Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2004)
R. Boulic, D. K. Pai (Editors)

Modeling Deformable Human Hands from Medical Images

Tsuneya Kurihara[1] and Natsuki Miyata[2]

[1]Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan
[2] Digital Human Research Center, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract

This paper presents a new method for constructing an example-based deformable human hand model from medical images. Realistic animation of human hands requires good estimates of the joint structure and properly weighted skeleton-driven surface deformation. For this purpose, we propose a method based on medical images of hands in several poses. Our method consists of the following 3 steps: First, using the measured bone shapes, we estimate the link structure (joint rotation centers) and the joint angles of each scan. Second, we construct a mutually consistent polygonal mesh of all the scans. For this purpose, a polygonal mesh of one pose, the base mesh, is deformed using skeletal subspace deformation, and then fitted interactively to the measured meshes from the other scans. Finally, the hand is deformed using a weighted pose space deformation. We demonstrate results of deformable hand models consisting of 100,000 triangle meshes derived from CT scans.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations, I.3.7 [Computer Graphics]: Animation

1. Introduction

The human hand is very important for realistic human character animation. Human hands are skillful and expressive, and are used for communication, grasping, and many other activities requiring dexterous precision. Good graphical models of the human hand are thus very useful for a variety of applications including movies and special effects, user interfaces, games and interactive virtual environments. However, modeling of the hand is extremely difficult due to the complexity of its shape, joint structure, and deformation.

For any realistic model of the human hand, quite detailed geometry is required. Recently, the detailed geometry of the hand can be obtained using 3D measurements [HDD*92, YCC99]. However, obtaining accurate skeleton-driven deformations of these models is not easy. Several methods have been proposed for the deformation of the hand. In all of these methods, the location of the center of rotation of each joint is critical to generating natural deformations. Because the center of rotation is difficult to estimate from the skin surface of the hand, it is specified interactively in most cases, which is a time-consuming and tedious task. When the center of the rotation is not adequate, the deformation will be unnatural. While the center of rotation can be estimated from motion capture data [OBBH00], this technique cannot be applied to the hand because of large skin deformation.

In addition, skeleton driven deformation is essential for realistic human hand animation. For this purpose, several techniques have been proposed such as SSD (Skeletal Subspace Deformation) [MTLT88], deformation using finite element methods [GTT89], anatomy-based methods using bones and pseudo muscles [AHS03]. However, these techniques require a lot of time for parameter tuning in order to achieve realistic deformation.

In contrast, example-based techniques have been proposed for realizing natural deformation of skin [LCF00, SIC01, WP02]. With example-based approaches, the skin surfaces of several poses are represented with mutually consistent meshes, and subsequently interpolated in pose space. While the deformation is therefore very realistic if appropriate data is prepared, the animators must provide mutually consistent meshes.

An example-based method using range scan data was proposed for deformation of the upper body [ACP02]. In this method, the centers of rotation are estimated from markers. This cannot be applied to the hand, however, because its skin deformation is very large. In addition, all the scans are pa-

rameterized by using displaced subdivision surfaces. It may therefore be difficult to construct consistent meshes with detailed measured skin surfaces.

Kry, James, and Pai proposed an example-based deformation method suited to commercial graphics hardware [KJP02]. Instead of using all the displacements for key poses, their method used precomputed principal components of deformation influences on individual joints. In their example results, many sample skin surfaces of the hand were generated using finite-element analyses, but the sample skin surfaces generated by simulation were not as detailed as those of the actual hand.

As discussed above, two aspects are very important in realistic deformation of the hand: precise estimation of the rotation centers, and example-based skin deformation with detailed measured surfaces. Using multiple medical images of the hand, we can obtain precise bone shape and skin surface measurements simultaneously. In addition, we can estimate the center of rotation for each joint if we compare the bone shapes in several poses. We therefore present a new method for model extraction and skeleton driven deformation of the hand using medical images. In this three-step method, we first derive the centers of rotation and poses from bone shapes, then transform the skin surfaces of all poses into mutually consistent meshes, and finally implement skeleton-driven deformation by using weighted pose space deformation.

The rest of this paper is organized as follows. Section 2 describes the data capture step using medical images. Section 3 describes our technique to estimate the rotation centers of the joints by comparing the bone shapes in several poses. Section 4 presents our techniques for producing mutually consistent skin meshes for all poses. Section 5 describes the skeleton-driven deformation method using resulting data, and presents the results of our experiments with the proposed techniques. Section 6 offers directions for future research.

2. Measurement of Hand using CT

We measured a real human hand in five different poses by using computed tomography (CT). Figure 1 shows an example of a CT image. In our measurements, the pixel size was 0.468×0.468 [mm], the slice thickness was 0.3 [mm], and the number of slices was about 750. We obtained $512 \times 512 \times 750$ voxels for each pose. Bone and skin surfaces were generated as isosurfaces using the marching cubes algorithm [LC87] (see Figure 2). The number of triangles was about 460,000 for bone and about 800,000 for the skin surface.

