

A Multi-Resolution ICP with Heuristic Closest Point Search for Fast and Robust 3D Registration of Range Images

Timothée Jost and Heinz Hügli

*Institute of Microtechnology, University of Neuchâtel,
Breguet 2, CH-2000 Neuchâtel, Switzerland.*

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

Abstract

The iterative closest point (ICP) algorithm is widely used for the registration of 3D geometric data. One of the main drawbacks of the algorithm is its quadratic time complexity $O(N^2)$ with the number of points N . Consequently, several methods have been proposed to accelerate the process. This paper presents a new solution for the speeding up of the ICP algorithm and special care is taken to avoid any tradeoff with the quality of the registration. The proposed solution combines a coarse to fine multiresolution approach with the neighbor search algorithm. The multiresolution approach permits to successively improve the registration using finer levels of representation and the neighbor search algorithm speeds up the closest point search by using a heuristic approach. Both multiresolution scheme and neighbor search algorithm main features are presented in this paper. Confirming the success of the proposed solution, typical results show that this combination permits to create a very fast ICP algorithm, with a closest point search complexity of $O(N)$, while preserving the matching quality.

1. Introduction

The geometric matching of 3D datasets consists in finding their correct relative alignment based on their intrinsic properties. Typical applications using registration as part of their working principle include the modeling of 3D objects, object recognition or quality inspection.

The iterative closest point (ICP) algorithm [2] figures among the principal and widely used low-level registration methods. 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 main practical difficulty of the ICP algorithm is that it requires heavy computations. When working with clouds of points or triangulated meshes, the complexity of the original algorithm is $O(N_p N_x)$, where N_p and N_x represent the number of points of the clouds to be matched. Consequently, matching high-resolution shapes can take a lot of time, even on current computers, and there is a need for ways to reduce the ICP computation time.

A review of the main solutions to speed up the ICP 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 - as measured i.e. by an increased registration error and/or a reduced range of successful initial configurations [8].

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 [9], 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 [10] 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 and 4 briefly describe the key points of the neighbor search algorithm and the multiresolution scheme ICP. An experimental comparison of the proposed algorithm with existing methods is presented in section 5. Finally, conclusions can be found in section 6.

2. Fast registration with ICP

The ICP algorithm registers two sets of points, P and X composed of respectively 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 (\mathbf{R}, \mathbf{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.

Different variants of the ICP algorithm exist that improve the matching quality and the robustness of the registration. Chen and Medioni [5] proposed a similar algorithm where couplings are made between points of one surface and planes parallel to the other surface. Several authors proposed to weight the point couplings [16] [17] in order to make the ICP algorithm more robust to the outliers that typically appear when registering shapes of different sizes or when data sets overlap each other only partially.

Others authors also suggested to use additional features, such as surface normals [4] or surface colors [14], 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 [14]. A review of the different variants can be found in [13].

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:

$$\forall \mathbf{p}_k \in P, \mathbf{x}_k = \{x \in X \mid d(\mathbf{p}_k, \mathbf{x}) = \min\}$$

2. Weight the couplings: define a w_k for each couple
3. Compute the best transformation, i.e. the rotation \mathbf{R} and translation \mathbf{t} that minimize

$$e(\mathbf{R}, \mathbf{t}) = \frac{1}{W} \sum_{N_p} w_k \|\mathbf{R}\mathbf{p}_k + \mathbf{t} - \mathbf{x}_k\|^2, \quad W = \sum_{N_p} w_k$$

4. Apply transformation (\mathbf{R}, \mathbf{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 falls below a threshold: $e_{i-1} - e_i < \tau$ or when the resulting best transformation is closer to identity than a threshold.

2.1. Fast ICP algorithms

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

Many solutions for the acceleration of the ICP algorithm have been proposed. Langis [12] recently proposed 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 iterations n , reduction of the number of data points N_p and N_x and acceleration of the closest points computation. A review of the different methods and their results are given in the next paragraphs.

2.1.1. Reducing the number of iteration n . In his original publication, Besl [2] proposed a variation named "*accelerated ICP*". It uses a linear or quadratic extrapolation of the registration parameters to reduce the number of iterations. Simon [15] 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 [13] recently proposed a review of the influence of many variants of the ICP (as presented above) on the number of iterations. Most of them were shown to only have a marginal influence on it.

