

Elastic Convolved ICP for the Registration of Deformable Objects

Ryusuke Sagawa Kiyotaka Akasaka Yasushi Yagi
Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki-shi, Osaka, 567-0047, JAPAN
{sagawa, akasaka, yagi}@am.sanken.osaka-u.ac.jp

Henning Hamer Luc Van Gool
ETH Zürich
Sternwartstrasse 7, CH - 8092 Zurich, Switzerland
{hhamer, vangool}@vision.ee.ethz.ch

Abstract

This paper describes a method for registering deformable 3D objects. When an object such as a hand deforms, the deformation of the local shape is small, whereas the global shape deforms to a greater extent in many cases. Therefore, the local shape can be used as a feature for matching corresponding points. Instead of using a descriptor of the local shape, we introduce the convolution of the error between corresponding points for each vertex of a 3D mesh model. This approach is analogous to window matching in 2D image registration. Since the proposed method computes the convolution for every vertex in a model, it incorporates dense feature matching as opposed to sparse matching based on certain feature descriptors. Through experiments, we show that the convolution is useful for finding corresponding points and evaluate the accuracy of the registration.

1. Introduction

Recently, fast range scanners [16, 23, 10, 13] have been developed that can capture range scans of a moving object in real time. To analyze the motion of the object, matching corresponding points between range scans is important. In particular, the registration of range scans is interesting if the object deforms in a non-rigid manner.

When an object deforms, the deformation of the local shape tends to be small whereas the global shape deforms to a greater extent in many cases. For example, when a hand deforms by bending the fingers, the global shape deforms quite markedly because the joints are flexible; however, the deformation of the local shape at each point on the surface is small, except near the joints. The local shape is therefore,

nearly invariant and can be used as a feature for matching corresponding points.

In order to use the local shape for matching corresponding points, several studies [9, 11, 6, 20, 5] have proposed feature descriptors for the local 3D shape. These descriptors are used for sparse feature points and for rigid registration. The descriptors are also used for deformable registration [2, 18]; however, finding feature points is problematic in many cases because salient features may not be detected and the correspondence between feature points may not be found in all cases. Another approach using local shapes is the piecewise iterative closest point (ICP) method [4, 8] that assumes a range scan as a set of several piecewise rigid parts. Since the number of parts is small, the degrees of freedom are not sufficient for general deformable motion. Since the motion of each vertex is different in deformable registration, the local shape should be used for all vertices. Sparse feature points are, therefore, used for initial registration to find rough correspondences.

Thus, instead of using a descriptor of the local shape, we introduce the convolution of the error between the corresponding points for every vertex in a 3D mesh model. This approach is analogous to window matching in 2D image registration. Window matching is commonly used to find corresponding points when an image deforms in a non-rigid manner, for example, in the case of computing optical flow. Since the proposed method computes the convolution for every vertex in a model, it incorporates dense feature matching in contrast to the sparse matching based on certain feature descriptors.

Several methods for deformable registration [1, 4, 19, 18, 2, 12, 7, 21] have been proposed thus far. Some methods estimate the position of every vertex after deformation, while others [1, 21] estimate the rigid transformation, including rotation and translation, of every vertex in the mesh.

These methods extend the ICP method [3] for a deformable mesh. The reason that ICP-based methods are often used and are successful in accurate rigid registration, is that they use dense vertices in a model for registration. Therefore, we propose a method for non-rigid registration that introduces dense local features into the ICP framework, instead of using sparse feature points. Since rotation and translation of each vertex is required for analyzing the motion of an object, the proposed method estimates both these parameters by minimizing the convolved error between corresponding points.

This paper is organized as follows. The formulation of the elastic convolved ICP is described in Section 2. The detailed implementation of the solution is explained in Section 3. The experiments on the proposed method and an evaluation of the accuracy of the model are discussed in Section 4. Finally, conclusions are presented in Section 5.

2. Elastic Convolved ICP

In this section, we describe a method of finding the correspondence between two range scans for each vertex. Let us consider the situation of finding corresponding points between the two range scans shown in Fig. 1. In this paper, the proposed method finds the corresponding points along the line of sight of the camera, a technique commonly used during the iterations in variants of the ICP-based method [17]. An ICP-based method estimates the rigid transformation $T = [R \ t]$, where R is the rotation matrix and t the translation vector, by minimizing the distance of the corresponding points as follows [14]:

$$T = \arg \min_T \sum_i (\mathbf{n}_i \cdot (T\mathbf{x}_i - \mathbf{y}_i))^2, \quad (1)$$

where \mathbf{x}_i is a vertex in range scan 1, \mathbf{y}_i is the corresponding point, and \mathbf{n}_i is the normal vector at \mathbf{y}_i . \mathbf{x}_i and \mathbf{y}_i are represented in a homogeneous coordinate system.