We removed unnecessary small bones interactively. While there are eight carpal bones around the wrist, only the biggest (capitate) was extracted because their movement is complex and difficult to handle in estimating the joint center (Figure 3). In addition, we extracted only the radius and ignored the ulna in the forearm for the sake of simplicity.

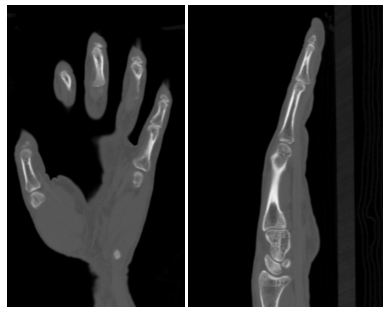


Figure 1: *Example CT image.*

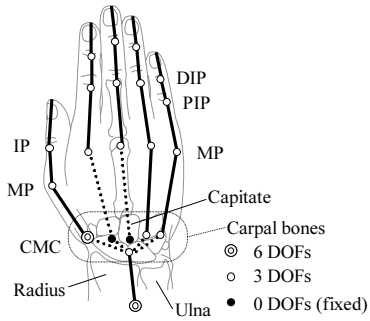


Figure 3: *Hand link model.*

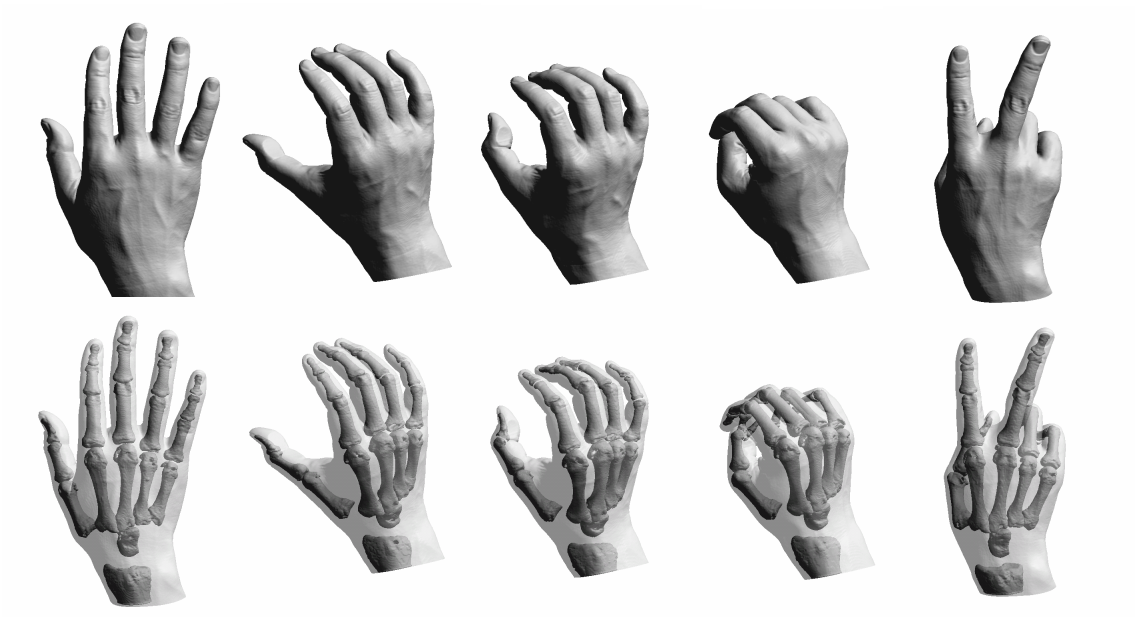
Note that CT may be harmful because of X-ray exposure. MRI can be used instead with the proposed method because the bone shape and the skin shape can be measured simultaneously. We used CT in our experiment because high-resolution images can be obtained, the measurement time is shorter, and the bone shape can be extracted more easily than with MRI.

3. Estimation of Joint Centers

Figure 3 shows our hand link model. It consists of 19 joints, two with 6 degrees of freedom (DOFs) and 17 with 3 DOFs for simplicity. Note that the CMC joint of the thumb has 6 DOFs because of its complex movement.

The center of rotation for each joint is estimated by comparing bone shapes in multiple poses. We estimate the center of rotation within the reference frame of a "base pose," which is a relaxed natural posture (see Figure 2(a)). Figure 4 shows the process by which the rotation center is estimated.

