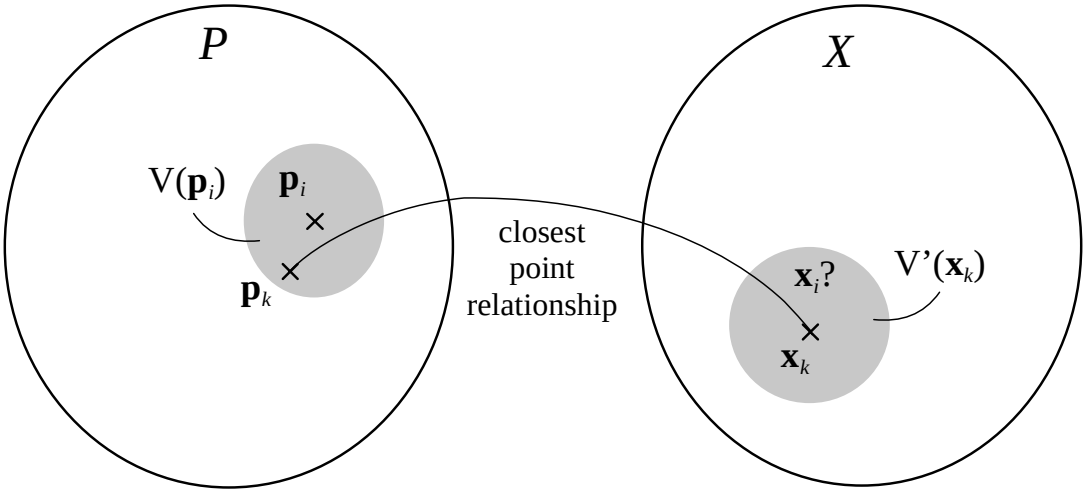

Figure 2. The neighborhood relationship assumption

Figure 3 presents the neighbor search performed in ranges images. Neighborhood V is the 3x3 window surrounding the point $\mathbf{p}_i$ in *P* (V-8 neighborhood) and neighborhood V' is a *n*x*n* window in *X*. In this case, we choose to scan the points of range image *P* row by row, starting from upper left. That way, the possible direct neighbors of $\mathbf{p}_i$ with a known closest point, $\mathbf{p}_k$, can be found on the previous point in the same row and in the previous row (see image P on Figure 3). Those 4 possible candidates are just checked sequentially and the first one that possesses a known closest point is chosen as $\mathbf{p}_k$.

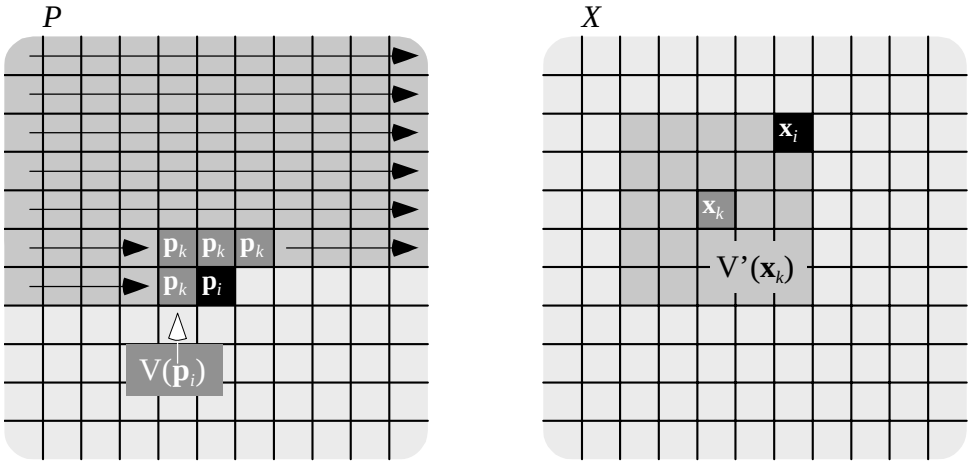

Figure 3. The neighbor search in range images

3.2 Multiresolution scheme

The principle of multiresolution ICP [JOSb] is to make the first few iterations using down sampled data and to further increase the resolution of the data in the following iterations, creating a coarse to fine matching. The main expected advantage of the multiresolution is the reduction of the computational cost, given that the duration of each iteration made at lower resolutions is reduced. In addition, it is expected this way that the total number of iterations will be reduced, mainly because a lower resolution matching generally implies more important rotations and translations, meaning a faster convergence.

The multiresolution pattern chosen here is to divide the number of points by a factor N for each resolution step. The lowest possible resolution is defined by keeping the number of points of the reduced data sets above a minimum value (typically 50 or 100).
The number of iterations at each resolution step isn't set. Instead, the algorithm goes to the next resolution step automatically when a defined stop criterion is reached at the current one.
Figure 4 shows a typical dataset at different resolutions steps.