In the proposed deformable registration, the rigid transformation is computed for each vertex \mathbf{x}_i by adding a smoothness constraint to the transformation of neighboring vertices. The error E to be minimized is represented by $E = E_d + E_s$, where E_d is the data term, in which T in Eq. 1 is replaced by the rigid transformation T_i for each vertex \mathbf{x}_i , and E_s is the regularization term expressed as [1]:

$$E_s = \sum_i \sum_{j \in \Omega} \|\lambda_{ij} \circ (T_i - T_j)\|_F^2, \quad (2)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, λ_{ij} is a weight matrix of the smoothness term, and \circ indicates the element-wise multiplication of two matrices.

In the formulation in Eq. 2, the local shape of the range scan is not preserved and it deforms like paper. Since finding the correspondence along the line of sight does not consider the local shape, a shift in the correspondence occurs if

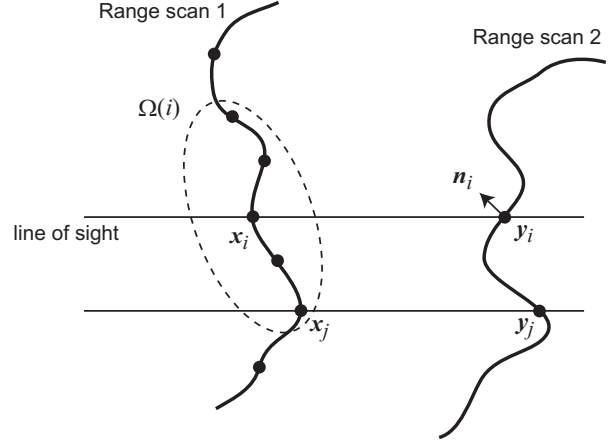


Figure 1. The corresponding point \mathbf{y}_i for each vertex \mathbf{x}_i in range scan 1 is found along the line of sight. To compute the transformation T_i for \mathbf{x}_i , the convolution of the error is computed by considering the neighborhood $\Omega(i)$ of \mathbf{x}_i .

the object moves in a direction perpendicular to the line of sight. To consider the local shape, we introduce the convolution of the difference between the corresponding points, which is analogous to window matching in methods used for stereo matching and optical flow. Hence, the data term becomes:

$$E_d = \sum_i \sum_{j \in \Omega(i)} \gamma_{ij} (\mathbf{n}_j \cdot (T_i \mathbf{x}_j - \mathbf{y}_j))^2, \quad (3)$$

where $\Omega(i)$ is the neighborhood of vertex \mathbf{x}_i , and γ_{ij} is the weight. To estimate the transformation T_i of vertex \mathbf{x}_i , the neighboring vertices \mathbf{x}_j are used to compute the error.

3. Implementation

3.1. Minimizing the Error E

Since the minimization of E is nonlinear, the transformation T_i for each vertex is computed iteratively. First, the initial transformation is given by the user and the corresponding points \mathbf{y}_i for each vertex \mathbf{x}_i are found along the line of sight. Next, the error E is minimized by using the given correspondence. The proposed method iterates, alternatively finding correspondences and minimizing the error until convergence.

In each iteration, the transformation is updated as follows [15]:

$$T_i^{(k+1)} = \Delta T_i \begin{bmatrix} T_i^{(k)} \\ 0 \ 0 \ 0 \ 1 \end{bmatrix}, \quad (4)$$

where $T_i^{(k)}$ is the transformation in the k -th iteration and ΔT_i is the updating parameter in the $(k + 1)$ -th iteration. By assuming that the rotation of ΔT_i is small, elements of

ΔT_i are given by

$$\Delta T_i = \begin{bmatrix} 1 & -c_{3i} & c_{2i} & t_{1i} \\ c_{3i} & 1 & -c_{1i} & t_{2i} \\ -c_{2i} & c_{1i} & 1 & t_{3i} \end{bmatrix}, \quad (5)$$

where c_{1i} , c_{2i} , and c_{3i} are rotation parameters, and t_{1i} , t_{2i} , and t_{3i} are translation parameters.

Let $\mathbf{m}_i = [c_{1i} \ c_{2i} \ c_{3i} \ t_{1i} \ t_{2i} \ t_{3i}]^T$ and $\boldsymbol{\delta} = [\mathbf{m}_1^T \ \dots \ \mathbf{m}_i^T \ \dots \ \mathbf{m}_N^T]^T$, where N is the number of vertices. Since the error given by Eq.2 and Eq.3 can be modified to take the following form

$$E = \sum_l^L (\mathbf{q}_l \boldsymbol{\delta} - r_l)^2, \quad (6)$$

where \mathbf{q}_l and r_l are a $1 \times 6N$ vector and a scalar, respectively, and L is the number of constraints, the least squares solution of the updating vector $\boldsymbol{\delta}$ is given by solving the following linear equation:

$$Q\boldsymbol{\delta} = \mathbf{r}, \quad (7)$$

where the l -th rows of Q and \mathbf{r} are \mathbf{q}_l and r_l , respectively.