We first align each bone segment $k(1 \leq k \leq n_{\text{link}})$ in pose $i(1 \leq i \leq n_{\text{pose}})$ with that in the base pose by using the ICP (Iterative Closest Point) algorithm [BM92]. The ICP is a method to align two objects with rigid transformation. With



(a) Pose 1 **(b) Pose 2** **(c) Pose 3** **(d) Pose 4** **(e) Pose 5**

Figure 2: *Skin shape and bone shape reconstructed from CT images.*

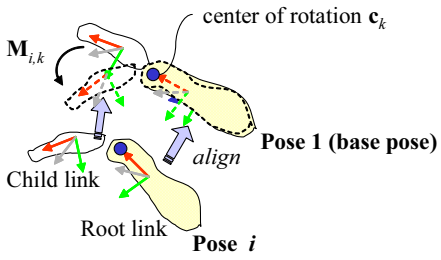


Figure 4: *Calculation of rotation center of joint.*

this alignment, we then calculate transformation matrix $\mathbf{M}_{i,k}$ that represents the local transformation of child link k in pose i if the parent link is transformed to the base pose. $\mathbf{M}_{i,k}$ is a 4×4 matrix that represents rotation and translation. Rotation center $\mathbf{c}_k$ of child link k should not move with transformation $\mathbf{M}_{i,k}$. We estimate the center of rotation where the error function

$$E = \sum_{i=1}^{n_{\text{pose}}} |\mathbf{M}_{i,k}\mathbf{c}_k - \mathbf{c}_k|^2 \qquad (1)$$

takes a minimum value. We used the Davidon Fletcher Powell method [PFTV92] for the minimization. Figure 5 shows the estimated centers of rotation. The square root of the mean error in Equation 1 for all the joints was 0.91[mm] except for the CMC joint of the thumb that has 6 DOFs. We also

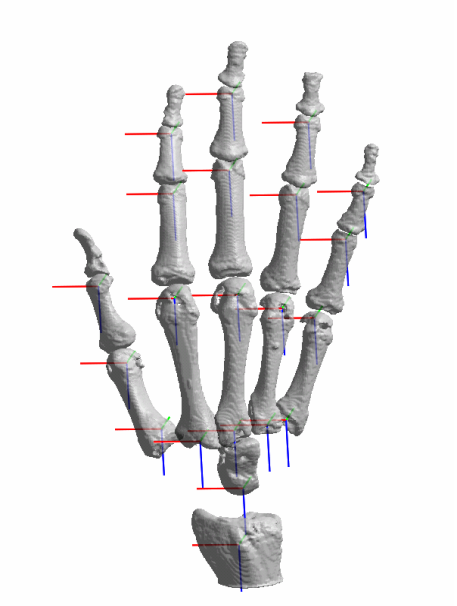
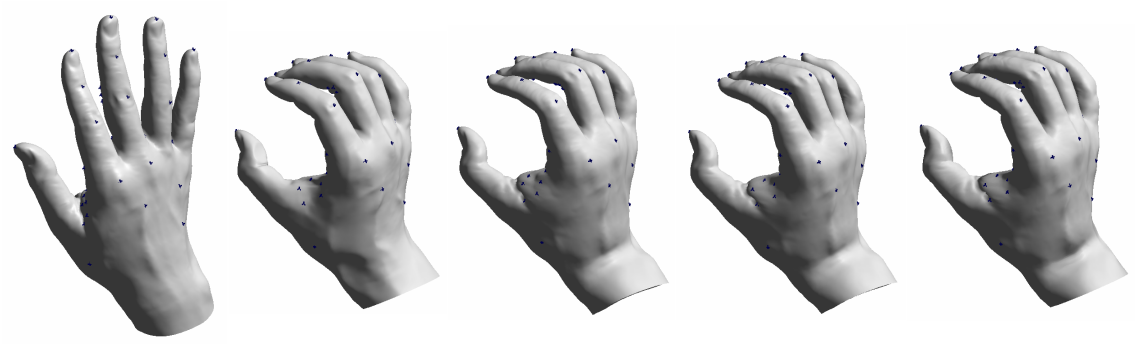


Figure 5: *Estimated rotation center of joint.*

estimate the rotation angles for each joint and pose from the transformation matrix $\mathbf{M}_{i,k}$ by solving Euler angles from transformation matrix.



(a) Base mesh $\mathbf{S}_0$ (b) Measured mesh $\mathbf{T}_i$ (c) Mesh $\mathbf{S}_i^s$ deformed using SSD (d) Mesh $\mathbf{S}_i^r$ deformed using RBF (e) Fitted mesh $\mathbf{S}_i^f$

Figure 6: *Fitting process.*

4. Skin Surface Fitting

The actual measured skin surface of the hand consists of about 800,000 triangles, and handling all of them would be excessively time-consuming. We therefore reduced the number of triangles to about 100,000 by using standard polygon reduction techniques. The parameterization of the measured (reduced) mesh for each scan differs from those of the others, and it would be difficult to deform them by morphing. We therefore generate mutually consistent meshes for all samples. For this purpose, we assume one skin mesh is a "base mesh" and fit this base mesh to the other scans. In our experiment, Pose 1 of Figure 2 was used as the base mesh. Hereinafter, we will refer to the base mesh as S_0, and to the measured mesh of pose i as T_i. We transform base mesh S_0 into measured mesh T_i as follows:

1. We make a first approximation for each pose by using Skeletal Subspace Deformation to deform S_0 into S_i^s.
2. We improve the accuracy of approximation by using RBF-based deformation with interactively specified feature points to deform approximate mesh S_i^s into S_i^r.
3. We generate the final mesh S_i^f by fitting the resulting mesh S_i^r to measured mesh T_i .