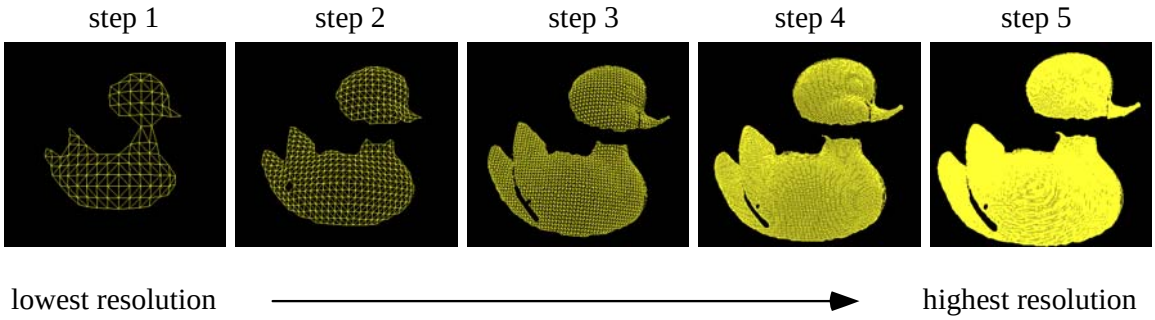

Figure 4. Duck toy data at the different resolution steps

4. Experimental results

The presented fast ICP algorithm has been tested on different data and compared with other fast ICP algorithms using tree search and neighbor search alone. The comparison focuses on two features: computation speed and matching quality. The expected goal is, as said before, to speed up the algorithm while keeping the same quality of the matching.
The following results have been obtained using surfaces of a duck toy measured with a structured light range finder, as seen in Figure 1.

Computation speed

Table 1 presents a comparison of the average total computation time for the successful registrations, using the different acceleration methods. One can see that the multiresolution and neighbor search combination is more than 25 times faster than a tree search fast ICP. This result is also expected to become higher for bigger data sets, due to the smaller complexity of the neighbor search algorithm. Finally, a look on the approximate gain in speed over a non-accelerated ICP algorithm shows that the presented fast ICP is over 1500 faster.

	total time (s)	**number of iterations**	**relative gain over k-D tree**	**absolute gain in speed**
k-D tree	504.1	42	1.0	60
n-search	259.0	90	2.0	117
MR k-D tree	59.3	36	8.5	513
MR n-search	18.8	34	**26.8**	**1621**

Table 1. Comparison of the average computation time and gains of the registration using the different acceleration methods (n-search = neighbor search, MR = multiresolution)

Matching quality

Two measures can be considered to examine the quality of the matching procedure: the matching error and the domain of convergence. To compare the matching error, the

resulting positioning of the successful registration has to be the same or at least in the same error range as when matching using exact closest points. It was the case with all the methods used in this chapter.
To examine the domain of convergence, we used a method presented in [HUG] that compares the domains of successful initial configurations (SIC). We won't enter into details but, basically, both datasets are registered starting from a lot of different initial rough positioning. Each registration is then compared to the correct known positioning and the results are plotted in a SIC-map, where a black sector represents a successful registration. Consequently, the more black sectors in a SIC-map, the bigger the SIC range (or domain) is.
The neighbor search algorithm, like the projection methods, uses a heuristic closest point search to improve matching speed. Experimental results [JOSc] showed that the SIC range was getting smaller when higher resolution data were used unless the local search window was made bigger as well. This effect can be seen in Figure 5: the SIC range when using the neighbor search is much smaller than in the non-accelerated case. This is something we want to avoid.

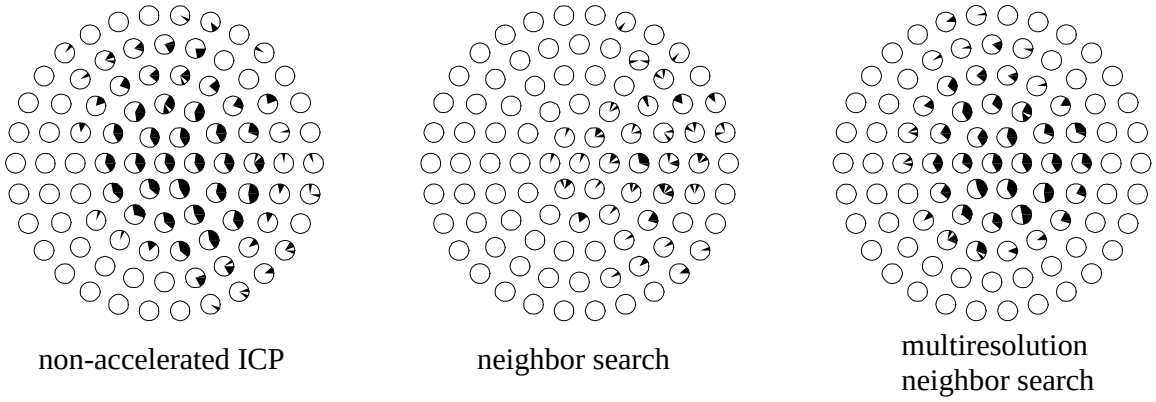

Figure 5. . SIC-maps

On the other hand, the results confirm that the multiresolution scheme does not affect the matching quality negatively and that it has beneficial effects when combined with the neighbor search algorithm, since the SIC range is again similar to the non-accelerated case.

