A Multi-Resolution Scheme ICP Algorithm for Fast Shape Registration

Timothée Jost and Heinz Hügli

Pattern Recognition Group, 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 geometric data. One of its main drawbacks is its quadratic time complexity $O(N^2)$ with the shapes number of points N, which implies long processing time, especially when using high resolution data. This paper proposes to accelerate the process by a coarse to fine multiresolution approach in which a solution at a coarse level is successively improved at a finer level of representation. Specifically, it investigates this multiresolution ICP approach when coupled with the tree search or the neighbor search closest point algorithms. A theoretical and practical analysis and a comparison of the considered algorithms are presented. Confirming the success of the multiresolution scheme, the results also show that this combination permits to create a very fast ICP algorithm, gaining speed up to a factor 27 over a standard fast ICP algorithm.

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. For example, the modeling of 3D object generally requires to assemble several complementary views into a single representation.

The iterative closest point (ICP) algorithm [2] is one of the best known and widely used low level geometric matching algorithms. It finds an optimal match iteratively. Each iteration proceeds by first creating closest point pairs between two sets of points (or more generally geometric data) and then by computing the rigid transformation (**R**, **t**) which minimizes the coupling error *e* of the two data sets. Several variants of the ICP have been proposed, and a recent overview and comparison can be found in [7].

The ICP algorithm has a basic complexity of $O(N_pN_x)$, where N_p and N_x represent the number of points considered in each data sets. Matching high resolution shapes requires heavy computations and several solutions to speed up the algorithm have been proposed. One can separate the different methods in three main classes: reduction of the number of iteration, reduction of the number of data points and acceleration of the closest points search.

Besl [2] and later Simon [9] proposed variations named *"accelerated ICP"* which use a linear or quadratic extrapolation of the registration parameters to reduce the number of iterations. Typical results from these authors showed reduction of the computation time by a factor of 3 and 4 respectively.

Some authors proposed to use a *coarse to fine strategy* [10, 11]. 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. Zhang [11] found a reduction factor of about 2 to 4 using this strategy. Chen and Medioni [4] and Brett [3] proposed to use subsets of the surfaces sitting respectively in smooth areas (for robust point to plane computation) and in high curvature areas (to keep significant features).

Besl [2] also suggested to use a k dimensional binary search tree, *k-D tree*, to decrease closest point search time. Theoretically, searching the closest point in a k-D tree having N entries is of complexity O(logN). Thus, the global complexity of the ICP algorithm becomes $O(N_p \log N_x)$. Practical results [8, 9] showed gains of about 15 for meshes containing ≈2500 pts. Benjemaa [1] proposed to project points into Z-buffers and then to perform local search in them. Recently, Jost [5] proposed the "neighbor search" closest point algorithm. It assumes that two neighbors on a surface possess closest points on the other surface that are neighbors and uses this property to obtain a first approximation of the closest point. It then refines the result with a local search in the range image. This method permits to avoid a global search for most points and leads to a closest points search (and therefore ICP) algorithm with a complexity of $O(N_p)$.

This paper proposes to accelerate the process by mixing a coarse to fine multiresolution approach with a reduction of the closest point search complexity. This is possible because both methods are quite independent and it should permit one to create a very fast ICP algorithm. This paper thus presents and analyses the multiresolution approach coupled with a k-D tree search or the neighbor search closest point algorithms, both in terms of speed and robustness.

2. Multiresolution ICP algorithm

The principle of multiresolution ICP is to make the first few iterations using down sampled data and to progressively 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 cost of each iterations 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.

Moreover, multiresolution matching is expected to have an added beneficial impact when combined with the tree search or the neighbor search methods. When using a tree search, the first iterations require a longer search time because of the coarse alignment of the data [5,11]. With multiresolution, these iterations are typically done with a low resolution, which greatly reduces the search time. When using the neighbor search closest point algorithm, higher resolutions and coarser matching decreases the exactness of the closest points pairing which typically reduces the range of successful initial configurations (SIC) [5]. Using a lower resolution for the initial iterations, should permit to avoid this problem.