3.2. Solving the Linear Equation

The data term in Eq.3 is modified as

$$\begin{aligned} & \mathbf{n}_j \cdot \left(T_i^{(k+1)} \mathbf{x}_j - \mathbf{y}_j \right) \\ = & \mathbf{n}_j^T \begin{bmatrix} 0 & x'_{3j} & -x'_{2j} & 1 & 0 & 0 \\ -x'_{3j} & 0 & x'_{1j} & 0 & 1 & 0 \\ x'_{2j} & -x'_{1j} & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{m}_i \\ & + \mathbf{n}_j^T (T_i^{(k)} \mathbf{x}_j - \mathbf{y}_j) \\ = & A_{ij} \mathbf{m}_i - a_{ij} \end{aligned} \quad (8)$$

where $[x'_{1j} \ x'_{2j} \ x'_{3j}]^T = T_i^{(k)} \mathbf{x}_j$, and A_{ij} and a_{ij} are a 1×6 vector and a scalar, respectively. Thus, each combination of vertex x_i and its neighbor x_j gives one constraint that the $6i$ -th to $(6i+5)$ -th columns of \mathbf{q}_l are $\gamma_{ij} A_{ij}$ and $r_l = \gamma_{ij} a_{ij}$.

Similarly, the Frobenius norm in Eq.2 is modified as

$$\begin{aligned} & \left\| \lambda_{ij} \circ \left(\Delta T_i \begin{bmatrix} T_i^{(k)} \\ 0 \ 0 \ 0 \ 1 \end{bmatrix} - \Delta T_j \begin{bmatrix} T_j^{(k)} \\ 0 \ 0 \ 0 \ 1 \end{bmatrix} \right) \right\|_F^2 \\ = & \left\| \lambda_{ij} \circ (B_i \mathbf{m}_i + \mathbf{b}_i - B_j \mathbf{m}_j - \mathbf{b}_j) \right\|^2, \end{aligned} \quad (9)$$

where B_i and B_j are 12×6 matrices, and \mathbf{b}_i and \mathbf{b}_j are 12×1 vectors. λ_{ij} is a 12×1 vector that lists the elements of λ_{ij} . Further details of B_i and \mathbf{b}_i are given in the Appendix. From the definition of the Frobenius norm, each regularization term gives 12 constraints that each element of Eq.9 is zero.

The size of Q is $L \times 6N$. If the number of neighbors of x_i is M , $L = 13MN$. Since N and M are $1K \sim 30K$

and $10 \sim 300$, respectively, in the experiments Q is quite a large, yet sparse, matrix. Eq.7 can be solved by applying Cholesky factorization to the associated normal equations

$$Q^T Q \boldsymbol{\delta} = Q^T \mathbf{r}. \quad (10)$$

Some sparse linear solvers such as TAUCS [22] can solve the system efficiently. However, the system can be rank deficient and the solution is not always unique. To determine a unique solution in such cases, we add μI to $Q^T Q$ according to the Levenberg-Marquardt algorithm, where μ is a damping parameter and I is an identity matrix.

Furthermore, since creating Q and computing $Q^T Q$ are time-consuming processes, we implemented an OpenMP-based parallel process for computing each row of Q and $Q^T Q$ using a multicore CPU.

3.3. Defining the Neighborhood Ω

In this paper, we define neighboring vertices, x_j , of vertex x_i by the geodesic distance between x_i and x_j . Since we assume that a range scan can be represented as a mesh model, the geodesic distance d_{ij} is approximated by the shortest path of the mesh edges from x_i to x_j , which is computed by the Floyd-Warshall algorithm. If the distance is smaller than the threshold D , j is a member of $\Omega(i)$. The complexity of the algorithm is $O(N^3)$, but we can reduce the computational cost by omitting the computation if the distance between two vertices is larger than the threshold D , since we only need the neighboring vertices within D .

3.4. Weighting Schemes

The weights γ_{ij} and λ_{ij} are determined for discarding an incorrect correspondence and adjusting the strength of the regularization term, respectively. Initially, both weights depend on the distance d_{ij} between x_i and x_j . The weight w_{neigh} is based on the Gaussian distribution as follows:

$$w_{\text{neigh}} = \exp(-d_{ij}^2/2D^2). \quad (11)$$

Weight γ_{ij} for the data term depends on the distance d_{xy} between x_i and y_i to remove an outlier from the correspondence. Using the threshold D_{xy} , weight w_{xy} is defined based on the Tukey weight function:

$$w_{xy} = \begin{cases} (1 - (d_{xy}/D_{xy})^2)^2 & |d_{xy}| < D_{xy} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

