



A fast algorithm for ICP-based 3D shape biometrics

Ping Yan *, Kevin W. Bowyer

Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

Received 1 May 2006; accepted 5 November 2006

Abstract

In a biometrics scenario, gallery images are enrolled into the database ahead of the matching step, which gives us the opportunity to build related data structures before the probe shape is examined. In this paper, we present a novel approach, called “Pre-computed Voxel Nearest Neighbor”, to reduce the computational time for shape matching in a biometrics context. The approach shifts the heavy computation burden to the enrollment stage, which is done offline. Experiments in 3D ear biometrics with 369 subjects and 3D face biometrics with 219 subjects demonstrate the effectiveness of our approach.

© 2006 Elsevier Inc. All rights reserved.

Keywords: 3D shape; Shape matching; Iterative closest point; Biometrics

1. Introduction

Since its introduction by Chen and Medioni [1] and Besl and McKay [2], the Iterative Closest Point (ICP) algorithm has been widely used for 3D shape matching [1,3–5]. It has been used in a wide range of application areas, including the integration of range images [6,7] and alignment of CT and MR images [8]. Here, we are specifically interested in 3D shape matching for biometrics [9–13]. The ICP algorithm is known to be computationally expensive. With two clouds of points, source S (probe) and target T (gallery), the complexity of a typical single ICP iteration is $O(N_S \log(N_T))$ using a k-d tree data structure [9] in the expected case, where N_S is the number of points in the source and N_T is the number of points in the target. The ICP algorithm iteratively finds the minimum distance between two surfaces. With N_I iterations, the overall complexity is $O(N_I \times N_S \times \log(N_T))$ [2]. Therefore, matching high-resolution images of both source and target leads to a heavy computational load. A fast ICP implementation is crucial for practical use in biometrics.

Using shapes sensed by a 3D scanner is a major recent trend in biometrics [9–14]. A scan yields a 3D surface that can be used as a representation of the subject. In this paper, we illustrate our approach using both 3D ear and 3D face shapes. There are two types of images in a biometric application, gallery and probe. The gallery images are those that have been enrolled and whose identities are known to the system, while the probe images are those that need to be matched against the images in the gallery. In a recognition scenario, one probe is matched against all the images in the gallery, and the algorithm returns the match with the minimum error distance. In a verification scenario, one probe is matched against just one gallery entry, the one enrolled for the claimed identity. In recognition or verification experiments, enrollment occurs once and is followed by many instances of recognition.

One special characteristic of a biometrics application is that all gallery images are enrolled into the database before the matching takes place. Probe images are introduced into the system for matching. Taking advantage of the fact that the gallery images are enrolled prior to matching, we propose a novel method to accelerate the ICP matching. Our new method is called the “Pre-computed Voxel Nearest Neighbor”. The idea is to voxelize a volume which can hold the 3D gallery surface, and for each voxel to

* Corresponding author.

E-mail addresses: pyan@nd.edu (P. Yan), kwb@cse.nd.edu (K.W. Bowyer).

pre-compute its distance to the 3D gallery surface and save this for future use.

In Section 2, we review several fast ICP approaches. Then in Section 3 we give details of our approach. Section 4 addresses the applicability of our approach by using the ear and face biometrics, and experimental results are presented and analyzed. Finally, Section 5 discusses further refinements and possible future directions.

2. Literature review

In biometrics applications, 3D shape is used by many researchers in face biometrics [11–13,15–18], has also been used in ear biometrics [9,10], and has also been used in hand biometrics [14].

There have been a number of efforts to speed up ICP matching. One line of work is focused on fast algorithms for computing the nearest neighbor. The use of the k-d tree data structure appears to be the standard method in this area [19,2]. Cleary and co-workers analyzed the “Elias” algorithm for searching nearest neighbor in the n -dimensional Euclidean space [20]. They claimed that by using the “Elias” algorithm, the number of search points is independent of the total number of points on the surface.

In [21], Greenspan et al. proposed a novel nearest neighbor algorithm for small point sets. They report that “Elias” is much faster than a plain k-d tree, and that the “spherical constraint” method improves the speed still further. Zinßer et al. analyzed the performance of the nearest neighbor algorithm for ICP registration [22]. Their work is not limited to range images or triangle meshes, but also can be used with 3D point sets generated by structure-from-motion techniques.

Benjemaa [23] proposed a multi-z-buffer technique to accelerate the ICP algorithm. All points are projected in a z-buffer to perform the local search, and they claimed that this space partition speeds up the search for point-to-projection correspondences. But in order for the multi-z-buffer technique to work properly, the two surfaces need to be sampled with a high and uniform density.

