Automatic Posing of a Meshed Human Model Using Point Clouds

Tamal K. Dey[a], Bo Fu[b], Huamin Wang[a], Lei Wang[a]

[a]*The Ohio State University*
[b]*University of Kentucky*

Abstract

We introduce a markerless approach to deform a quality human body template mesh from its original pose to a different pose specified by a point cloud. The point cloud may be noisy, incomplete, or even captured from a different person. In this approach, we first build coarse correspondences between the template mesh and the point cloud through a squeezed spectral embedding technique that exploits human body extremities. Based on these correspondences, we define the goal of non-rigid registration using an elastic energy functional and apply a discrete gradient flow to reduce the difference between a coarse control mesh and the point cloud. The deformed template mesh can then be obtained from the deformation of the control mesh using mean value coordinates afterwards. Our experiments show (see the supplementary video) that the approach is capable of equipping a mesh with the pose of a scanned point cloud data even if it is incomplete and noisy.

1. Introduction and Previous Work

The flexibility of the human body allows human to perform many large deformations in the real world. This makes non-rigid registration of the human body a difficult problem in computer graphics. When the body deforms continuously over time, the deformation between two consecutive frames is relatively small, and spatio-temporal coherence can be fully explored to improve the registration quality. Commonly known as *performance capture*, continuous non-rigid registration has been widely studied by researchers and it has been successfully handled by many techniques for human bodies [1, 2, 3, 4] and human faces [5, 6, 7, 8, 9].

Compared with performance capture, a more generic yet challenging problem is how to perform non-rigid registration between two arbitrary human body shapes. If this problem can be solved, the resulting technique will be useful in human model reconstruction and completion, performance capture initialization, human pose and shape detection, and many other applications. Existing techniques on this problem can be roughly classified into two groups: those that require parametric body models and those that do not.

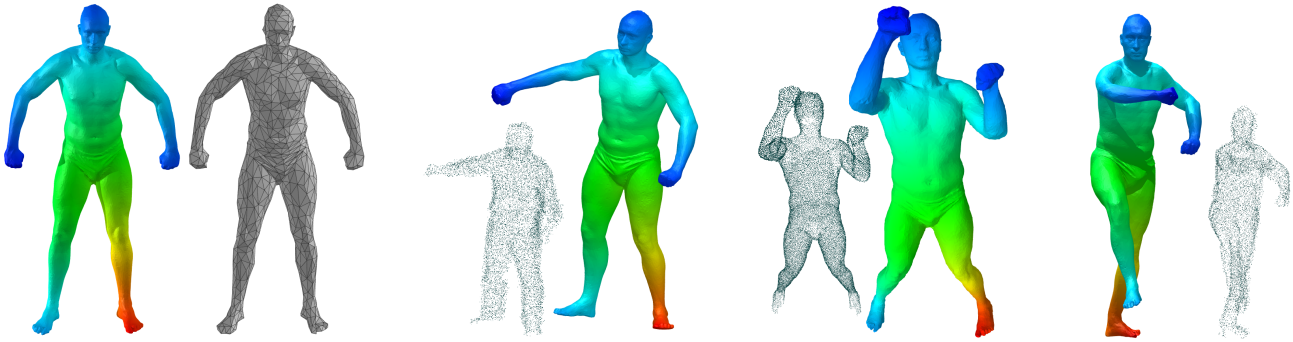
Model-based Registration. The parametric human body models [10, 11, 12] and their variations [13, 14] developed by researchers previously can be used as templates for non-rigid registration of two body shapes under large deformation. Using these models, aligning a human body model with a novel body shape in a different pose can be formulated into a parameter optimization problem. In general, template-based registration is robust and fast. But since it must represent human bodies in the parametric space, it is difficult to handle the detailed body of each individual person, without using a large number of parameters (and data samples). Constructing a large and detailed human body database and extracting models from it is also complex and time consuming.

Registration without a Model. Any technique that does not rely on a parametric human model can be classified into this group, including those using human body templates. Many of these techniques [15, 16, 17] require manual intervention to specify point correspondences. Without using user-specified correspondences, Li et al. [18] handled non-rigid registration between two incomplete surfaces by nonlinear optimization, if the deformation is not too large. Other techniques [19, 20, 21] are typically formulated based on isometric constraints, assuming that the geodesic distances should not change when the body deforms from one pose to another. Especially, spectral methods based on Laplace operator has been proved to be powerful tools for this purpose, see the survey by Zhang et al. [22] for detailed coverage of them.