The weight for the data term is given by

$$\gamma_{ij} = w_{\text{neigh}} w_{xy}. \quad (13)$$

With regards weight λ_{ij} for the regularization term, we use a weight that depends on the direction with respect to the vector $x_j - x_i$. If the surface of an object is oblique and

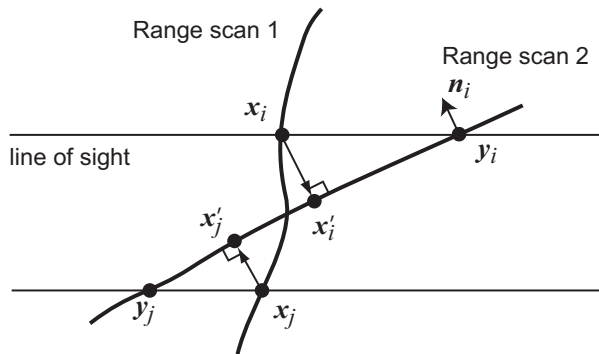


Figure 2. Where the surface of an object is oblique and nearly parallel to the line of sight, the vertices after deformation based on the correspondences are x'_i and x'_j , which are much closer to each other than vertices x_i and x_j . The deforming mesh model tends to shrink due to the definition of the data term defined in Eq.1.

nearly parallel to the line of sight as shown in Fig. 2, the deforming mesh model tends to shrink due the definition of the data term defined in Eq.1. The vertices after deformation based on the correspondences are x'_i and x'_j , which are much closer to each other than vertices x_i and x_j . Therefore, we impose a large weight along the vector $x_j - x_i$ to avoid shrinkage.

This affects the translation elements in Eq.2. If the 4-th column vector inside the Frobenius norm is τ , the coordinate is rotated with respect to the vector $v_1 = \frac{x_j - x_i}{\|x_j - x_i\|}$ by

$$M\tau = [v_1 \ v_2 \ v_3]^T \tau, \quad (14)$$

where v_2 and v_3 are unit vectors perpendicular to v_1 as shown in Fig. 3. Instead of evaluating the error $\|\tau\|^2$, the weighted error is given by

$$\|W_\lambda M\tau\|^2, \quad (15)$$

where $W_\lambda = \text{diag}(w_{\lambda 1}, w_{\lambda 2}, w_{\lambda 3})$. We set $w_{\lambda 1} = 10$ and $w_{\lambda 2} = w_{\lambda 3} = 1$ in the experiments to avoid shrinkage. Finally, the weight vector λ_{ij} of the regularization term is given by

$$\lambda_{ij} = w_0 w_{\text{neigh}} [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ w_{\lambda 1} \ w_{\lambda 2} \ w_{\lambda 3}]^T, \quad (16)$$

where w_0 is the weight relative to the data term and $w_0 = 1$ in this paper.

4. Experiments

In the experiments, we scaled the models so that the width of the bounding box is 1.0 in solving the linear equation, and fixed $w_0 = 1$. Since the object sizes are about 100.0~200.0 in the experiments, we chose $D = 5.0$ and $D_{xy} = 10.0$, which were also scaled to build the equation. We used an Intel Core i7 940 processor, with 4 CPU cores with hyper-threading and 12GB memory. In this paper, a range scan sequence is captured by a fast range finder [23].

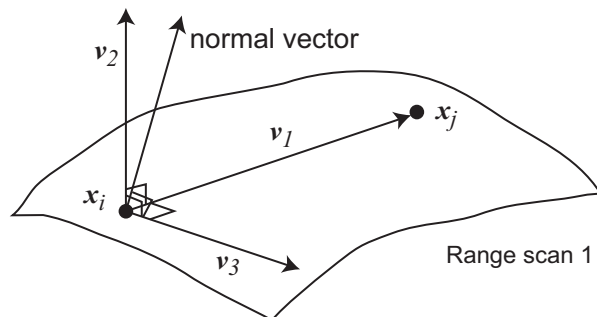


Figure 3. The weight for the translation elements is computed in the coordinate system with respect to the vector $v_1 = x_j - x_i$. v_3 is perpendicular to the normal vector at x_i .

4.1. Simulation with an artificial model

We created an artificial mesh model of a planar object, which is slightly bent in the middle, as illustrated by the red model in Fig. 4. Next, we shifted the model 10.0 units along the X-axis, while keeping the width of the model as 100.0. The resulting model is illustrated by the blue model in Fig. 4. The line of sight is along the Z-axis. In this simulation, the red model is registered to the blue one.

We compared the proposed method with a deformable registration method without convolution. The data term for this method is given by

$$E_d = \sum_i \gamma_i (n_i \cdot (T_i x_i - y_i))^2, \quad (17)$$

while the regularization term is the same as for the proposed method.