Another line of work in this area looks at different subsample strategies to reduce computation time. One strategy is using multi-resolution approaches; that is, start with a coarse point set and use progressively finer point sets as the algorithm proceeds. The idea of the average distance between points in the current resolution in comparison to the average distance between matched points is the standard way to automate the switching between resolutions [24].

In [3], Gelfand et al. describe the importance of the quality of the point pairs. In the presence of noise or miscalibration in the input data, it is easy to create poor correspondences between pairs of points. Therefore, the least-squares technique might lead to wrong pose, or make it difficult for the algorithm to converge. They propose a technique to decide whether a pair of meshes has good quality by measuring the covariance matrix between two

meshes which have been sparsely and uniformly sampled. This technique tries to avoid the unstable movement between two surfaces by sampling the features from the input data which are the best constraint for this kind of movement.

In [25], Rusinkiewicz and Levoy discussed the variants of ICP which affect all phases of the algorithm. They list most of these variants, and evaluate their effects on the speed with which the correct alignment is reached. Also in the paper, they proposed a combination of ICP variants optimized for high speed.

Researchers have also looked at mixing the two lines of work, having some multi-resolution mixed with some constrained search for nearest neighbor. Jose and Hügli proposed a solution that combines a coarse to fine multi-resolution approach with the neighbor search [26]. The multi-resolution 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. They claim this technique reduces the time complexity of searching from $O(N \log(n))$ to $O(N)$, while preserving the matching quality [27].

Research related to ICP is also prominent in the graphics community. Leopoldseeder et al. used d^2 -tree to store a local quadratic approximant of the squared distance function to a surface [28]. Mitra et al. consider a general framework for matching two shapes represented by point clouds, in which the point-to-point and point-to-plane versions of ICP can be considered special cases [29]. Cheng et al. consider a method to fit a subdivision surface to an unorganized point cloud dataset [30]. However, none of these efforts are undertaken in a biometrics context. Also, while Leopoldseeder and Mitra use a subdivision of 3D space, they still use a tree search to find the closest point correspondence between two point sets, rather than reducing it to an indexing operation as in this paper.

3. Fast ICP matching for 3D shapes

The most time consuming part of the ICP algorithm is that for each point on the probe surface, the algorithm needs to find the closest point on the gallery surface. By using these pairs of corresponding points, the ICP algorithm iteratively refines the transforms between two surfaces, finding the translation and rotation to minimize the mismatch.

This search for a closest point on the gallery surface is initially done using a k-d tree, as described in [9], and each search takes $O(\log N_G)$, where N_G is the number of the points on the gallery surface. Our goal is to reduce this search time to a constant value. The main idea is that if we can pre-compute the distance from any point in the 3D space to the gallery surface, and use it when needed, then the search time for a closest point is a constant.

Our “Pre-computed Voxel Nearest Neighbor” approach is illustrated on the application of matching 3D surfaces for biometric recognition. At the time of enrollment, the

gallery 3D shape sits in a 3D volume that we think of as a set of voxels, shown in Fig. 1. In our experiment, the volume size depends on the size of the biometric source, face or ear. A detailed explanation is given in next section. Fig. 1 illustrates how the voxelization is done.

Placing the enrolled 3D surface into a voxelized volume, each point on the gallery surface falls into a voxel. A given voxel can be empty or hold one or more points from the gallery surface. If a probe surface is placed into this volume, every point on the probe surface should also fall into some voxel if the volume size is big enough. Suppose that there is a point P_1 on the probe surface that lies in the voxel V_1 in the volume. P_2 which lies in voxel V_2 is the point on the gallery surface which is closest to P_1 . The distance between two points P_1 and P_2 can be approximated by the distance between the center of the two voxels V_1 and V_2 with the precision of the voxel size, shown in Fig. 2.

In the ICP algorithm, given an enrolled surface in the volume, different probe surfaces attempt to find the minimum distance error to the enrolled surface. Here, the gallery surface is fixed, but the position of points on the probe surface varies within the volume from iteration to iteration. If all the points from the probe are within the volume which holds the gallery surface, each point should be in some voxel. For a given point P on the probe surface, suppose we know that its closest point on the gallery surface is P' and voxel V_p is the voxel this given point is in. The distance between P and P' is approximately equal to the distance between P' and the center of the voxel V_p , with the precision of the voxel size. Each voxel in the data structure can index a distance value pre-computed at enrollment of the surface. Therefore, given the position of one point, the index of the voxel can be calculated easily.