2.1. Basic algorithm

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 fixed. Instead, the algorithm goes to the next resolution step automatically when a defined stop criterion - in this case when the change in coupling error falls below a threshold - is reached at the current one.

2.2. Cost reduction factor

As seen before, the complexity of the ICP algorithm changes when using tree search or neighbor search. From $O(N_p N_x)$. originally, it becomes $O(N_p \log N_x)$ with a tree search and $O(N_p)$ with the neighbor search. Consequently, the cost reduction factors between the different resolutions are different in these 3 cases: if N_p and N_x are reduced by a factor N, we obtain the cost reduction factors for an iteration showed in table 1.

Table 1. Cost reduction factors for data reduced with a factor N

	Reduce N_p	reduce N_x
normal ICP	N	N
k-D tree ICP	N	$\log(N_x)/(\log(N_x)-\log(N))$
neighbor search ICP	N	1

If we examine the k-D tree case, the value of the expression "$\log(N_x)/(\log(N_x)-\log(N))$" is typically situated between 1 and 2. This means that it is basically worthless to use a lower resolution on data set X when using a tree search. Firstly because the cost reduction

value is much lower than when using a similar resolution on data set P and secondly because it would imply to build a new tree at each resolution.

In the case of the neighbor search, the reduction of data set X doesn't have a direct influence on the cost at all. Note, however, 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.

Taking the above into account, the cost reduction factor for one iteration is N if both data sets are reduced by a factor N, in both k-D tree and neighbor search cases, opposed to a value of N^2 in the normal ICP case.

2.3. Estimation of the speedup gain

Now, say m is the number of iterations needed for a registration in monoresolution and n_i is the number of iterations performed at resolution step i for the same registration in a *k* steps multiresolution scheme. The cost of the complete registration is:

-monoresolution: $\quad\quad m\, C_1$

-multires.: $\quad n_1 C_1 + ... + n_k C_k = n_1 C_1 + \dfrac{n_2 C_1}{N} + ... + \dfrac{n_k C_1}{N^{k-1}}$

and the gain of multiresolution is:

$$G^N = \frac{m}{n_1 + \dfrac{n_2}{N} + \dfrac{n_3}{N^2} + ... + \dfrac{n_k}{N^{k-1}}} \quad\quad (1)$$

To analyze the numerical value of the gain, we make the hypothesis that the total number of iterations remains the same in both case. We will also distinguish 3 cases here on how iterations are distributed for each steps:

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.

Basically, they can be seen as medium, worst and best theoretical cases.

The associated linear functions of n_i are:

1. $n_i = \dfrac{m}{k}$, 2. $n_i = m\dfrac{k+1-i}{(1+2+...+k)}$, 3. $n_i = m\dfrac{i}{(1+2+...+k)}$

This gives us the following result:

$$G_1^N(k) = \frac{k}{\left(\sum_{i=1}^{k}\dfrac{1}{N^{i-1}}\right)} , \; G_2^N(k) = \frac{\sum_{j=1}^{k} j}{\left(\sum_{i=1}^{k}\dfrac{k+1-i}{N^{i-1}}\right)} , \; G_3^N(k) = \frac{\sum_{j=1}^{k} j}{\left(\sum_{i=1}^{k}\dfrac{i}{N^{i-1}}\right)}$$

Figure 1 shows a graphical presentation of these functions for N=4. One can see that the higher the number of steps, the higher the expected gain is. The theoretical best case (3) has a quadratic behavior, while gain functions 1 and 2 are nearly linear.
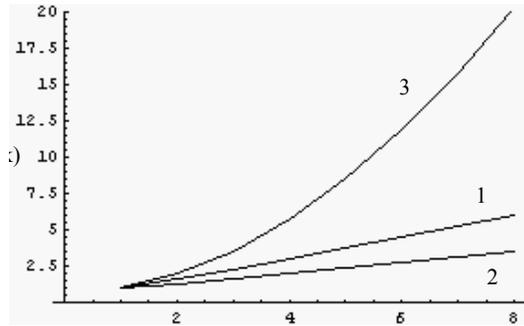
Figure 1. Expected multiresolution gains G(k) for fast ICP algorithms, for N= 4

3. Experiments and results

The multiresolution ICP algorithm has been tested on different data and compared with classical fast ICP algorithms using tree search and neighbor search. The comparison focuses on two features: computation speed and matching quality.