In this work, we propose an approach to deform a detailed human body template from its reference pose to a largely deformed pose, represented by a point cloud. Our approach first establishes a few correspondences between points of the template mesh and the point cloud. For these correspondences, we draw upon spectral based methods that have been shown to be effective in dealing with non-rigid transformations [23, 24, 25]. Based on Global Point Signature (GPS), our method first calculates only five correspondences as Figure 2a shows. They are further extended to more automatically selected points. Our method then uses the correspondences and the discrete gradient flow to evolve a coarse control mesh from the reference pose into the new pose as shown in Figure 2b. Finally, it refines the alignment by minimizing the difference between the control mesh and the point cloud, and transfers deformation to the fine template mesh, as Figure 2c shows.

The main advantage of this automatic approach lies in its robustness against noise and occlusions as demonstrated by our experiments. This is due to two unique contributions that we propose in this paper:

• *A spectral based method specifically geared toward estab-*



(a) The template mesh and its control mesh (b) Mesh posed by 20K points (c) Mesh posed by 20K points (d) Mesh posed by 19K points

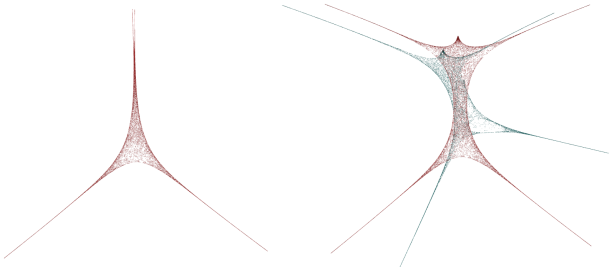
Figure 1: Our method can robustly and automatically deform a template mesh in (a) into different poses as shown in (b), (c), and (d), even when the point clouds are noisy and incomplete. We color the meshes to indicate the correspondences among them.

lishing correspondences between different human body poses. The main observation is that the five extremities of human body, the four arm and leg ends and the head-top can be matched robustly and reliably by a *squeezed* spectral embedding even in the presence of typical occlusions and noise. Furthermore, we bootstrap the initial five correspondences to a larger matching by registering and uniformly sampling the squeezed embedding.

• *A variational approach for human body deformation and alignment.* By formulating an energy functional as the sum of the elastic energy and the alignment energy, we use a discrete gradient flow to find the deformed body shape in an iterative way. Using this approach, we can conveniently adjust the result quality between better shape size preservation and better alignment (i.e., non-rigid registration). Experiments with real data scanned by Kinect depth cameras demonstrate that our approach is efficient and robust against data noise and incompleteness.

2. Computing Correspondences

We are given a complete template surface mesh **M** of a human body and a scanned point cloud **P** of another body possibly in a different pose and with occlusions and noise. Our goal is to establish a few reliable correspondences between the points of **M** and those of **P** first which are bootstrapped further. In particular, we look for identifying the extremities such as the end points of hands, legs, and the top of the head. One could choose to do so in the model spaces of **M** and **P** directly using heat kernel signature (HKS) [23, 24, 26]. However, separating these specific extremities from others in the model space is hard. We take the advantage of a squeezed spectral embedding to detect them. Also, the presence of occlusions and noise make the HKS approach challenging. More serious is the problem posed by the symmetry present in the human body. Although spectral matchings are good in factoring out the isometric deformations [27, 25], they sometimes provide wrong correspondences in presence of human body symmetries. For example, the left foot extremity in one model may match to that of the right foot in another model leading to catastrophic result when



(a) Two eigenfunctions (b) Three eigenfunctions

Figure 3: Squeezed embedding of shapes in the GPS domain. Red point cloud is the embedding of the template mesh. Green point cloud is the embedding of input scan.

such a coarse matching is attempted to be extended to the whole body. Taking the cue from [27], we try all possible matchings between only a very few extremities and settle on the one that provides the best overall matching in a spectral embedding.