3.1. Volume size

The initial experiments used a volume around the 3D shape corresponding to the max size of the object. For an ear, the volume size is set to 8 cm wide, 10 cm tall and 8 cm deep. For a face, the volume size is set as 10 cm wide, 14 cm tall and 7 cm deep. The volume is subdivided into voxels. The voxel size is related to the precision of the

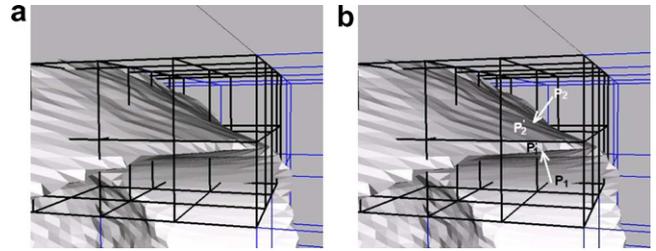


Fig. 2. Close look of voxels and example distance between voxel and gallery surface. (P_1 is the center of the voxel 1 and the closest point on the gallery surface to P_1 is P_1' . P_2 is the center of the voxel 2 and the closest point on the gallery surface to P_2 is P_2').

3D scanner. There is no point in making the voxelization a finer scale than the effective average depth resolution of the scanner. In our case, the average depth resolution of the Minolta Vivid 910 is no better than 0.5 mm. If the size for each voxel is $0.05\text{cm} \times 0.05\text{cm} \times 0.05\text{cm}$, we have $160 \times 200 \times 160 = 5.12\text{M}$ voxels per volume for an ear. The fixed volume size is usually larger than 3D objects in the volume, and the reason that it has extra space is that we need to consider the orientation of the 3D objects. Even though the width of the ear is usually small, the overall crossing will be large if the ear is rotated along the z axis instead of straight up. Unfortunately much space is wasted for fixed volume. Thus, we reduce the volume size by applying principal component analysis (PCA) on the 3D point cloud for the ear to give it a standardized pose.

Principal components analysis is used for computing the dominant variances representing a given data set. As we apply PCA on the 3D data, it yields three eigenvectors, the first eigenvector is the direction of greatest variation in the data, the second eigenvector is the direction of second greatest variation, and the third eigenvector is the third greatest variation. And all eigenvectors are orthogonal to each other. According to our 3D shape data, the greatest variation is related to the height of the 3D shape, the second one to the width, and third one to the depth of the data. After obtaining these three eigenvectors, a new coordinate system $[V_x^T \ V_y^T \ V_z^T]$ is generated, each V_i is a vector. If we project the old 3D points into the new system, the ranges along these three new axes represent the size of a box enclosing the 3D shape.

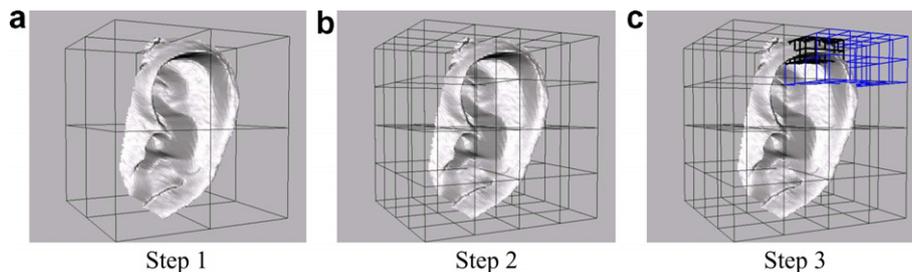


Fig. 1. Voxelization of 3D Ear Data. In order to show it clearly, we present it from coarse to fine. In step 1 the volume is subdivided into 8 small voxels, and in step 2 each small voxel is subdivided into 8 even smaller voxels. And continue this subdivision until the size of each voxel is smaller than a threshold. (To implement this idea, we subdivide the volume once using a fixed voxel size).

$$[X'Y'Z'] = [XYZ] * [V_x^T \ V_y^T \ V_z^T] \quad (1)$$

$$\text{Width} = \max(X') - \min(X') \quad (2)$$

$$\text{Height} = \max(Y') - \min(Y') \quad (3)$$

$$\text{Depth} = \max(Z') - \min(Z') \quad (4)$$

Fig. 3 illustrates the steps of this procedure. When compared to the width, height, and depth in Figs. 3(a) and (c), the overall size of the bounding box of the new 3D shape is smaller. For the ear experiment, the overall file size can be reduced by a factor of 10. With a smaller file size to save the information, it requires less memory to build and read the data. Therefore, this reduces the building time, and sometimes it also reduces the matching time when swapping is needed in the old approach.