3.1. Matching quality comparison

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 algorithm must converge and the resulting alignment has to be the same or at least in the same error range as when matching using exact closest points.

To examine the domain of convergence, we use a method presented in [6] that compares the domains of successful initial configurations (SIC). A successful initial configuration is basically defined by a relative initial positioning of the two data sets that leads to a successful matching. As examining the whole configuration space, which possesses 6 dimensions, is not conceivable (due to both heavy computations and difficulty to display the results), the idea is to examine a subset of "interesting" initial configurations, liable to converge successfully. The dedicated setup [6] consists in a 3 dimensional space, the different initial configurations being defined by placing one surface on several points of the circumsphere of the other one. The results are plotted in 2D in a SIC-map, where black sectors represent the successful initial configurations (figure 3). The higher the number of black sectors the bigger the SIC domain.

3.2. Matching experiment

The presented experimental data set consists in 2 partially overlapping range images of a duck toy (figure 2). They typically represents the problem of views registration for virtual modeling. The overlap of both sets is approximately 50% of their total surface.

Full resolution data (approx. 24000 points, which implies k=5 resolution steps with N=4) and down-sampled data (approx. 6000 points, which implies k=4 resolution steps with N=4) are considered for testing. For practical reasons, the down sampled data have been used to compute the SIC ranges, as the time requested for the whole computation at full resolution was too long. On the other hand, a sub set of 30 initial configurations has been chosen to test the matching of full resolution data.
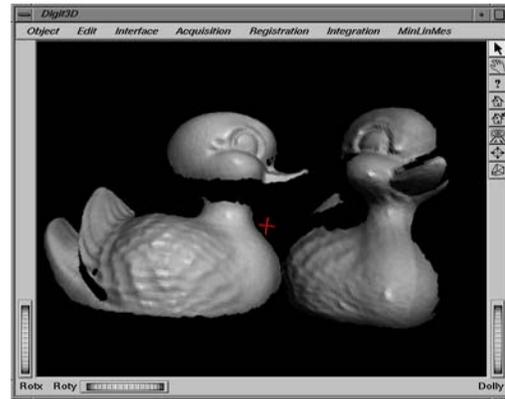

Figure 2. Surface rendering of the partially overlapping range images of a duck toy

Matching quality. A comparison of the average registration errors shows that, in case of successful convergence, all methods possess errors in the same range. These errors are in the order of less than 0.5° in rotation and 0.2% in translation.
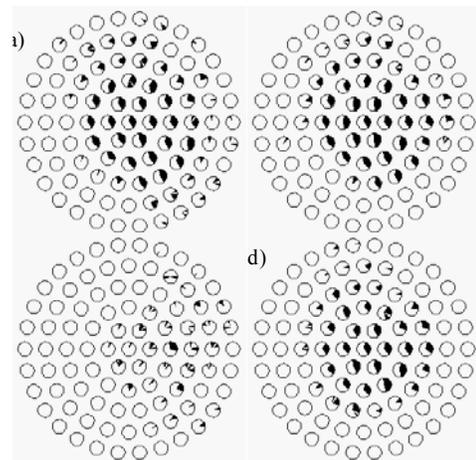

Figure 3. SIC-maps: a) tree search, b) multiresolution and tree search, c) neighbor search, d) multiresolution and neighbor search

The SIC-maps of the 4 considered cases using down sampled data can be found in figure 3. The results show that multiresolution does not affect the domain of successful convergence when coupled with a tree search and has clear beneficial effects on it when coupled with the neighbor search.

In the latter case, note that the resolution was too big for the neighbor search to work correctly (with a 9x9

search area) and the SIC range is much reduced in this case (3c). However, the SIC range obtained combining multiresolution and neighbor search (3d) is again very similar to the one obtained while using a k-D tree.

The results are even better when using full resolution data: in this case, the range of SIC is clearly improved by the multiresolution in both cases, using a k-D tree and the neighbor search.