The details of this procedure are explained in the following subsections.

4.1. Approximation using Skeletal Subspace Deformation

As it would be hard to transform the base mesh to the measured mesh all at once because the shapes of the measured meshes differ largely for each pose (see Figures 6(a), 6(b)), we make a first approximation by using SSD (Skeletal Subspace Deformation) [LCF00] to deform base mesh S_0 into approximate mesh S_i^s. SSD can be used because the rotation angles and centers of rotation were determined in the first step.

With SSD, deformed vertex position $\overline{\mathbf{p}}_j$ is computed as the weighted sum of transformed vertex positions of related links:

$$\overline{\mathbf{p}}_j = \sum_{k=1}^{n_{\text{link}}} w_{j,k} \mathbf{L}_k \mathbf{p}_j, \tag{2}$$

where $\mathbf{p}_j$ is the original position of vertex j, $\mathbf{L}_k$ is the transformation matrix of Link k, n_{link} is the number of links, $w_{j,k}$ is the weight, and

$$\sum_{k=1}^{n_{\text{link}}} w_{j,k} = 1. \tag{3}$$

In our experiment, we specified weight $w_{j,k}$ interactively using commercial software MAYA 4.5. Figure 6(c) shows resulting mesh S_i^s obtained with SSD. Although SSD is not anatomy based, the resulting meshes approximate measured mesh T_i.

4.2. Interactive Deformation using Feature Points

To improve the accuracy of the mesh, we employ feature-points-based deformation to deform approximate mesh S_i^s. We first interactively specify feature points on both approximate mesh S_i^s and measured mesh T_i. Figures 6(b) and 6(c) show the feature points. Because the approximate mesh produced by SSD resembles the measured mesh, it is not difficult to specify the feature points. In addition, we can specify the feature points using shape details, such as wrinkles. About 70 to 150 feature points were specified for each pose in our experiment.

We use an interpolation method with RBF (Radial Basis Functions) [Nie93, PSS02] for the deformation, which is an effective method of scattered data interpolation. Let $\mathbf{r}_j$ be the feature point position on measured mesh T_i and let $\mathbf{q}_j$ be the corresponding feature point position on approximated mesh

S_i^s. Deformed vertex position $\mathbf{v}'$ is calculated by RBF as

$$\mathbf{v}' = \mathbf{v} + \sum_{j=1}^{n_{\text{cp}}} c_j \phi(|\mathbf{v} - \mathbf{q}_j|)(\mathbf{r}_j - \mathbf{q}_j), \tag{4}$$

where n_{cp} is the number of feature points and $\phi(r)$ is a radially symmetric basis function. We chose to use $\phi(r) = e^{-r/D}$ as the basis function after the experiments. D is a user-specified constant coefficient, and we used $D = 1600$ [mm] for our experiment. Coefficient c_j is calculated by solving

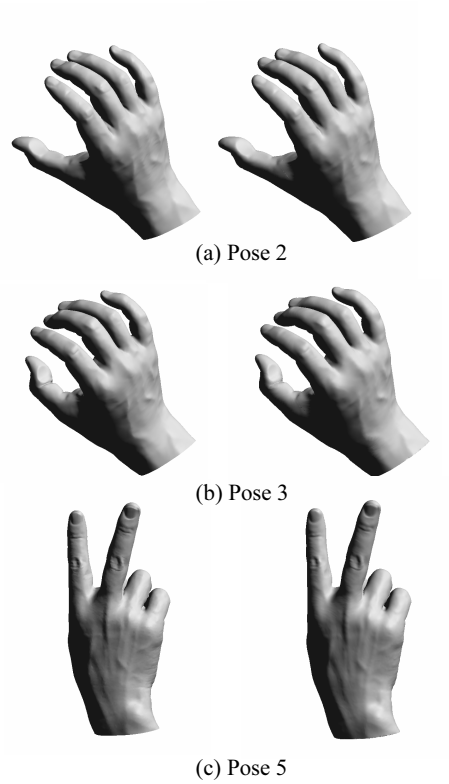
$$\mathbf{r}_j - \mathbf{q}_j = \sum_{i=1}^{n_{\text{cp}}} c_i \phi(|\mathbf{q}_j - \mathbf{q}_i|)(\mathbf{r}_i - \mathbf{q}_i). \tag{5}$$

With deformation using RBF, approximated mesh S_i^s is deformed into the more accurate mesh S_i^r. Figure 6(d) shows the resulting deformed mesh.