2.1.2. Reducing the number of data points N_p and N_x . 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 [16][17]. 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 [17] found a reduction factor of about 2 to 4.

Chen and Medioni [5] 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. Other ways of choosing control points have also been proposed. Brett [4] 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 . A pure random choice, an even distribution of the normals [13] or a generalized feature-oriented random sampling [6] are among other possible choice schemes.

2.1.3. 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* [17] or the *spherical triangle constraint nearest neighbor (STCNN)* [7], 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 a best case $O(N_p)$ 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* [3] 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* [1] is another solution. Both these methods permits to reduce the complexity up to $O(N_p)$.

2.2. Discussion

An in-depth review and comparison of the different methods can be found in [11] but here are the main 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, the method chosen to select the points can have a negative impact on the computation time of the registration. Less control points also means a bigger registration error and, in this perspective, a coarse to fine approach would be preferred.

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 [17] obtained a time reduction factor of about 15 for meshes containing about 2500 points when using a *k-D tree* and it should increase with the number of points. The main problem with search structures is that they lose a lot of their speeding up advantages 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, they only give approximations of the closest points, especially when datasets are only coarsely aligned, and are not very adapted to the use of 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 [15] mixed accelerated ICP with *k-D tree* and Zhang [17] 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.

As our main conclusion, the analysis of the literature shows that most of the existing solutions lead to a tradeoff between the actual speeding up and the quality of the registration – i.e. registration error and range of successful initial configurations [8].

3. The neighbor search algorithm

The neighbor search algorithm [9] assumes the existence of a neighborhood relationship between the two sets of points P and X . Given that there exist neighborhoods V and V' defined in respectively datasets P and X , the relationship hypothesis is that two neighbors in a data set possess closest points that are neighbors in the other data set. Formally, the principle of this neighborhood relationship is exposed in Figure 1: given a point \mathbf{p}_k in data set P and its corresponding closest point \mathbf{x}_k in data set X , the closest point \mathbf{x}_i of \mathbf{p}_i , if \mathbf{p}_k belongs to neighborhood V of \mathbf{p}_i , $V(\mathbf{p}_i)$, is found in the neighborhood V' of \mathbf{x}_k , $V'(\mathbf{x}_k)$.

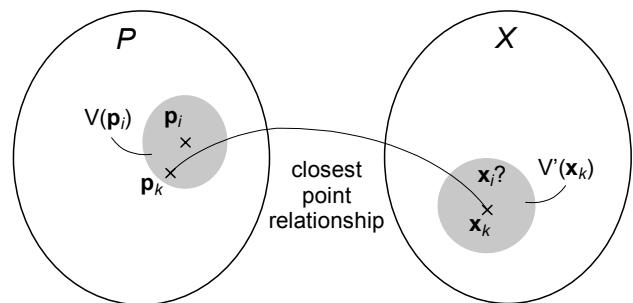


Figure 1. The neighborhood relationship assumption

The proposed idea towards a faster search is to use good approximations of the closest points instead of exact closest points. The neighborhood relationship is used to get a first approximation of the closest point and, then, a local search can be performed to refine the result instead of an exhaustive (or global) one: if \mathbf{p}_i possesses a neighbor \mathbf{p}_k in data set P , with a known 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 , $V'(\mathbf{x}_k)$.

The following pseudo-code formulates the closest point neighbor search algorithm:

input: datasets P and X , with associated neighborhood, $V(P)$ and $V'(X)$

output: for each p_i of P , an approximation x_i of its closest point in X

procedure neighbor_closest_point_search (P, X)

```

for (each  $p_i$  of  $P$ ) do
  if ( $\exists x_k$  closest point of  $p_k \in V(p_i)$ ) then
     $x_i = \text{local\_closest\_point}(p_i, V'(x_k))$ 
  else
     $x_i = \text{global\_closest\_point}(p_i, X)$ 

```

It appears that for each point p_i , the closest point search is performed either in the full set X or only in the neighborhood $V'(x_k)$, depending whether at least one neighbor of p_i has already a known closest point or not. Formally, this closest point search algorithm has therefore a theoretical best-case complexity of $O(N_p)$, if a local search can be performed in each case, and a worst case complexity equal to the one of the global search. One can note that a basic fast search method, such as a k-D tree, can be used for the global search.