In our experiments, a uniform voxel size is used. Using a non-uniform voxel size could result in a smaller data structure. However, if a non-uniform voxel size is used, then the accuracy of the pre-computed correspondence and distance that is stored for each voxel will effectively vary with the voxel size. Consider that if one large voxel replaces a neighborhood

of nine smaller voxels, then every probe point that falls in that larger voxel will index to the same pre-computed corresponding point and distance. The accuracy of the distance and correspondence will be coarser.

3.2. Implementation

To implement our strategy of pre-computed voxel nearest neighbor, we compute ahead of time for each voxel in the 3D space, it's closest point on the gallery 3D shape. The first step is to place the 3D surface into a volume whose center is the center of the 3D surface. The position of the gallery surface center x , y and z are defined as following: $x_{\text{center}} = \frac{x_{\text{max}}+x_{\text{min}}}{2}$, $y_{\text{center}} = \frac{y_{\text{max}}+y_{\text{min}}}{2}$, $z_{\text{center}} = \frac{z_{\text{max}}+z_{\text{min}}}{2}$. Fig. 4 shows a volume holding both gallery and probe. For each voxel element in the volume, we use a k-d tree to find the closest point on the gallery surface to that voxel's center. Once the point is found, the index of the point is stored as the value of the voxel element. A data structure *VoxelElement*[Width][Height][Depth] is used to represent the subdi-

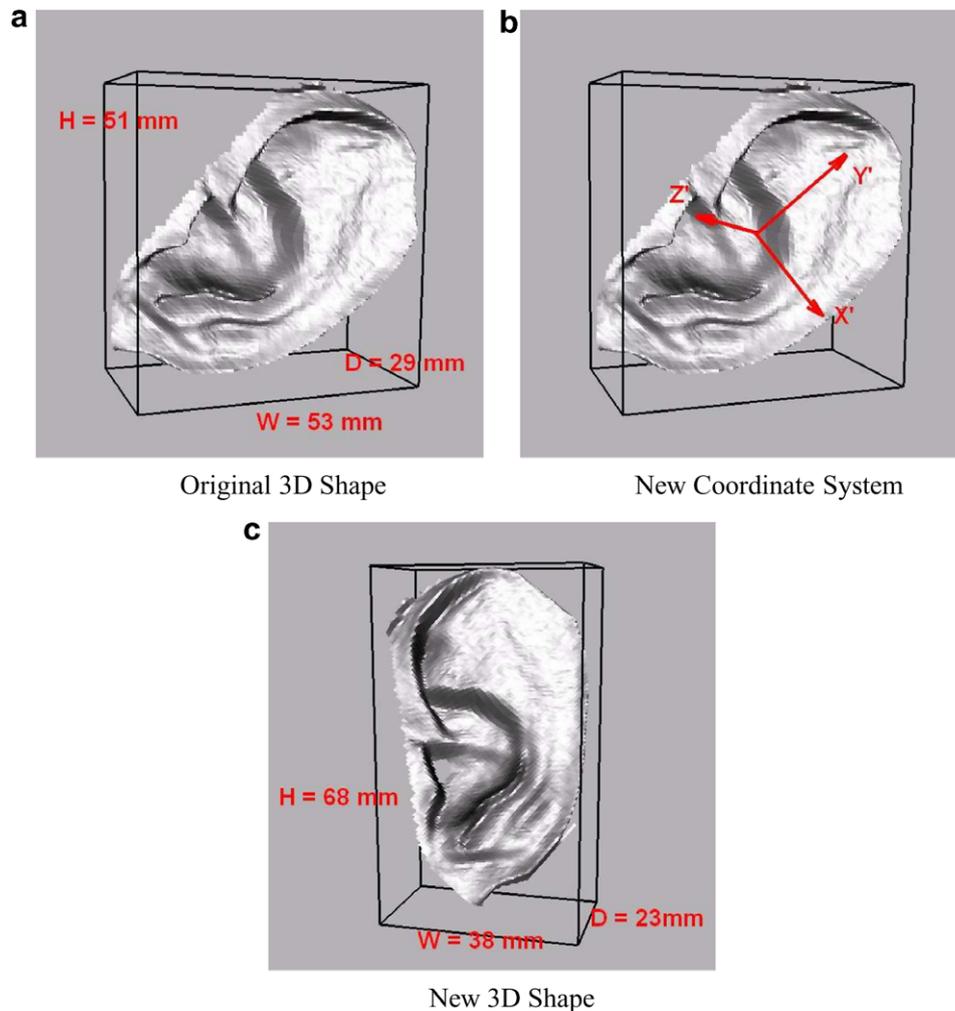


Fig. 3. Steps to calculate the volume size. In part (b), a new coordinate system is generated from eigenvectors of the covariance matrix, where Y' is according to the direction of the largest variance in the dataset, X' to the second, and Z' to the third. Part (c) shows the new 3D shape after projecting every old point onto new coordinate system.