5. Conclusion

This paper proposed a review of the ICP algorithm and the principal methods to speed it up. It also placed emphasis on a recently proposed solution that combines the neighbor search algorithm with a multiresolution scheme to create a very fast and robust ICP algorithm.
Reviewing the main solutions for the speeding up of the ICP algorithm, it was shown that most of them imply a tradeoff between the gained acceleration and the quality of the resulting matching. More precisely, many solutions either reduce the range of successful initial configurations – a better initial positioning of the data is then needed - or enlarge the registration error.
The presented ICP algorithm combines the use of the neighbor search algorithm with a multiresolution scheme. The neighbor search uses the assumption that two neighbors on a surface possess closest points that are neighbors on the other surface to easily obtain a first approximation of the closest point and then proceeds with a local search to refine the result. The chosen multiresolution scheme proceeds from coarse to fine and successively

improves a previous solution at the finer representation level. This combination allows obtaining a very fast and robust registration of two datasets.
Experimental results showed that when combining multiresolution with the neighbor search method, the registration is up to around 25 times faster than when using a tree search, which represents a gain of more than 1600 over a non-accelerated ICP algorithm. Moreover, the multiresolution scheme permits to maintain the range of successful initial configurations, as well as the registration error, unchanged.

References

[BEN] JL. Bentley (1975), "Multidimensional Binary Search Tree Used for Associative Searching", *Communications of the ACM*, vol. 8(9), pages 509-517.

[BENb] R. Benjemaa and F. Schmitt, "Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique," *Proceedings of the International Conference on Recent Advances in 3D Imaging and Modeling (3DIM97)*, Ottawa, pages 113-119, 1997.

[BES] PJ. Besl, ND. McKay (1992), "A Method for Registration of 3D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14(2), pages 239-256.

[BLA] G. Blais and M. Levine, "Registering multiview range data to create 3D computer objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 17, no. 8, pp. 820-824, 1995.

[BRE] AD. Brett, A. Hills, CJ. Taylor (1999), "A Method of 3D Surface Correspondence and Interpolation for Merging Shape", *Image and Vision Computing*,, vol. 17(8), pages 635-642.

[CHE] Y. Chen, G. Medioni (1992), "Object modeling by registration of multiple range images", *International Journal of Image and Vision Computing*, vol. 10(3), pages 145-155.

[GRE] M. Greenspan, G. Godin, "A nearest neighbor method for efficient ICP," *Proceedings of Int. Conf. on Recent Advances in 3-D Digital Imaging and Modelling (3DIM01)*, Quebec, pp. 161-168, 2001.

[HUG] H. Hügli, C. Schütz (1997), "Geometric Matching of 3D Objects: Assessing the Range of Successful Initial Configurations", *Proceedings of International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 101-106, Ottawa.

[JOS] T. Jost, H. Hügli, "Fast ICP algorithms for shape registration," *Proceedings of DAGM02*, Zürich, pages 91-99, 2002.

[JOSb] T. Jost, H. Hügli, "A multi-resolution scheme ICP algorithm for fast shape registration," *Proceedings of the 1st Int. Symp. on 3D Data Processing, Visualisation and Transmission (3DPVT02)*, Padova, pages 540-543, 2002.

[JOSc] T. Jost, "Fast Geometric Matching for Surface Registration," PhD thesis, University of Neuchatel, Switzerland, 2003.

[LAN] C. Langis, M. Greenspan, G. Godin, "The parallel Iterative closest point algorithm," *Proceedings of Int. Conf. on Recent Advances in 3-D Digital Imaging and Modelling (3DIM01)*, Quebec, pp. 195-204, 2001.

[RUS] S. Rusinkiewicz, M. Levoy, "Efficient Variants of the ICP Algorithm," *Proceedings of Int. Conf. on Recent Advances in 3-D Digital Imaging and Modelling (3DIM01)*, Quebec, pages 145-152, 2001.

[SCH] C. Schütz, T. Jost, H. Hügli (1998), "Multi-Featured Matching Algorithm for Free-Form 3D Surface Registration", *Proceedings of International Conference on Pattern Recognition*, pages 982-984, Brisbane.

[SIM] DA. Simon, M. Hebert, T. Kanade (1995), "Techniques for Fast and Accurate Intra Surgical Registration", *Journal of Image Guided Surgery*, vol. 1(1), pages 17-29.

[TUR] G. Turk, M. Levoy (1994), "Zippered Polygon Meshes from Range Images", *Proceedings of ACM Siggraph*, pages 311-318, Orlando.

[ZHA] Z. Zhang, (1994), "Iterative Points Matching for Registration of Free Form Curves and Surfaces", *International Journal of Computer Vision*, vol. 13(2), pages 119-152.