Of course, the order in which points p_i of P are scanned is important. Using a random method is a bad idea, as it would create a high number of global searches and push complexity toward the worst case. Consequently, the basic idea is to scan points using a diffusion method, so that the next point p_i is chosen in the neighborhood of the points that already have a known neighbor.

3.1. Algorithm applied to range images

When considering range images, each point possesses either 4 or 8 direct neighbours (except points on borders), depending on the considered topology ($V-4$ or $V-8$ neighbourhood). A very basic but effective algorithm is considered here. Neighbourhood V is the 3×3 window surrounding the point p_i in P ($V-8$ neighbourhood) and neighbourhood V' is a $n \times n$ window in X . We choose to scan the points of range image P row by row, starting from upper left. That way, the possible direct neighbours of p_i with a known closest point p_k can be found on the previous point in the same row and in the previous row (see image P on Figure 2). Those 4 possible candidates are just checked sequentially and the first one that possesses a known closest point is chosen as p_k .

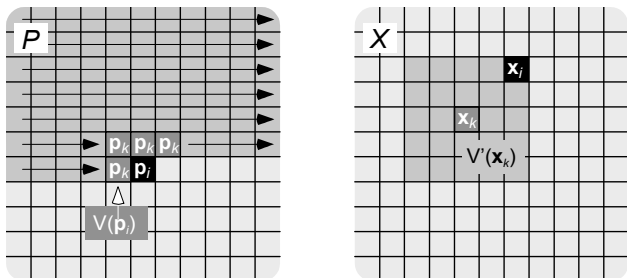


Figure 2. The neighbor search in range images

Normally, any of the candidate neighbors possesses a known closest point, except for the first scanned point and in case of missing data points.

Once p_k and its corresponding closest point x_k are known, the local closest point search of p_i is done in a square neighborhood zone of size $n \times n$, centered on the approximation x_k (see image X in Figure 2). If no point p_k can be found, the global search is performed in X using a k-D tree search, as suggested previously.

4. Multiresolution scheme

The principle of the multiresolution ICP 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. The precision of the final matching is expected to be the same as when using all the points for the whole registration. In addition, we expect the total number of iterations not to be higher than in the “monoresolution” case.

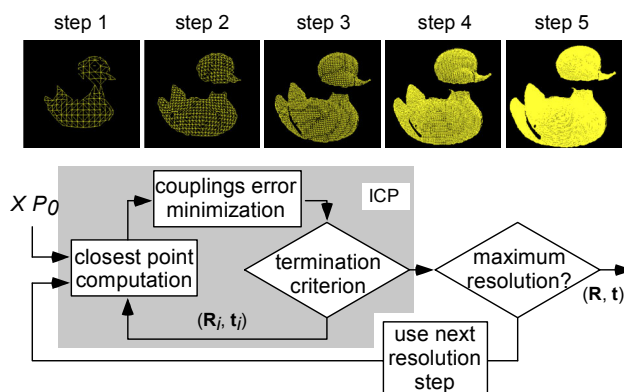


Figure 3. The multiresolution ICP principle

4.1. Chosen multiresolution pattern

The multiresolution coarse to fine strategy is not a new concept and has been widely used in image processing and other domains for years. A few publications (as seen in the introduction) also briefly presented some coarse to fine solutions applied to ICP.

The pattern we chose has been presented in [10]. It consists in dividing 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 3). A complete analysis of the presented scheme can be found in [11].

4.2. Multiresolution neighbor search algorithm

Both multiresolution scheme and neighbor search are quite independent and consequently can be combined together. Such a combination should permit to create a very fast ICP algorithm and the following remarks can be done.

In the case of the neighbor search, the reduction of data set X doesn't have a big influence on the cost given that it has a best-case complexity of $O(N_p)$. However we can note that in practice, data set X needs to be reduced as well in order that the relative size of the local search area stays the same.