4.3. Fitting

After approximated meshes S_i^r of measured mesh T_i have been obtained, we fit mesh S_i^r to measured mesh T_i to generate final mesh S_i^f. For fitting, we simply find the nearest point on measured mesh T_i for each vertex on S_i^r, and move it to the nearest point. Figure 6(e) shows the fitting results for Pose 3, and Figure 7 shows the fitting results for Poses 2, 3, and 5. This simple fitting technique cannot be applied to Pose 4 in Figure 2 because some parts of the skin surface are missing, where the two fingers touch each other. A more sophisticated fitting technique must be used in such cases. A fitting technique using smoothness [ACP03] may work well although we have not tested it.

It took about 10 minutes to specify the feature points for each pose. RBF fitting took about 1.0 [sec] and final fitting took about 9.0 [sec] for each pose with a Pentium IV PC with a 2.2-GHz CPU. To find the nearest point quickly, we used PQP, a Proximity Query Package developed by the UNC Research Group on Geometry, Physically-Based Simulation, and Applications [LGLM99].



(a) Pose 2

(b) Pose 3

(c) Pose 5

Figure 7: *Fitting results. Left is measured mesh, right is fitted mesh.*

5. Skeleton-Driven Hand Deformation

5.1. Pose Space Deformation

Once we have the rotation centers for all joints, the rotation angles of joints, and consistent skin surfaces for all poses, skeleton-driven skin deformation is accomplished by interpolating skin surface in the pose space. We use the Pose Space Deformation method [LCF00, SIC01] for this interpolation. Pose space deformation is a hybrid approach that combines SSD and morphing (Figure 8). Sample surfaces of various poses are deformed into the "base pose" with inverse SSD, and the resulting meshes are then morphed and deformed with SSD.

Let $\mathbf{v}_{i,j}$ be the position of vertex j of sample i and $w_{j,k}$ be the weight value of vertex j about link k. Sample surface i is first transformed into its "base pose":

$$\mathbf{v}_{i,j}^0 = (\sum_{k=1}^{n_{\text{link}}} w_{j,k} \mathbf{L}_{i,k})^{-1} \mathbf{v}_{i,j}, \tag{6}$$

where $\mathbf{v}_{i,j}^0 =$ is the position of vertex j of sample i in its base pose and $\mathbf{L}_{i,k}$ is the transformation matrix of link k of sample i. Let s_i be the weight value for the interpolation of each sample with $\sum_{i=1}^{n_{\text{pose}}} s_i = 1$.

Then, each sample surface in the base pose is interpolated by using a morphing method:

$$\mathbf{u}_j^0 = \sum_{i=1}^{n_{\text{pose}}} s_i \mathbf{v}_{i,j}^0, \tag{7}$$

where $\mathbf{u}_j^0$ is the interpolated vertex position. Finally, the morphed surface is deformed with SSD:

$$\mathbf{u}_j = \sum_{k=1}^{n_{\text{link}}} w_{j,k} \mathbf{L}_k \mathbf{u}_j^0, \tag{8}$$

where $\mathbf{u}_j$ is the vertex position of the resulting deformed surface, $\mathbf{L}_k$ is the transformation matrix that is calculated by interpolating the joint angle using weight value s_i .

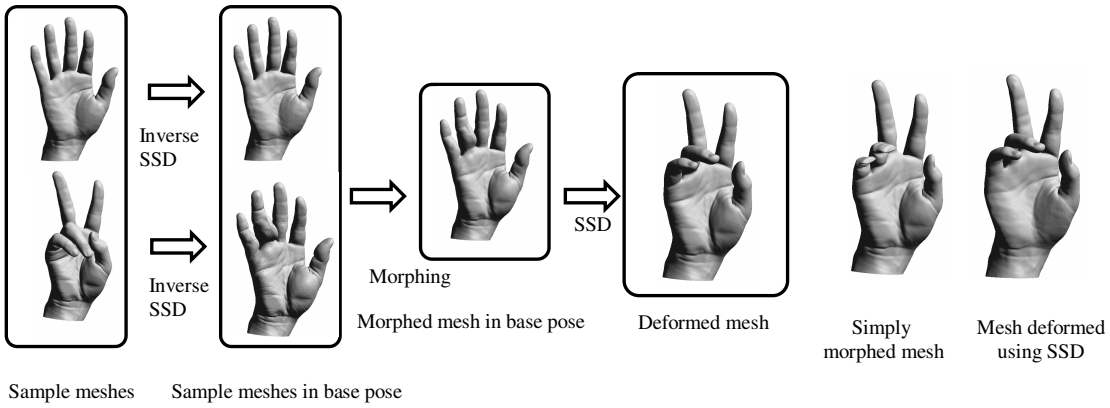


Figure 8: *Pose space deformation.*

Figure 9 shows the result of deformation. In this example, Pose 1 of Figure 2 is first transformed into Pose 5 and then into Pose 3. That is, the mesh is linearly interpolated from $s_1 = 1, s_2 = s_3 = s_5 = 0$ into $s_1 = s_2 = s_3 = 0, s_5 = 1$, then interpolated into $s_1 = s_2 = 0, s_3 = 1, s_5 = 0$. The detailed mesh is interpolated as shown in Figure 9.