Fig. 5 shows the motion vector for each vertex as estimated by the two methods after four iterations. 'x' indicates the vertex position in the red model. The motion vector created by the proposed method is depicted by the red line, while the green line gives the result for the method without convolution. The RMS errors are 0.92 in the proposed method and 7.48 in the method without convolution. The only local shape feature in this model is the crease in the middle. The proposed method successfully found an accurate correspondence by using the local feature; however, the correct correspondence was not found by the method without convolution because the local shape was not taken into consideration.

4.2. Simulation with a range scan

Next, we manually deformed a range scan and tested the registration between two range scans before and after deformation. The red model in Fig. 6 is the original range scan, which is registered to the deformed model depicted by the blue model. Since the motion vector is relatively small, this situation can be regarded as the initial registration roughly estimated by matching sparse features.

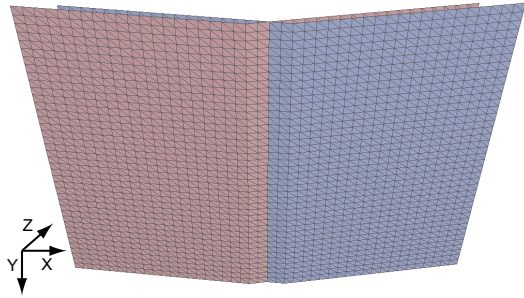


Figure 4. The red model is an artificial planar model that is slightly bent in the middle. The width of the model is 100.0. The blue model is the corresponding one after shifting 10.0 units along the X-axis.

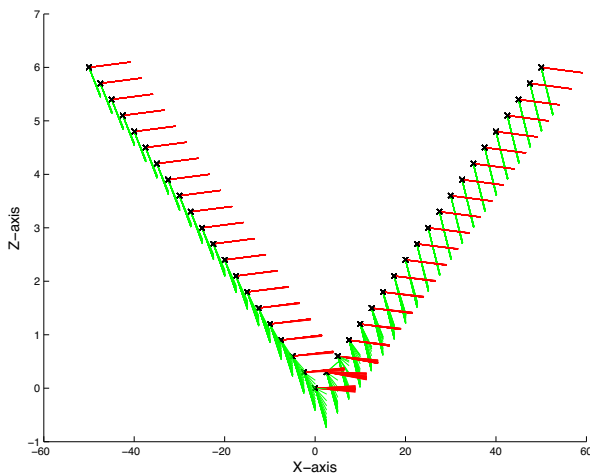


Figure 5. The motion vector for each vertex is estimated by the two methods. 'x' indicates the vertex position in the red model in Fig. 4. The motion vector created by the proposed method is depicted by the red line, while the green line gives the result by the method without convolution.

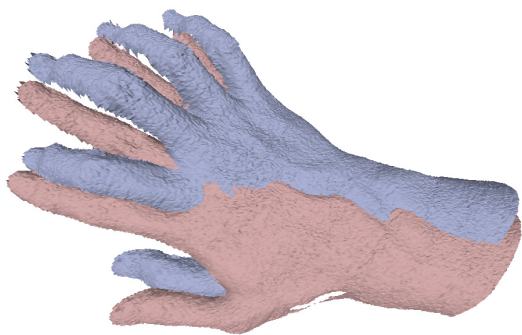
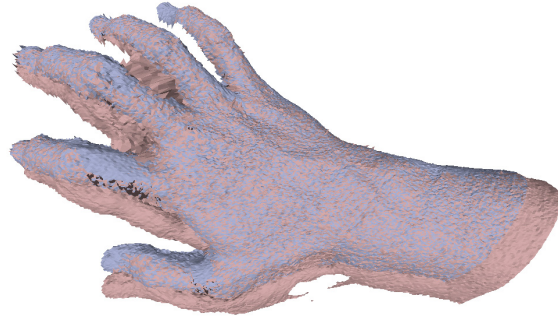


Figure 6. The red model is the original range scan, which is registered to the deformed model depicted by the blue model.

We compared the proposed method with the method without convolution. The registered results after 30 iterations by the two methods are shown in Fig. 7. The red



(a) the proposed method



(b) method without convolution

Figure 7. The registered results after 30 iterations by the two methods: the red models in (a) and (b) are the results by the proposed method and the method without convolution, respectively. The blue model shows the target of the registration.

models in (a) and (b) are the results by the proposed method and the method without convolution, respectively. The proposed method succeeded in registering all fingers, while the method without convolution failed to fit them.

Fig. 8 shows the RMS error of the position of the vertices with respect to the iteration. Since the deformation was done manually, the ground truth of the correspondence was known. 'o' indicates the results of the proposed method, while 'x' denotes the results of the method without convolution. In this experiment, the unit of distance is the millimeter. The RMS error of the proposed method was reduced to less than 1mm. However, the error of the method without convolution was not reduced. This shows that although the method without convolution minimized the distance of the corresponding points, it was a local minimum and the method was unable to find the correct correspondence.