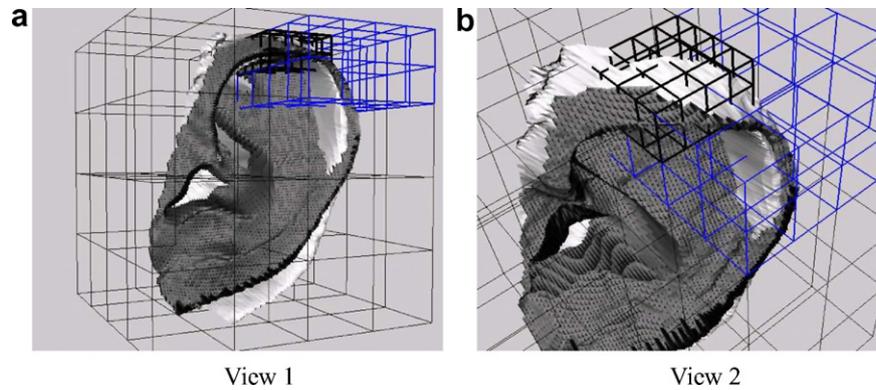


Fig. 4. Gallery and probe images show in the same volume. Each voxel in the volume corresponds to a point index on the gallery surface.

vision of 3D space into voxels, and the value of width, height and depth comes from 4. The value of $VoxelElement[x][y][z]$ is the index of the gallery point, which is closest to the point (x,y,z) . We store the index of the point instead of the point position to save space. Pre-computed results are saved to a file which can be read into memory when needed.

Then, computing the closest neighbor for a current position of the probe surface is simply indexing into the voxel data structure. Thus, constant computational time instead of $O(\log N_G)$ is achieved. This is blazingly fast in comparison to any of the other nearest neighbor finding methods, but of course it is offset by the size of the storage required. Furthermore, since the access time is constant, we can use the finest resolution for the gallery image, which avoids the computation expense of using the point-to-surface approach [1]. Fig. 4(a) shows an example with probe surface matching to gallery surface.

4. Experiments

In order to evaluate the efficiency of this method, we compare the recognition rate, space and running time between the original algorithm and our proposed approach.

We present results using ear range data from 369 subjects and face range data from 219 subjects, and each subject has two images taken on two different date. For each subject, the earlier 3D images are used for the gallery, and the later 3D images are used as probes. The detailed description of the ear and face extraction from raw image can be found in [31,32]. For the ear experiment, the gallery images use the full resolution, and the probes are subsampled by every 4 rows and every 4 columns. The average number of points is 5500 for a gallery ear shape, and 400 for a probe ear shape. And for the face experiments, both gallery and probe images are subsampled by every 4 rows and every 4 columns. The average number of points on a gallery and a probe surface are 4000 and 3000, respectively, for face shapes. In addition, different voxel sizes are tested, and comparison results are presented. The system runs on

dual-processor Pentium Xeon 2.8 GHz machines with 2 GB RAM, and the implementation is written in C++.

4.1. Voxel size

Three voxel sizes are examined using the same dataset for both ear and face biometrics. For the ear experiments, they are 1 mm^3 , 0.5 mm^3 and 0.25 mm^3 . For the face experiments, they are 2 mm^3 , 1 mm^3 and 0.5 mm^3 . The reason for using different voxel size for ear and face is because the gallery face images are subsampled by every 4 columns and rows. Before the matching procedure takes place, we build the volume for every gallery ear/face. For each voxel in the volume, a k-d tree structure is used to find the closest point on the gallery and we save the results on the disk. In order to utilize our method, we read one voxelized gallery data structure into memory and match it against all the probes. Therefore, our recognition experiment has two processes, offline building and online matching. Tables 1 and 2 illustrate the time requirement for each process.

For the ear experiments, all the images are acquired using a Minolta Vivid 910 with the “Tele” lens, and the subject sat approximately 1.5 m away from the sensor. Within that distance range, the sensor has a depth accuracy of approximately 0.55 mm. According to our results, going to a finer voxel size from 0.5 to 0.25 mm does not yield much in term of increased accuracy, yet, it requires significantly more storage space and longer time to process. Even though the access time is a constant value, when the number of voxels is too big, it will exceed the size of available memory, and force the algorithm to use swap space, which will slow down the computation. If we increase the voxel size from 0.5 to 1 mm, the reading time drops, the matching time is at the same level, and the performance drops by around 0.3%, which is not statistically significantly different from the smaller voxel size. For the face experiments, the image acquisition is the same. But since the gallery images are subsampled by 4×4 , there is no statistically significant difference in performance for voxel size variations.