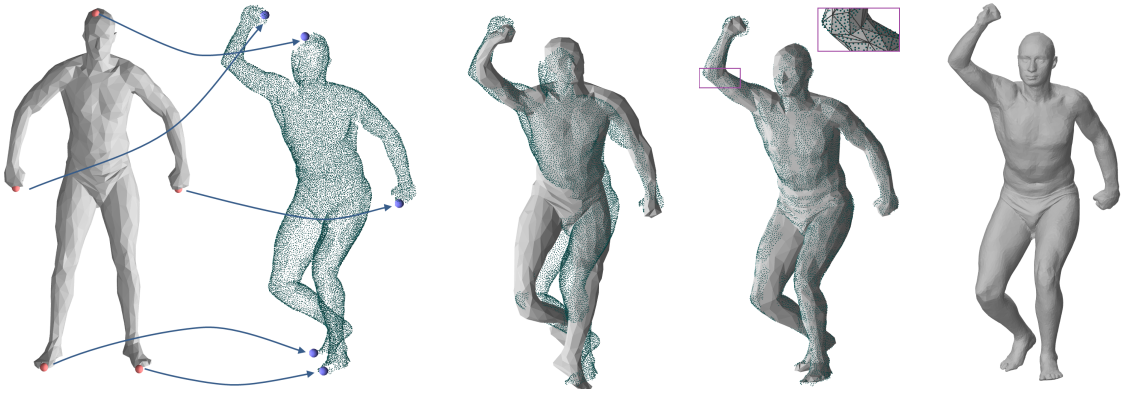
2.1. GPS Embedding

We use the Global Point Signature (GPS) proposed by Rustamov [28] for spectral embedding of **M** and **P**.

Let $\lambda_0 = 0 < \lambda_1 < \lambda_2 < \cdots$ denote the eigenvalues of the Laplace-Beltrami operator of a surface in $\mathbb{R}^3$. Let the corresponding eigenfunctions be $\phi_0, \phi_1, \phi_2, \cdots$. The GPS embedding maps a point p of the shape to:

$$\mathrm{GPS}(p) = \left(\frac{\phi_1(p)}{\sqrt{\lambda_1}}, \frac{\phi_2(p)}{\sqrt{\lambda_2}}, \frac{\phi_3(p)}{\sqrt{\lambda_3}}, \cdots\right). \quad (1)$$

The above eigenspace embedding builds upon the principles that eigenfunctions are invariant under isometric deformation and are orthogonal to each other. In our case, the model **M** and the point cloud **P** are derived from surfaces of human bodies. Thus, they are close metric-wise in most cases though may not be exactly isometric. The spectra of such surfaces are known to be close [29]. Notwithstanding this positive aspect, we face with two difficulties when dealing with two representations **M** and **P** of the same or different human bodies. The first difficulty concerns with the scale. To bring the GPS embeddings of



(a) Initial correspondences (b) Initial alignment (c) Final alignment (d) Deformed mesh

Figure 2: Intermediate results of our method.

the two representations into the same scale, we rigidly scale the embedding of **P** by the ratio of **M**'s λ_1 to that of **P**.

The second difficulty is more serious which primarily motivates our squeezed embedding approach. Since we deal with noisy and incomplete data, the eigenfunctions of the two representations are not readily comparable. In particular, there can be extra or missing eigenfunctions due to occlusion and noise. To alleviate this problem, we consider only the first three eigenfunctions ϕ_1, ϕ_2, and ϕ_3 corresponding to the first three non-zero eigenvalues. The GPS embedding only with the first eigenfunction ϕ_1 squeezes the human body along the central symmetry line. Adding the second eigenfunction ϕ_2 flattens this embedding to a triangular shape capturing the extremities of the two legs and the top of the head. The third eigenfunction ϕ_3 unfolds it to a three dimensional shape that have the two legs and the two arms laid on planes that are somewhat orthogonal. Figure 3 illustrates these embeddings. We consider GPS embedding with the three eigenfunctions that capture the arms, legs, and head-top extremities and call it a *squeezed* GPS embedding. Such a small number of eigenfunctions also saves time spent on eigen decomposition.