A problem with the proposed method is its slow convergence due to the relatively large number of parameters. Consequently, many iterations are needed if the initial registration is not good. This is one of the issues to be considered in future work.

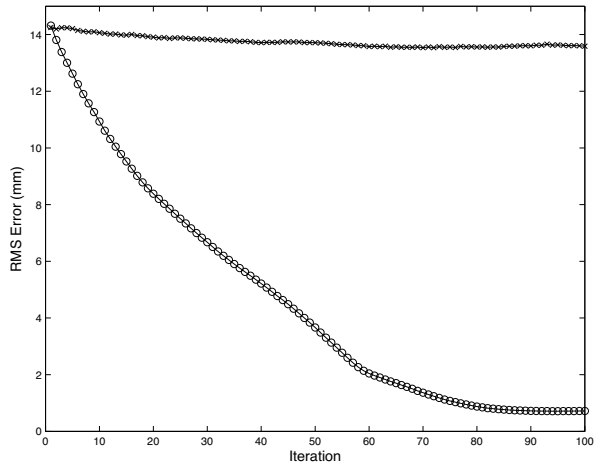


Figure 8. The RMS error of the position of the vertices with respect to the iteration: 'o' indicates the results of the proposed method, while 'x' denotes the results of the method without convolution. The unit of distance is the millimeter in this experiment.

4.3. Registration with a range scan sequence

Finally, we tested the proposed method with a range scan sequence of a moving hand. In this test, we registered the first frame to the other frames sequentially. By assuming that the motion of the hand is slow, the registered result of a frame is used as the initial registration of the next frame. The number of frames is 300. Since the first frame has 29,834 vertices, the number of parameters is 179,004. The number of correspondences in each iteration is 2,031,564 if correspondences exist for all $j \in \Omega(i)$. Thus, the average number of members of $\Omega(i)$ is 68.1. In our implementation, the maximum size of memory used is about 4.5GB. The computational time for each iteration is about 190 seconds. Due to the time limitation, the number of iterations for each frame is four in this experiment.

Fig. 9 shows the frames of the registration results. The top figure is the first frame to be registered. In the other three figures, the red models are the deformed models from the first frame while the blue ones are the target range scans. The proposed method successfully registered the range scans for almost all the frames even with a large deformation of the fingers. In some frames, the convergence was imperfect due to insufficient iterations, for example, the finger tips in frame 50.

Fig. 10 shows another result of registering the first frame (a) to the rest of the sequence. (b), (c) and (d) are three of 525 frames in the sequence. In the similar manner, the red models are the deformed models from the first frame while the blue ones are the target range scans.