Fig. 5 compares the original ICP algorithm and our pre-computed ICP on the ear dataset. We compared voxel

Table 1
Ear biometrics: Different parameters affected by voxel size

Voxel size (mm)	Building time (per ear) (s)	Reading time (per ear) (s)	Matching time (1 against 369)	File size (369 Images) (s)	Performance
1	10–50	0	15–25	127 M	97.0% (1)
0.5	30–200	1	20–30	1009 M	97.3% (2)
0.25	100–500	1–5	20–30	7.8 G	97.3% (3)

At 0.1 level of significance, there is no statistically significant difference between (1), (2) and (3). Times are given as a range; for example 15–25 s. This is an approximate range for min to max time required across 369 possible probes, any one of which can be matched against 369 gallery images.

Table 2
Face biometrics: Different parameters affected by voxel size

Voxel size (mm)	Building time (per face) (s)	Reading time (per face) (s)	Matching time (1 against 219)	File size (219 Images) (s)	Performance
2	25–35	0–1	80–90	983 M	93.6% (1)
1	150–160	5–6	80–90	15 G	94.1% (2)
0.5	1500–1600	35–45	80–95	63 G	93.2% (3)

At 0.1 level of significance, there is no statistically significant difference between (1), (2) and (3).

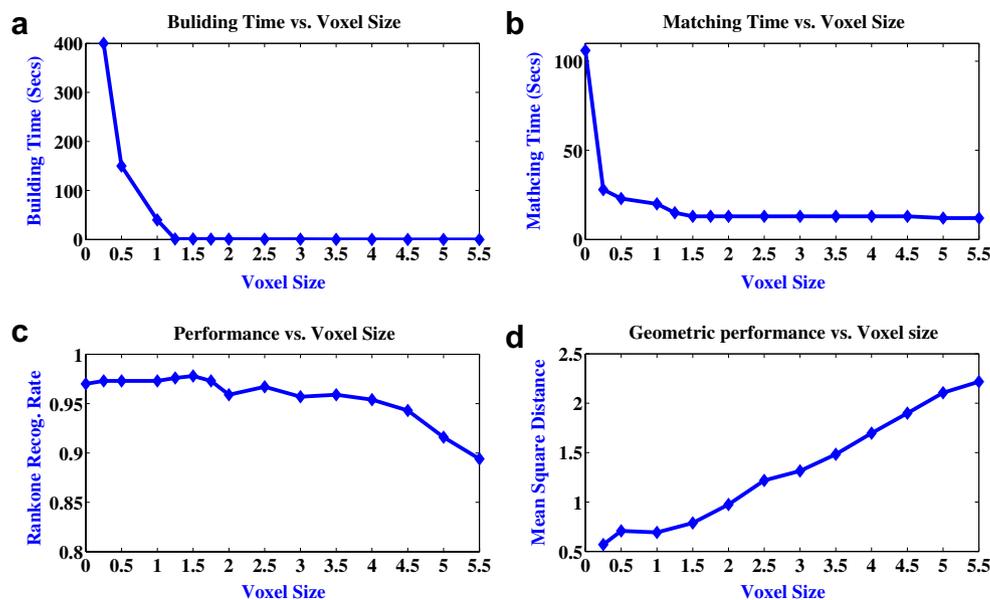


Fig. 5. Ear experiments: How does voxel size affect building time, matching time, rank-one recognition rate and geometric performance.

building time, matching time and recognition performance and mean-square distance of the final aligned point sets for voxel sizes from 0.25 to 5 mm. Fig. 5(b) demonstrates that pre-computed ICP is much faster than the original ICP. When the voxel size increases, the matching time decreases. But once the voxel size increases up to 1.5 mm, there is little reduction in matching time. Fig. 5(a) shows that the voxel building time drops dramatically when the voxel size is increased from 0.25 to 1 mm, while the performances stays at essentially the same level. After the voxel size increased beyond 1.5 mm, the building time can be almost ignored. The performances of the pre-computed ICP keeps better than 95% recognition rate even when the voxel size is increased up to 4 mm. In order to demonstrate the quality of the final mean-square distance as a function of the voxel size, Fig. 5(d) shows an example from a correct match pair.