A complete analysis of the proposed multiresolution scheme can be found in [11]. To estimate the numerical value of the gain in speed $G(N_p)$ of a multiresolution case in relation to a full resolution case, we made the hypothesis that the total number of iterations remains the same in both cases. We also distinguished 3 cases on how iterations are distributed for each step:

1. a constant number of iterations at each step
2. an increasing number of iterations with the higher resolutions
3. a decreasing number of iterations with the higher resolutions.

These different cases can be seen as average (1), worst (2) and best (3) theoretical cases. Figure 4 presents a graphical representation of these estimates as a function of N_p . One can see that the bigger the datasets, the higher the expected gain is.

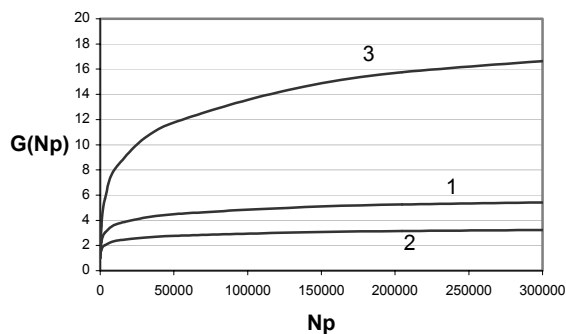


Figure 4. Expected multiresolution gains $G(N_p)$ for fast ICP algorithms, for $N=4$

Besides its pure speeding up potential linked to the reduction of the number of points, the multiresolution matching scheme is expected to have an added beneficial impact on the neighbor search method. When using the neighbor search algorithm, higher resolutions tend to decrease the exactness of the closest points pairing,

especially on coarse misalignment of the datasets, which typically reduces the range of successful initial configurations (SIC). This effect can be seen in Figure 5 in the next section. Using a lower resolution for the initial iterations, when the misalignment is coarse, should permit to avoid this problem since the registration of lower resolution datasets possess the best SIC range.

5. 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: matching quality and computation speed. The expected goal is, as said before, to speed up the algorithm while keeping the same quality of the matching.

The following typical results have been obtained using overlapping surfaces of a duck toy measured with a structured light range finder. The overlap is approximately 35% of the surfaces and the datasets contain about 25000 to 30000 points. Beside the modifications presented above, our ICP algorithm uses the normals in the distance computation [14], 2 fixed thresholds depending of the resolution of the data to get rid of outliers and a balanced k-D tree search similar to the one proposed by Zhang [17].

5.1. 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 [8] that compares the domains of successful initial configurations (SIC) and present the results in SIC-maps. Basically, the SIC-map represents by black sectors the range of successful initial configurations of two datasets to be matched relative to a three-dimensional angular misalignment (φ, θ, ω space) of the datasets. Consequently, the more black sectors in a SIC-map, the bigger the SIC range (or domain) is.

As mentioned previously, the neighbor search algorithm uses a heuristic closest point search to improve matching speed. This means that the exactness of the matching decreases when datasets are further apart and when their resolution is bigger. As a consequence, coarser initial alignments that previously lead to a successful convergence may not converge correctly anymore and, thus, the SIC range tends to get smaller when higher resolution data are used, as shown in the SIC-maps of

Figure 5. This is something we want to avoid thank to the multiresolution scheme.

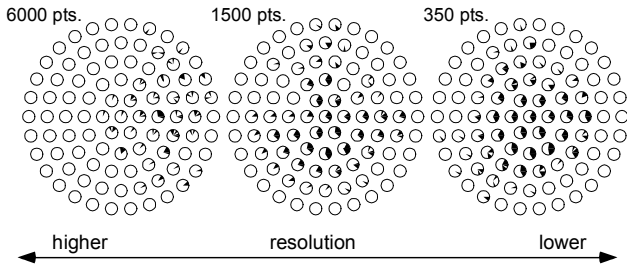


Figure 5. SIC-maps using a 9x9 neighbor search for different data resolution

Figure 6 compares the SIC-maps obtained using a k-D tree search ICP, the neighbor search accelerated ICP and their multiresolution variants. The results first show that multiresolution does not affect the domain of successful convergence when coupled with a k-D tree search, since the SIC range remains nearly the same in both cases. Furthermore, it has the expected beneficial effect on it when coupled with the neighbor search. More precisely, one can see that the multiresolution scheme permits to maintain the SIC range of the ICP when combined with the neighbor search.