To compute the eigenvalues and the eigenfunctions from the point cloud **P**, we consider the standard Graph Laplacian on the adjacency graph of **P**. The sparse matrix $L = (a_{p,\,q})$ of the Graph Laplacian is defined as:

$$a_{p,\,q} = \begin{cases} w_{p,\,q} > 0 & \text{if } \|p - q\| < r \\ 0 & \text{otherwise} \end{cases}$$

$$a_{p,\,p} = -\sum_{q} w_{p,\,q},$$

where r is a small threshold distance used to connect points in **P**. We take r as the average distance to the 20th neighbor for 10 randomly chosen points, so each point has about 20 neighbors within the radius. Entries of L are Gaussian weighted based on the Euclidean lengths of the connections. More specifically,

$$w_{p,\,q} = \frac{e^{\frac{\|p-q\|^2}{4h}}}{4\pi h^2} \text{ and } h = \frac{r^2}{36}.$$

Practically, we resample **M** to a point cloud of 25K points to build its matrix.

2.2. Extrema Correspondence

As **M** and **P** do not represent exactly the same shape, we cannot expect that their squeezed GPS embedding aligns properly. Incompleteness and/or noise in data may contribute more to this misalignment. For example, for embeddings in Figure 3, a rotation is needed to properly register the two squeezed GPS embeddings. Traditional ICP [30] can not be used here because it can be easily trapped in local minimum. We exploit the 5 extrema to guide the registration. We detect them as local maxima with respect to their Euclidean distance to the origin in the squeezed embedding. The maximum at the smallest distance is taken as head and are matched first. Then, we try all 4! possible extrema correspondence for limbs, compute the corresponding rotation matrix for each and choose the one that generates the smallest error.

2.3. Extending Correspondences

Our goal is to deform the mesh **M** to the pose of the point cloud **P**. To actuate this deformation we need a handful of correspondences between the vertices of **M** and the points in **P**. The initial correspondence of 5 extrema as described in the previous subsection is not sufficient for this purpose. We extend this correspondence further to a few more points, usually 60 points, that uniformly sample the squeezed embeddings.

Let **m** and **p** denote the points in **M** and **P** respectively to which this correspondence is extended. We build the sets **m** and **p** incrementally as follows. Each set is initialized with the five extrema. Then, we continue inserting new vertices in **m** that are furthest from the existing points in **m** where the distance is calculated in the squeezed GPS embedding of **M**. Similarly we build **p**. The furthest point strategy is availed to sample the squeezed GPS embedding more or less uniformly.

Our next step is to establish a correspondence between the sets **m** and **p**. For each vertex v in **m** (similarly for **p**), we compute a 5-element descriptor vector as follows. Let $d_i(v)$, $i = 1, ..., 5$, be the five distances of v to the five extrema in the

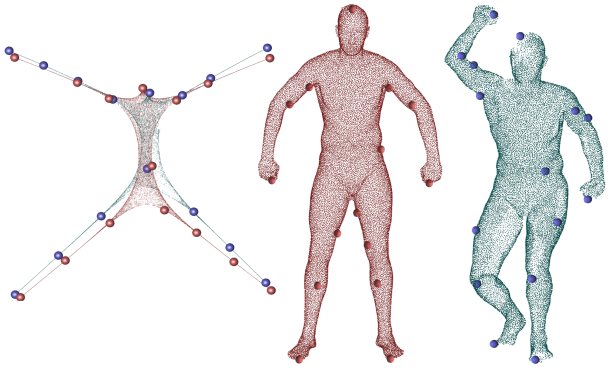


Figure 4: Aligned embedding of two shapes in the GPS domain. Red balls and blue balls are matched points of the template mesh and the input point cloud, respectively.

squeezed GPS embedding of $\mathbf{M}$. Each $d_i(v)$ is taken as the distance between v and the corresponding ith extremum. Our aim is to match points in $\mathbf{m}$ and $\mathbf{p}$ by comparing these distance vectors. For this we need to scale them appropriately. For each extremum i, we have $d_i(v)$ for all points of $\mathbf{m}$ divided by the distance between i and its farthest neighbor in $\mathbf{m}$. The correspondence between points in $\mathbf{m}$ and $\mathbf{p}$ is determined by the Manhattan distance between their descriptors. Figure 4 shows aligned embeddings and the matched points of $\mathbf{M}$ and $\mathbf{P}$.