5.2. Weighted Pose Space Deformation

In this subsection, we present a method for the deformation of a given arbitrary pose. With pose space deformation, shape deformation is accomplished by interpolating sample shapes, and this is done by using RBF in the pose space [SIC01]. The possible deformations therefore depend on the sample poses. Now let $\mathbf{a}_i$ be a vector that consists of n_{dof} rotation angles for pose i. Then, pose $\mathbf{a}$ in which the shape can be interpolated with sample shapes is constrained by

$$\mathbf{a} = \sum_{i=1}^{n_{\text{pose}}} s_i \mathbf{a}_i. \tag{9}$$

Hence, with the limited samples in our experiment, it is difficult to make a deformed shape of an arbitrary pose where, for example, only the index finger is extended and the other fingers are bent. For arbitrary poses, more samples are required for deformation. The hand is thought to have about 30 DOFs, so we need samples that correspond to these DOFs.

Preparing such samples is time-consuming, however, and a great deal of computational time is needed to interpolate many samples. Fortunately, deformation with poses has some independence. For example, deformation of the index finger is not affected by rotation angles of the little finger.

Next, we present an interpolation method with limited sample poses. It is an extension of pose space deformation, which we will describe here in more detail before explaining our method.

Given pose $\mathbf{a}$, we need to calculate weight $s_i(\mathbf{a})$ for each sample. These weights are subject to the following constraints:

1. At an example point, the weight for that example must be one, and all the other weights must be zero:

$$\begin{aligned} s_i(\mathbf{a}_j) &= 1 \quad (i = j) \\ s_i(\mathbf{a}_j) &= 0 \quad (i \neq j). \end{aligned} \tag{10}$$

2. The weights must always add up to one:

$$\sum_{i=1}^{n_{\text{pose}}} s_i(\mathbf{a}) = 1. \tag{11}$$

3. $s_i(\mathbf{a})$ must be continuous according to the change of $\mathbf{a}$ for smooth animation.

We first calculate $f_i(\mathbf{a})(1 \leq i \leq n_{\text{pose}})$:

$$\begin{aligned} f_i(\mathbf{a}_j) &= 1 \quad (i = j) \\ f_i(\mathbf{a}_j) &= 0 \quad (i \neq j), \end{aligned} \tag{12}$$

where $f_i(\mathbf{a})$ has the following RBF form:

$$f_i(\mathbf{a}) = \sum_{j=1}^{n_{\text{pose}}} c_{i,j} \phi(d(\mathbf{a} - \mathbf{a}_j)), \tag{13}$$

where $d(\mathbf{a} - \mathbf{a}_j)$ is the distance between pose $\mathbf{a}$ and pose $\mathbf{a}_j$, $c_{i,j}$ is a coefficient that can be solved with Equation 12. Then, $s_i(\mathbf{a})$ can be calculated with $f_i(\mathbf{a})$:

$$s_i(\mathbf{a}) = \frac{f_i(\mathbf{a})}{\sum_{j=1}^{n_{\text{pose}}} f_j(\mathbf{a})}. \tag{14}$$

Calculated weight $s_i(\mathbf{a})$ is used in the deformation of a given pose. In our experiment, the distance between pose $\mathbf{a}$ and pose $\mathbf{b}$ is defined as:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{k=1}^{n_{\text{dof}}} (a_k - b_k)^2}. \tag{15}$$

Note that weight $s_i(\mathbf{a})$ for interpolation is the same for all vertices.

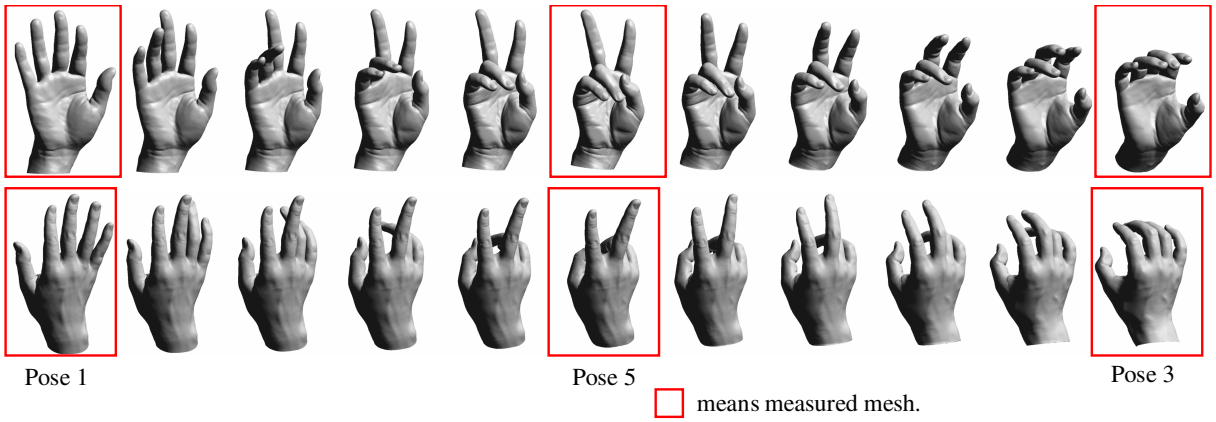


Figure 9: *Results of shape interpolation.*

Next, we will explain Weighted Pose Space Deformation (WPSD) with limited examples. The influence of links on deformation changes from vertex to vertex. Let $u_{j,k}$ be the weight coefficient of vertex j that describes the influence from the k-th component of vector $\mathbf{a}$ on vertex j. We define the distance between two poses $\mathbf{a}$ and $\mathbf{b}$ for each vertex j with this weight:

$$d_j(\mathbf{a},\mathbf{b}) = \sqrt{\sum_{k=1}^{n_{\text{dof}}} u_{j,k}(a_k - b_k)^2}. \qquad (16)$$

With this distance, we calculate weight $s_i(\mathbf{a})$ for each vertex. With this weight, we interpolate each vertex position in the base pose using Equation 7. Figure 10 shows the results for weighted pose space deformation. Natural deformation is obtained with limited samples because of the weight coefficient and the independence of deformation on joints. Figure 11 compares three methods: SSD, original pose space deformation, and weighed pose space deformation. Unnatural deformation can be seen in the little finger with SSD. Original pose space deformation has a problem with the ring finger because not enough samples were prepared. The weighted pose space deformation method results in natural deformation even with limited samples. Note that the original pose space deformation may result in natural deformation if enough samples are prepared. In our experiment, we used the same value for weighting that we used for weight $w_{j,k}$ of the SSD and obtained good results.

We can precompute coefficient $c_{i,j}$ for every vertex before WPSD deformation. However, distance d_j and weight $s_i(\mathbf{a})$ must be computed for every vertex for deformation. This causes additional computation time. The computation cost of WPSD is $O(N_{\text{pose}} \times N_{\text{vertex}})$.

The example images in Figures 9 and 10 were generated with OpenGL hardware rendering. The shape interpolation in Figure 9 took 62 [ms] and the hardware rendering took 25 [ms]. The weighted pose space deformation in Figure 10 took 157 [ms]. Deformation can be done in near real-time (6 fps) with a Pentium IV PC with a 2.2-GHz CPU, 1 GB of memory, and a Radeon 9500 Pro graphics board. No hardware acceleration was used in our experiment, but graphics hardware may be used for pose space deformation [KJP02].

6. Conclusion

We have developed techniques that enable skeleton driven deformation of a detailed hand model derived from CT images. The joint centers and joint angles of each scan are estimated by comparing the shapes of bones in several poses. Mutually consistent skin surfaces are generated by fitting a base skin surface into other skin surfaces. These joint centers, joint angles, and consistent skin surfaces are used in Weighted Pose Space Deformation to deform the hand model according to a given pose. The result is a richly detailed example-based posable hand model.

There are many ways of extending the techniques that we have described. Fitting techniques that can handle missing parts in skin (e.g., Pose 4 in Figure 2) need to be developed. Methods of calculating the weights for the SSD automatically [MG03] instead of specifying them interactively should also be developed. To produce consistent meshes for all scans, we have to specify the feature points. Since the accuracy of deformation very much depends on this, more sophisticated methods that make correspondence between meshes should be developed. One possibility is to use geometric feature correspondence. Finally, our posable model can only be used to simulate deformation of a measured subject. Modeling the human hand by examples [ACP03] could be accomplished by measuring large numbers of people's hands.