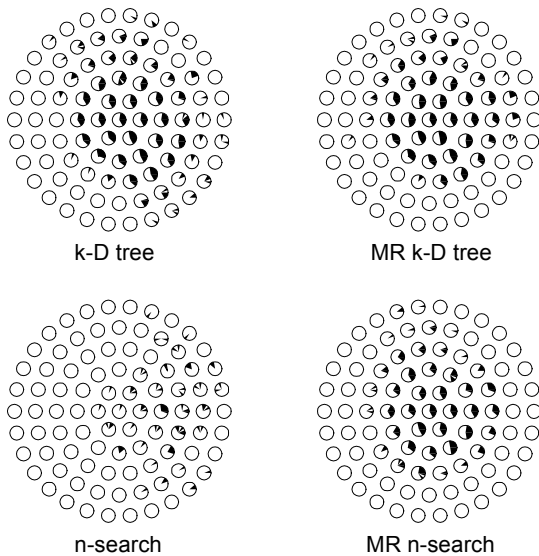


Figure 6. SIC-maps obtained using the different fast ICP methods (n-search=neighbor search, MR = multi-resolution)

All in all, these results confirm that the multiresolution scheme does not affect the matching quality for both the matching error and the domain of convergence. Furthermore, it has beneficial effects on the SIC range, especially when combined with the neighbor search algorithm.

5.2. Computation speed

Figure 7 graphically presents the average closest point computation time per point of P, t_p , as a function of N_x . Two values of k-D tree time are given at each resolution, a minimal time and a maximal time. This is to reflect the difference in the computation time depending on the distance between both data sets when using a k-D tree. Basically, the tree search is longer when data sets are farther from each other. The min / max values must be considered as a best-case / worst-case type of measure.

One can observe that the theoretical complexity $O(N_p)$ was practically reached for the neighbor search since $t_p \sim \text{const}$. For k-D tree, one can see that $t_p \sim \log N_x$, which follows the theoretical complexity $O(N_p \log N_x)$. The gain in speed lies typically between 2 and 5 over a best-case k-D tree. Considering the worst-case tree search values, the gain can go up to 13 times!

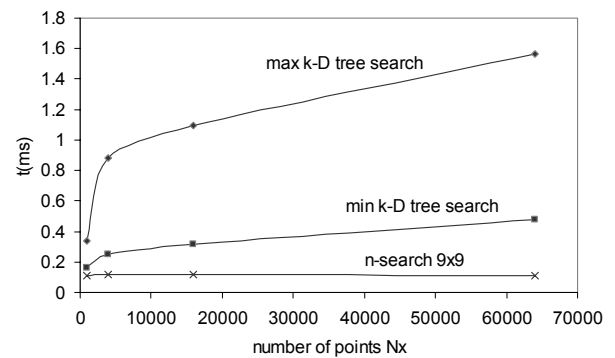


Figure 7. Average closest point computation time per point of P, $t_p(N_x)$, for the n-search and k-D tree

Figure 8 presents a comparison of the average gains in the computation time of 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 using a k-D tree search fast ICP and 1500 faster than a non-accelerated ICP.

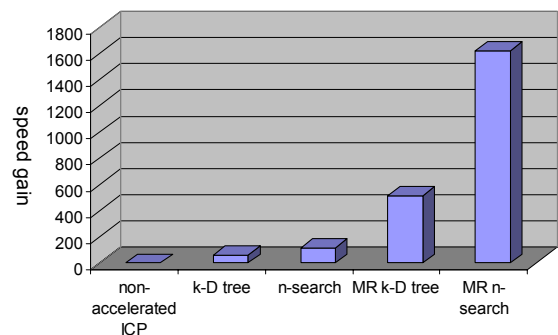


Figure 8. Comparison of the average gains in the computation time of the registration using the different acceleration methods

These results are also expected to become higher for bigger data sets, due to the smaller complexity ($O(N_p)$) of the neighbor search algorithm.