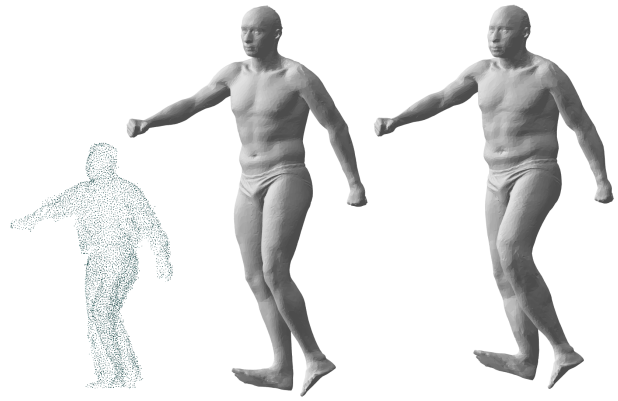
3. Posing Template Mesh

Subsequent to establishing the correspondences, we actuate the deformation of the template mesh $\mathbf{M}$ to the pose of the point cloud $\mathbf{P}$. We observe that simulating the deformation for all vertices in $\mathbf{M}$ is costly. We sidestep this difficulty by considering a coarse mesh, which we call the control mesh.

3.1. Control Mesh Representation

We simplify $\mathbf{M}$ into a surface mesh $\mathbf{M}_c$ with about roughly 1000 triangles and then deform $\mathbf{M}_c$ instead of $\mathbf{M}$. We call $\mathbf{M}_c$ the *control mesh* for $\mathbf{M}$ because we deform the smaller mesh $\mathbf{M}_c$ and recover $\mathbf{M}$ from it by exploiting the mean value coordinate of [31]. Each point of $\mathbf{M}$ can be represented as the weighted sum of the coordinates of the vertices of $\mathbf{M}_c$. The weight matrix which remains fixed across the deformation of $\mathbf{M}_c$ can be precomputed. In practice, we observe that taking $\mathbf{M}_c$ as a volume mesh rather than a surface mesh preserves volume and surface area better during deformation. Therefore, $\mathbf{M}_c$ is further converted into a tetrahedral mesh using TetGen [32]. We choose the option in TetGen that does not introduce any extra vertex on the surface while producing the volume mesh $\mathbf{M}_c$. There exist deformation methods with good performance, such as [33]. However, twisting can happen for large deformation at joints, and local geometry may be distorted by inaccurate correspondence caused by noise and occlusion. We apply the invertible finite volume method [34] to invert the flipped tetrahedra and preserve the volume of the mesh.

(a) Mesh posed by a female TOSCA model



(b) Mesh posed by a 7K point cloud captured by Kinect

Figure 5: Left: the original model. Middle: template mesh deformed by small K_p. Right: template mesh deformed by large K_p.

3.2. The Variational Deformation Framework

Given the correspondences $\langle \mathbf{m}, \mathbf{p} \rangle$ established in Section 2, our next goal is to pose the control mesh $\mathbf{M}_c$ to the point cloud, such that their difference becomes small. Mathematically, we define this as an energy minimization problem:

$$E_{\text{total}} = E_{\text{elastic}} + E_{\text{align}}, \tag{2}$$

in which E_{elastic} is the internal elastic energy of the control mesh and E_{align} is the alignment energy that measures the difference between the control mesh and the point cloud. The solution that minimizes E_{total} can then be found by solving mesh evolution over time, in which the movement of a vertex $\mathbf{x}_i$ with mass m_i can be given as:

$$\frac{d\mathbf{x}_i}{dt} = -\frac{1}{m_i}\frac{\partial E_{\text{total}}}{\partial \mathbf{x}_i}, \tag{3}$$

according to the discrete gradient flow formulation given by Eckstein et al. [35]. Intuitively, Equation 3 can be considered as an elastostatic simulation process, where $\frac{\partial E_{\text{total}}}{\partial \mathbf{x}_i}$ gives the force applied on vertex $\mathbf{x}_i$. According to Equation 2, there are two forces: an elastic force and an alignment force. Here we use the invertible FEM method developed by Irving et al. [34] to compute the elastic force in each tetrahedron of the control mesh. Our experiment shows that inverted tetrahedra can be common in joint regions (such as elbows and knees), where large body

deformation exists. To allow the body to deform more freely, we use a relatively small tensile stiffness but a large Poisson ratio in the constitutive model, so that the body volume change can be reduced. For simplicity, we choose an explicit time integrator to evolve $\mathbf{M}_c$ over time using Equation 3. Since the control mesh is in low resolution, we did not notice any numerical instability issue even when using large time steps. In the rest of this section, we will discuss how to formulate the alignment energy and its force.