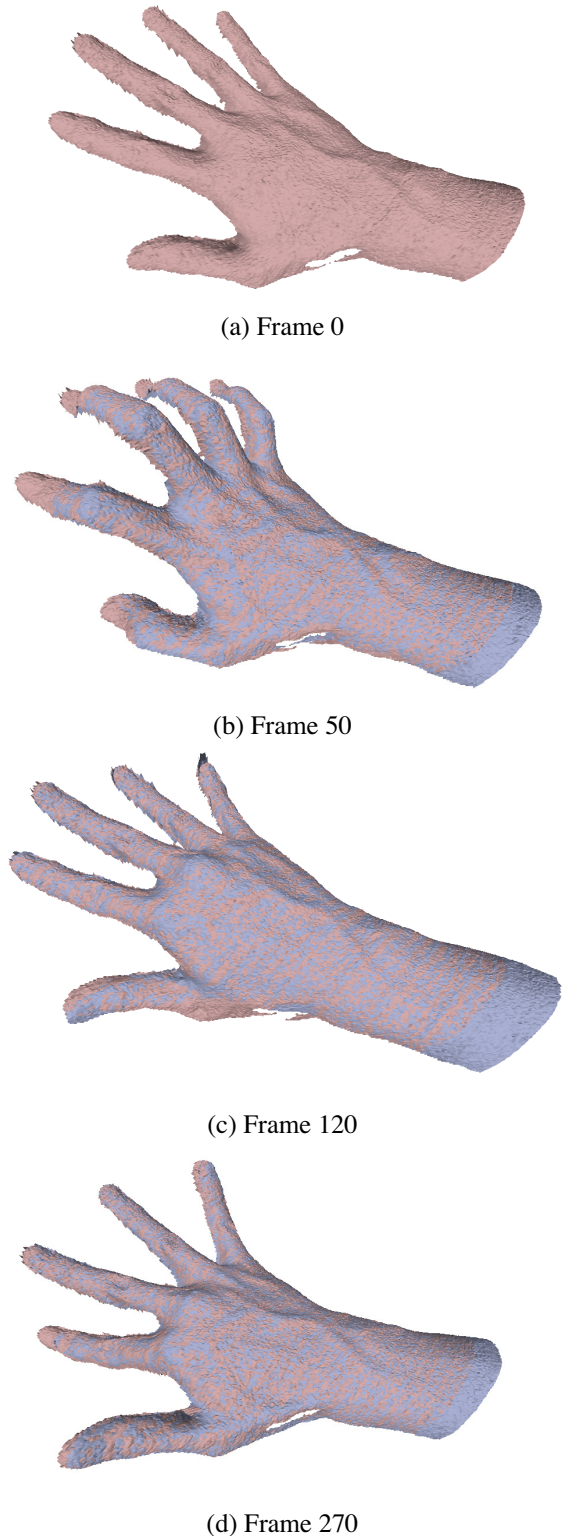


Figure 9. The top figure shows the first frame to be registered. In the other three figures, the red models are the deformed models from the first frame, while the blue ones are the target range scans.

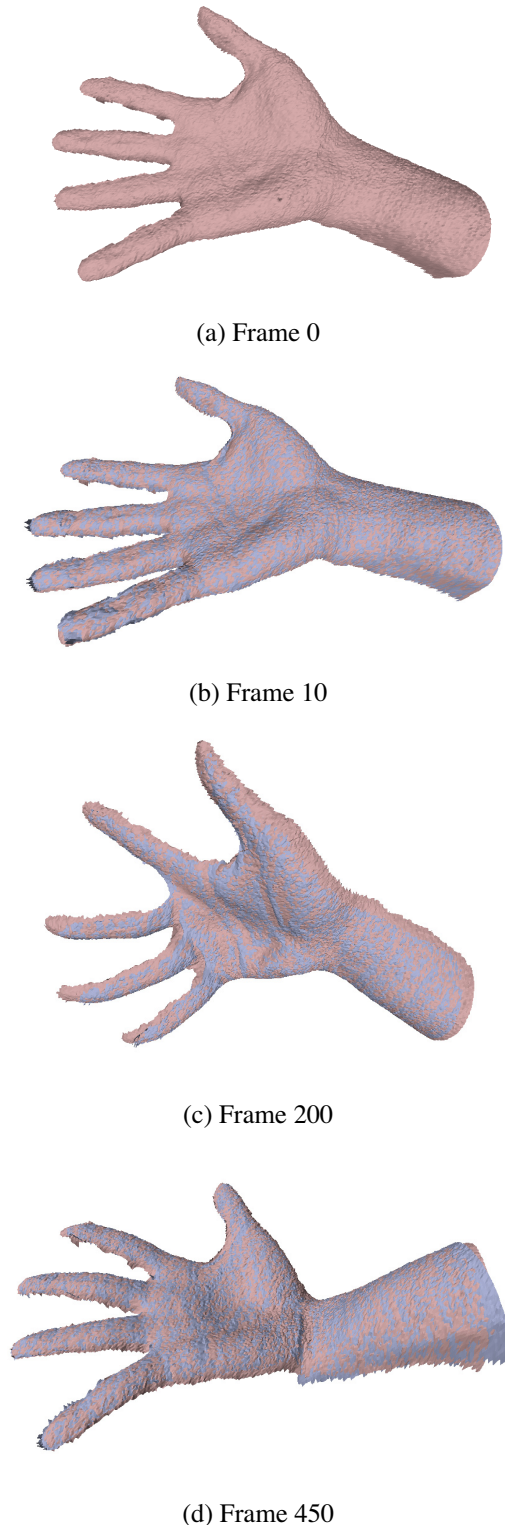


Figure 10. Another result of registering the first frame to the rest of the sequence: the red models are the deformed models from the first frame, while the blue ones are the target range scans.

5. Conclusion

In this paper, we have presented a method for elastic registration of deformable objects. To introduce the local shape around each vertex into the ICP framework, we have proposed an elastic convolved ICP that minimizes the convolved error between the corresponding points. In other words, the proposed method is a vertex-wise ICP with a smoothness constraint. Utilizing the dense local shape contributes to improving the accuracy of registration as shown in the experiments. On the other hand, the proposed method has certain limitations. For example, the correspondence of vertices may not be unique if the registration of the local shape is ambiguous. In such cases, texture information on the surface is necessary for finding the correct correspondence. In future work, we aim to combine shape and texture information to improve the accuracy and robustness of registration.

Appendix

The elements of B_i and \mathbf{b}_i in Eq.9 are given by Eq.18, where T_{uv} is the element at the u -th row and v -th column of $T_i^{(k)}$, and v_{uv} is the element at the u -th row and v -th column of M .