Considering the impact of the sole multiresolution scheme, it permits to reduce the total registration time by an average factor of about 9 in the presented case. Practically, the number of iterations is effectively decreasing with the higher resolutions steps, although not as much as the best theoretical case. On the other hand, the total number of iterations is slightly reduced – typically 10 to 20% less iterations –, which was not taken into account in the theoretical case [11]. All in all, the results are basically similar to the theoretical best-case gain shown in Figure 4 (case 3).

6. Conclusion

This paper proposes a solution to the speeding up of the ICP algorithm that combines a heuristic closest point search algorithm with a multiresolution scheme. A review of existing solutions showed that most of them imply a tradeoff between the gained acceleration and the quality of the resulting matching. Our goal is to obtain a fast but also robust ICP algorithm, both in terms of resulting alignment error and range of successful initial configuration.

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.

More precisely, experimental results obtained with typical datasets 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. The speedup gain is also expected to be higher for bigger data sets, due to the smaller complexity ($O(N_p)$) of the neighbor search algorithm. Moreover, the multiresolution scheme permits to maintain the range of successful initial configurations, as well as the registration error of the ICP, unchanged. This clearly shows that the proposed fast ICP algorithm exploits the fundamental nature of shape registration to substantially contribute to improve its computation.

Finally, the neighbor search principle was applied to range images in this work but the concept can be extended to three-dimensional polygonal meshes or point sets in the future.

References

- [1] 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, 1997, pp. 113-119.
- [2] P.J. Besl, ND. McKay, “A Method for Registration of 3D Shapes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14(2), 1992, pp. 239-256.
- [3] 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(8), 1995, pp. 820-824.
- [4] AD. Brett, A. Hills, CJ. Taylor, “A Method of 3D Surface Correspondence and Interpolation for Merging Shape”, *Image and Vision Computing*, vol. 17(8), 1999, pp. 635-642.
- [5] Y. Chen, G. Medioni, “Object modeling by registration of multiple range images”, *International Journal of Image and Vision Computing*, vol. 10(3), 1992, pp. 145-155.
- [6] G. Godin, D. Laurendeau, R. Bergevin, “A method for the registration of attributed range images”, *Proceedings of 3DIM01*, Quebec, 2001, pp. 179-186.
- [7] M. Greenspan, G. Godin, “A nearest neighbor method for efficient ICP”, *Proceedings of 3DIM01*, Quebec, 2001, pp. 161-168.
- [8] H. Hügli, C. Schütz, “Geometric Matching of 3D Objects: Assessing the Range of Successful Initial Configurations”, *Proceedings of 3DIM97*, Ottawa, 1997, pp. 101-106.
- [9] T. Jost, H. Hügli, “Fast ICP algorithms for shape registration”, *Proceedings of DAGM02*, Zürich, 2002, pp. 91-99.
- [10] T. Jost, H. Hügli, “A multi-resolution scheme ICP algorithm for fast shape registration”, *Proceedings of the 1st Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT02)*, Padova, 2002, pp. 540-543.
- [11] T. Jost, *Fast Geometric Matching for Surface Registration*, PhD thesis, University of Neuchatel, Switzerland, 2002.
- [12] C. Langis, M. Greenspan, G. Godin, “The parallel Iterative closest point algorithm”, *Proceedings of 3DIM01*, Quebec, 2001, pp. 195-204.
- [13] S. Rusinkiewicz, M. Levoy, “Efficient Variants of the ICP Algorithm”, *Proceedings of 3DIM01*, Quebec, 2001, pp. 145-152.
- [14] C. Schütz, T. Jost, H. Hügli, “Multi-Featured Matching Algorithm for Free-Form 3D Surface Registration”, *Proceedings of International Conference on Pattern Recognition*, Brisbane, 1998, pp. 982-984.
- [15] DA. Simon, M. Hebert, T. Kanade, “Techniques for Fast and Accurate Intra Surgical Registration”, *Journal of Image Guided Surgery*, vol. 1(1), 1995, pp. 17-29.
- [16] G. Turk, M. Levoy, “Zippered Polygon Meshes from Range Images”, *Proceedings of ACM Siggraph*, Orlando, 1994, pp. 311-318.
- [17] Z. Zhang, “Iterative Points Matching for Registration of Free Form Curves and Surfaces”, *International Journal of Computer Vision*, vol. 13(2), 1994, pp. 119-152.