Table 3

Ear biometric: Run time vs. Gallery size for both original ICP and pre-computed ICP

Gallery size	Original ICP (s)	Pre-computed ICP (s) (voxel = 0.5 mm)
10	5	3
50	20	7
100	35	10
200	75	15
300	106	25
369	150	30

As the voxel size increases, the mean-square distance increases approximate linearly.

Table 3 illustrates how execution time increases when gallery size gets larger for both original ICP and pre-com-

puted ICP. When the gallery size is small, there is no advantage to the voxel approach. However, for very large galleries the voxel approach yields an enormous improvement in speed. Here, we suppose all the gallery images can be kept in the memory. In a real biometrics application, some or all of the gallery might be kept in memory all the time.

5. Improvement

As we stated in the previous section, the most time consuming part of the ICP algorithm is closest point searching. There are two common ways to find the closest point, point-to-point and point-to-surface. A detailed comparison between them for a biometric application can be found in [9]. The point-to-point approach is fast, and accurate when all the points on the probe surface can find a good closest point on the gallery surface. But if the gallery is subsampled or coarse in the original, the point-to-point approach loses accuracy. On the other hand, the greatest advantage of the point-to-surface approach is that it is accurate through all the different subsample combinations. But this behavior comes at a substantial computational expense. Our voxel algorithm can shift the computation burden to offline, therefore if the gallery images are not in a fine resolution, the point-to-surface method for pre-computed distance should be able to yield better performance without increasing the running time for the recognition. This is proved by the experimental results. By using point-to-surface method for pre-computing, the ear recognition rate is improved from 97.3% to 98.7%, and the face recognition rate is improved from 94.1% to 96.4%. The improvement is more obvious in the face recognition experiment, which also demonstrates that the point-to-surface method is more accurate than point-to-point method when the gallery images are coarse.

6. Summary and discussion

The main contribution of this paper is the “Pre-computed Voxel Closest Neighbor” strategy to improve the speed of the ICP algorithm for use in biometrics. This technique is aimed at a particular application in human identification. The idea is based on the possibility of computing the data structure before the matching procedure taking place.

Different voxel sizes are examined, and the performance and running time are compared with the results from the original ICP algorithm. Our experimental results verify the performance of our approach on our 369 subjects dataset for ear biometrics, and 219 subjects dataset for face biometrics. The online matching time drops significantly when we use the pre-computed results from the enrolled 3D shape offline computation. Our results demonstrate that for very large galleries the voxel approach yields a dramatic improvement in speed. In real biometric security applications, the number of persons in the gallery could easily be in the thousands or larger.

The 2D and 3D image data sets used in this work are available to other research groups. See the web page www.nd.edu/cvrl for the release agreement and details.

Acknowledgments

Biometrics research at the University of Notre Dame is supported by the National Science Foundation under grant CNS01-30839, by the Central Intelligence Agency, by the US Department of Justice/National Institute for Justice under Grants 2005-DD-CX-K078 and 2006-IJ-CX-K041, by the National Geo-spatial Intelligence Agency, and by UNISYS Corp. The authors thank Patrick Flynn and Jonathan Phillips for useful discussions about this area.

References

- [1] Y. Chen, G. Medioni, Object modeling by registration of multiple range images, *Image and Vision Computing* 10 (1992) 145–155.
- [2] P. Besl, N. McKay, A method for registration of 3-D shapes, *IEEE Transaction on Pattern Analysis and Machine Intelligence* 14 (1992) 239–256.
- [3] N. Gelfand, S. Rusinkiewicz, L. Ikemoto, M. Levoy, Geometrically stable sampling for the ICP algorithm, in: *Fourth International Conference on 3-D Digital Imaging and Modeling*, October 06–10, 2003, pp. 260–267.
- [4] D. Zhang, M. Hebert, Harmonic maps and their applications in surface matching, *Computer Vision Pattern Recognition* 2 (1999).
- [5] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scene, *IEEE Transaction on Pattern Analysis and Machine Intelligence* 21 (1999) 433–449.
- [6] G. Turk, M. Levoy, Zippered polygon meshes from range images, *Proceeding of ACM Siggraph* 94 (1994) 311–318.
- [7] A. Hilton, A.J. Stoddart, J. Illingworth, T. Windeatt, Marching triangles: range image fusion from complex object modeling, *International Conference on Image Processing* 2 (1996) 381–384.
- [8] J. Feldmar, N. Ayache, F. Betting, 3D-2D projective registration of free-form curves and surfaces, *Computer Vision and Image Understanding* 65 (1997) 403–424.
- [9] P. Yan, K.W. Bowyer, ICP-based approaches for 3D ear recognition, in: *Biometric Technology for Human Identification II*, *Proceedings of SPIE*, vol. 5779, 2005, pp. 282–291.
- [10] H. Chen, B. Bhanu, Contour matching for 3D ear recognition, in: *Seventh IEEE Workshop on Application of Computer Vision*, 2005, pp. 123–128.
- [11] G. Medioni, R. Waupotitsch, Face recognition and modeling in 3D, in: *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2003)*, 2003, pp. 232–233.
- [12] K. Chang, K.W. Bowyer, P.J. Flynn, Multiple nose region matching for 3D face recognition under varying facial expression, *IEEE Transaction on Pattern Analysis and Machine Intelligence* 28 (2005) 1695–1700.
- [13] X. Lu, D. Colbry, A.K. Jain, Three-dimensional model based face recognition, in: *7th IEEE Workshop on Applications of Computer Vision*, 2005, pp. 156–163.
- [14] D.L. Woodard, P.J. Flynn, Finger surface as a biometric identifier, *Computer Vision and Image Understanding* 100 (2005) 357–384.
- [15] X. Lu, A.K. Jain, Deformation analysis for 3D face matching, *7th IEEE Workshop on Applications of Computer Vision (WACV '05)*, 2005, pp. 99–104.
- [16] T. Papatheodorou, D. Rueckert, Evaluation of 3D face recognition using registration and pca, LNCS 3546: *International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA 2005)* (2005) 997–1018.