References

- [1] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. on Graphics (TOG)*, 22(3):587–594, 2003. 1, 2
- [2] D. Anguelov, D. Koller, P. Srinivasan, S. Thrun, H.-C. Pang, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems (NIPS 2004)*, Vancouver, Canada, 2004. 1
- [3] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(2):239–256, Feb 1992. 2
- [4] B. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-D scans. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), Aug. 2007. 1
- [5] C. Chua and R. Jarvis. 3-d free-form surface registration and object recognition. *Int'l Jour. Computer Vision*, 17(1):77–99, 1996. 1
- [6] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Symposium on Geometry Processing*, pages 197–206, 2005. 1
- [7] D. Hähnel, S. Thrun, and W. Burgard. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Aca-pulco, Mexico, 2003. IJCAI. 1
- [8] L. Ikemoto, N. Gelfand, and M. Levoy. A hierarchical method for aligning warped meshes. In *Proc. International Conference on 3D Imaging and Modeling*, 2003. 1

$$B_i = \begin{bmatrix} 0 & T_{31} & -T_{21} & 0 & 0 & 0 \\ -T_{31} & 0 & T_{11} & 0 & 0 & 0 \\ T_{21} & -T_{11} & 0 & 0 & 0 & 0 \\ 0 & T_{32} & -T_{22} & 0 & 0 & 0 \\ -T_{32} & 0 & T_{12} & 0 & 0 & 0 \\ T_{22} & -T_{12} & 0 & 0 & 0 & 0 \\ 0 & T_{33} & -T_{23} & 0 & 0 & 0 \\ -T_{33} & 0 & T_{13} & 0 & 0 & 0 \\ T_{23} & -T_{13} & 0 & 0 & 0 & 0 \\ -v_{12}T_{34} + v_{13}T_{24} & v_{11}T_{34} - v_{13}T_{14} & -v_{11}T_{24} + v_{12}T_{14} & v_{11} & v_{12} & v_{13} \\ -v_{22}T_{34} + v_{23}T_{24} & v_{21}T_{34} - v_{23}T_{14} & -v_{21}T_{24} + v_{22}T_{14} & v_{21} & v_{22} & v_{23} \\ -v_{32}T_{34} + v_{33}T_{24} & v_{31}T_{34} - v_{33}T_{14} & -v_{31}T_{24} + v_{32}T_{14} & v_{31} & v_{32} & v_{33} \end{bmatrix} \quad \mathbf{b}_i = \begin{bmatrix} T_{11} \\ T_{21} \\ T_{31} \\ T_{12} \\ T_{22} \\ T_{32} \\ T_{13} \\ T_{23} \\ T_{33} \\ v_{11}T_{14} + v_{12}T_{24} + v_{13}T_{34} \\ v_{21}T_{14} + v_{22}T_{24} + v_{23}T_{34} \\ v_{32}T_{24} + v_{33}T_{34} + v_{31}T_{14} \end{bmatrix} \quad (18)$$

- [9] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999. 1
- [10] H. Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi. Dynamic scene shape reconstruction using a single structured light pattern. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008*, pages 1–8, June 23–28 2008. 1
- [11] T. Masuda. Multiple range image registration by matching local log-polar range images. In *The 7th Asian Conference on Computer Vision*, pages 948–957, Hyderabad, India, January 2006. 1
- [12] N. J. Mitra, S. Flory, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic geometry registration. In *Symposium on Geometry Processing*, pages 173–182, 2007. 1
- [13] S. Narasimhan, S. Koppal, and S. Yamazaki. Temporal dithering of illumination for fast active vision. In *European Conference on Computer Vision*, volume 4, pages 830–844, October 2008. 1
- [14] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proc. Int. Conf. on Shape Modeling and Application*, pages 130–139, Mar 1997. 2
- [15] T. Oishi, A. Nakazawa, R. Kurazume, and K. Ikeuchi. Fast simultaneous alignment of multiple range images using index images. In *Proc. The 5th International Conference on 3-D Digital Imaging and Modeling (3DIM 2005)*, pages 476–483, 2005. 2
- [16] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *Proc. SIGGRAPH*, pages 438–446, 2002. 1
- [17] S. Rusinkiewicz and M. Levoy. Efficient variants of the ipc algorithm. In *Proc. Int. Conf 3-D Digital Imaging and Modeling*, pages 145–152, 2001. 2
- [18] R. Sagawa, N. Osawa, and Y. Yagi. Deformable registration of textured range images by using texture and shape features. In *Proc. The 6th International Conference on 3-D Digital Imaging and Modeling*, pages 65–72, Montreal, Canada, August 21–23 2007. 1
- [19] C. Shelton. Morphable surface models. *Int. J. Comput. Vision*, 38(1):75–91, 2000. 1
- [20] F. Stein and G. Medioni. Structural indexing: efficient 3-d object recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992. 1
- [21] R. Sumner and J. Popović. Deformation transfer for triangle meshes. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 399–405, 2004. 1
- [22] S. Toledo. TAUCS: A library of sparse linear solvers. Available online at <http://www.tau.ac.il/~stoledo/taucs/>, 2003. 3
- [23] T. Weise, B. Leibe, and L. Van Gool. Fast 3d scanning with automatic motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pages 1–8, June 2007. 1, 4