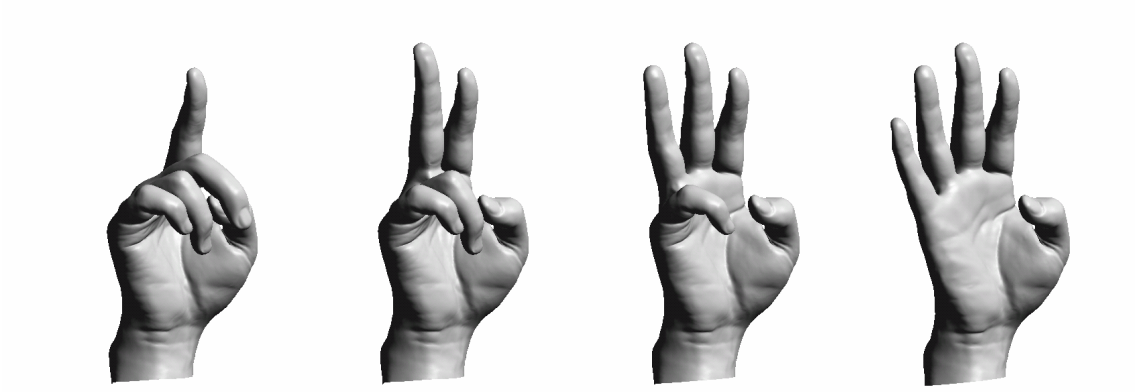


Figure 10: *Results of weighted pose space deformation.*

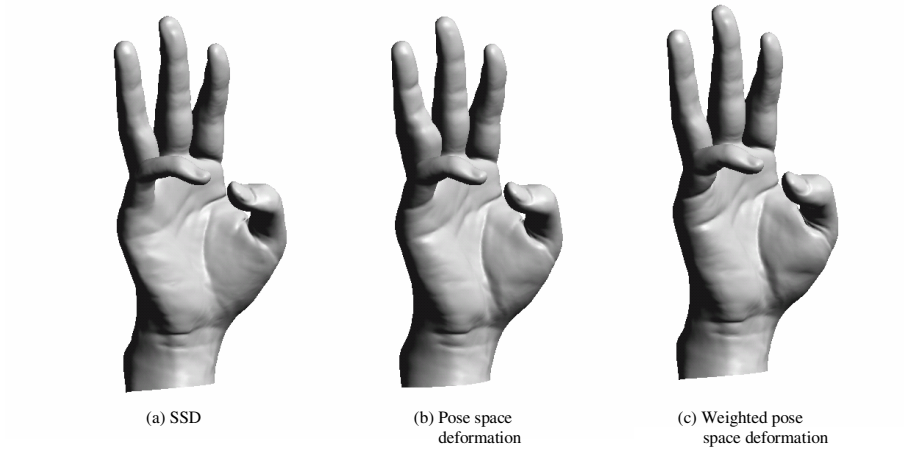


Figure 11: *Comparison of deformation methods.*

7. Acknowledgements

The authors would like to thank Assistant Prof. Yoshitaka Masutani, Image Computing and Analysis Laboratory, the University of Tokyo Hospital (UTRAD/ICAL) for developing the filters to segment the bones and extract the skin surface, and Satomi Kamojima of the University of Tokyo for extracting the bone and skin shapes.

References

[ACP02] ALLEN B., CURLESS B., POPOVIĆ Z.: Articulated body deformation from range scan data. *ACM Transactions on Graphics 21*, 3 (July 2002), 612–619.

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Transactions on Graphics 22*, 3 (July 2003), 587–594.

[AHS03] ALBRECHT I., HABER J., SEIDEL H.-P.: Construction and animation of anatomically based human hand models. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), Eurographics Association, pp. 98–109.

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2 (1992), 239–256.

[GTT89] GOURRET J.-P., THALMANN N. M., THALMANN D.: Simulation of object and human skin deformations in a grasping task. In *Computer*

Graphics (Proceedings of SIGGRAPH 89) (July 1989), vol. 23, pp. 21–30.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *Computer Graphics (Proceedings of SIGGRAPH 92)* (July 1992), vol. 26, pp. 71–78.

[KJP02] KRY P. G., JAMES D. L., PAI D. K.: Eigenskin: Real time large deformation character skinning in hardware. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 153–160.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 87)* (July 1987), vol. 21, pp. 163–169.

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of ACM SIGGRAPH 2000* (July 2000), Computer Graphics Proceedings, Annual Conference Series, pp. 165–172.

[LGLM99] LARSEN E., GOTTSCHALK S., LIN M. C., MANOCHA D.: *Fast Proximity Queries with Swept Sphere Volumes*. Tech. Rep. TR99-018, Department of Computer Science, University of N. Carolina, Chapel Hill, 1999.

[MG03] MOHR A., GLEICHER M.: Building efficient, accurate character skins from examples. *ACM Transactions on Graphics 22*, 3 (July 2003), 562–568.

[MTLT88] MAGNENAT-THALMANN N., LAPERRI'ERE R., THALMANN D.: Joint-dependent local deformations for hand animation and object grasping. In *Graphics Interface '88* (June 1988), pp. 26–33.

[Nie93] NIELSON G. M.: Scattered data modeling. *IEEE Comput. Graph. Appl. 13*, 1 (1993), 60–70.

[OBBH00] O'BRIEN J., BODENHEIMER R., BROSTOW G., HODGINS J.: Automatic joint parameter estimation from magnetic motion capture data. In *Graphics Interface* (2000), pp. 53–60.

[PFTV92] PRESS W. H., FLANNERY B. P., TEUKOLSKY S. A., VETTERLING W. T.: *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, Cambridge (UK) and New York, 1992.

[PSS02] PIGHIN F., SZELISKI R., SALESIN D. H.: Modeling and animating realistic faces from images. *International Journal of Computer Vision 50*, 2 (Nov. 2002), 143–169.

[SIC01] SLOAN P.-P. J., III C. F. R., COHEN M. F.: Shape by example. In *2001 ACM Symposium on Interactive 3D Graphics* (Mar. 2001), pp. 135–144.

[WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: Least-squares approximation techniques for skin animation. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 129–138.

[YCC99] YASUMURO Y., CHEN Q., CHIHARA K.: Three-dimensional modeling of the human hand with motion constraints. *Image and Vision Computing 17*, 2 (1999), 149–156.