These results confirm that the multiresolution scheme does not affect the matching quality for both the matching error and the domain of convergence. On the contrary, they show beneficial effects on the SIC range, especially when combined with the neighbor search algorithm.

Computation time. Table 2 presents a comparison of the average total computation time for the successful registrations, using the different acceleration methods on the full resolution data sets. The results related to neighbor search have been grayed out to reflect that they come from a single measurement.

The multiresolution scheme permits to reduce the total registration time by an average factor between 8.5 and 14 times for the different cases. These results are at least equal to or better than the theoretical best case value found with equation 4.3, $G_3^4(5) = 8.5$. Practically, one can effectively see a decreasing number of iterations in the higher resolutions steps, although close to but not as important as the best theoretical case. The high gain results can be further explained by the fact that other factors were not taken into account in the theoretical estimations, like the reduction of the total number of iterations, especially when using the neighbor search, or the beneficial effects of the very low resolution in the first iterations when using a tree search.

Table 2. Comparison of the total computation time of the registration (at full resolution) using the different acceleration methods

	avg. total time (s)	avg. number of iterations	avg. relative gain
kD tree	504.1	42	1.0
neighbor search	*259.0*	*90*	*1.9*
kD tree MR	59.3	36	8.5
neighbor search MR	18.8	34	26.8

Finally, the multiresolution neighbor search ICP can be up to 3 times faster than the multiresolution tree search ICP and nearly 27 times faster than a k-D tree ICP, which shows the advantage of neighbor search. This gain is also expected to be even higher for bigger data sets, due to the smaller complexity of the neighbor search algorithm.

One can also note that these relative gains refer to a fast ICP algorithm. The approximate gain in speed over a non-accelerated ICP algorithm is over 1300!

4. Conclusion

This paper proposes and analyses a solution for fast shape registration by applying, to the ICP algorithm, a combination of a coarse to fine multiresolution scheme and a fast closest point search.

Multiresolution combined with a k-D tree search and a neighbor search methods have been theoretically and experimentally compared in a 3D shape matching test. In the experiments, special attention is given to allow only good quality matches and not to go in any quality versus speedup compromise. Under these circumstances, the practically observed speedup gains are as good or even better than the best theoretical expectations.

Specifically, in both cases of fast ICP matching using a tree search or a neighbor search, multiresolution improves the registration speed by factors up to 14. Combining multiresolution with the neighbor search method, the registration can be up to around 27 times faster than when using a tree search - 1300 faster than a regular ICP -, which represents a very high-performance ICP algorithm.

Finally, the pure speedup potential goes together with improvements observed with respect to the convergence speed and the matching quality. This clearly shows that the multiresolution scheme exploits the fundamental nature of shape registration to substantially contributes to improve its computation.

References

[1] R. Benjemaa, F. Schmitt, "Fast Global Registration of 3D sampled Surfaces Using a Multi-Z-Buffer Technique", *Proc. 3DIM,* Ottawa, 1997, pp. 113-119.

[2] PJ. Besl, ND. McKay, "A Method for Registration of 3D Shapes*", IEEE PAMI*, vol. 14(2), 1992, pp. 239-256.

[3] 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.

[4] 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.

[5] T. Jost, H. Hügli, " A fast ICP algorithm for shape registration*", 2002, submitted*

[6] H. Hügli, C. Schütz, "Geometric Matching of 3D Objects: Assessing the Range of Successful Initial Configurations*", Proc. 3DIM*, Ottawa, 1997, pp. 101-106.

[7] S. Rusinkiewicz, M. Levoy, "Efficient Variants of the ICP Algorithm", *Proc. 3DIM*, Quebec, 2001, pp. 145-152.

[8] C. Schütz, *Geometric Points Matching of Free From 3D Objects*, PhD thesis, Uni. of Neuchatel, Switzerland, 1997.

[9] 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.

[10] G. Turk, M. Levoy, "Zippered Polygon Meshes from Range Images", *Proc. of ACM Siggraph*, Orlando, 1994, pp. 311-318.

[11] 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.