- [17] G. Passalis, I. Kakadiaris, T. Theoharis, G. Toderici, N. Murtuza, Evaluation of 3D face recognition in the presence of facial expressions: an annotated deformable model approach, *IEEE Workshop on Face Recognition Grand Challenge Experiments (June 2005)* 171–171.
- [18] B. Gokberk, A.A. Salah, L. Akarun, Rank-based decision fusion for 3D shape-based face recognition, *LNCS 3546: International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA 2005) (July 2005)* 1019–1028.
- [19] D. Simon, M. Hebert, T. Kanade, Techniques for fast and accurate intra-surgical registration, in: *First International Symposium on Medical Robotics and Computer Assisted Surgery, 1994*, pp. 90–97.
- [20] J.G. Cleary, Analysis of an algorithm for finding nearest neighbors in Euclidean space, *ACM Transactions on Mathematical Software (TOMS) 5 (1979)* 183–192.
- [21] M. Greenspan, G. Godin, A nearest neighbor method for efficient ICP, in: *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling, 2001*, pp. 161–170.
- [22] T. Zinßer, J. Schmidt, H. Niemann, Performance analysis of nearest neighbor algorithms for ICP registration of 3-D point sets, in: *8th International Fall Workshop of Vision, Modeling, and Visualization, November 19–21, 2003*.
- [23] R. Benjemaa, F. Schmitt, Fast global registration of 3D sampled surfaces using a multi-z-buffer technique, in: *International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97), 1997*, pp. 113–119.
- [24] K.-L. Low, A. Lastra, Reliable and rapidly-converging ICP algorithm using multiresolution smoothing, in: *Fourth International Conference on 3-D Digital Imaging and Modeling, October 06–10, 2003*, pp. 171–178.
- [25] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: *Third International Conference on 3D Digital Imaging and Modeling (3DIM), June 2001*, pp. 145–152.
- [26] T. Jost, H. Hügli, Fast ICP algorithms for shape registration, *Lecture Notes in Computer Science 2449 (2002)* 91–99.
- [27] T. Jost, H. Hügli, Multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images, in: *Fourth International Conference on 3-D Digital Imaging and Modeling, October 06–10, 2003*, pp. 427–433.
- [28] S. Leopoldsdeder, H. Pottmann, The d2-tree: A hierarchical representation of the squared distance function, in: *Technical Report No. 101. Institute of Geometry, Vienna University of Technology, 2003*.
- [29] N.J. Mitra, N. Gelfand, H. Pottmann, L. Guibas, Registration of point cloud data from a geometric optimization perspective, in: *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, 2004*, pp. 22–31.
- [30] K.-S.D. Cheng, W. Wang, H. Qin, K.-Y.K. Wong, H. Yang, Y. Liu, Fitting subdivision surfaces to unorganized point data using sdm, in: *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04), 2004*, pp. 16–24.
- [31] P. Yan, K.W. Bowyer, Biometric recognition using three-dimensional ear shape, *IEEE Transaction on Pattern Analysis and Machine Intelligence* to appear.
- [32] K. Chang, K. Bowyer, P. Flynn, Face recognition using 2D and 3D facial data, in: *Workshop on Multimodal User Authentication, 2003*, pp. 25–32.