3.3. Initial Alignment

To improve the convergence speed of our system, we use an initial alignment process to get the control mesh quickly aligned with the point cloud, before performing detailed alignment in Subsection 3.4. Given a pair of correspondences $\mathbf{x}_m$ and $\mathbf{x}_p$ in $\langle \mathbf{m}, \mathbf{p} \rangle$, we would like to make the Euclidean distance $\left\|\mathbf{x}_m - \mathbf{x}_p\right\|$ as small as possible. But since $\mathbf{x}_m$ may not exist in the control mesh $\mathbf{M}_c$, we find the control mesh vertex $\mathbf{x}_c$ closest to $\mathbf{x}_m$, and then define the energy E_{align} as:

$$E_{\text{align}} = K_i \sum_{\langle \mathbf{m},\mathbf{p} \rangle} \left\|\mathbf{x}_c - \mathbf{x}_p\right\|^2. \tag{4}$$

The resulting force can be intuitively considered a spring force connecting between $\mathbf{x}_c$ and $\mathbf{x}_p$, where K_i adjusts the spring stiffness.

3.4. Point Cloud Alignment

The point correspondences are sparse and not sufficient to get the control mesh aligned with the point cloud in details. So in this subsection, we define the alignment energy E_{align} using the difference between the control mesh and the point cloud immediately:

$$\sum_{\mathbf{x}_c \in \mathbf{M}_c} K_c \min_{\mathbf{x}_p \in \mathbf{P}} \left\|\mathbf{x}_c - \mathbf{x}_p\right\|^2 + \sum_{\mathbf{x}_p \in \mathbf{P}} K_p \min_{\mathbf{x}_c \in \mathbf{M}_c} \left\|\mathbf{x}_c - \mathbf{x}_p\right\|^2, \tag{5}$$

in which K_c and K_p are the stiffness of two types of springs, similar to K_i used in Equation 4. However, there are two issues when using Equation 5 to formulate our alignment forces for surface evolution. The first problem is due to the incompleteness of a point cloud. When a point cloud has missing regions, the difference between a control mesh vertex and its nearest point does not correctly reflect the difference between the two underlying shapes. To solve this problem, we define K_c using a Gaussian falloff: $K_c = de^{-\lambda d}$, in which $d = \min_{\mathbf{x}_p \in \mathbf{P}} \left\|\mathbf{x}_c - \mathbf{x}_p\right\|^2$ and λ is a constant coefficient. We still define the other stiffness K_p as a constant. The second problem is that Equation 5 cannot be differentiated directly. One possible solution is to derive an approximation of Equation 5, which is similar to the pseudo-Hausdorff distance and its gradient flow given by Eckstein and colleagues [35]. Our solution instead is to simply find the nearest point of each vertex (or the nearest vertex of each point), and assume that it remains as the nearest through each surface evolution iteration. As a result, we can still formulate the alignment forces as spring forces between control mesh vertices and points in the point cloud.

Figure 6: Process of posing by depth camera data. Top row, from left to right: pose is captured by three cameras from different views, captured pose in 3D, captured pose is down sampled to about 20K points for deformation. Bottom row: posed template overlapped with captured pose, posed template.

Figure 5 demonstrate the effect of using different stiffness coefficients in our system, when the point cloud is captured from a person with a different body size. If the stiffness is small, the elastic energy makes the result more volume preserving, and if the stiffness is large, the alignment energy makes the mesh more aligned with the point cloud.

4. Results

We tested our system on a workstation with 16GB memory and an Intel Xeon 3.5GHz CPU. Most of our algorithms are implemented in C++, while the eigen-decomposition process is done by MATLAB.

In our experiment, we selected a watertight SCAPE mesh in a rest pose as our template mesh. Its control mesh contains 3.5K tetrahedra and 1K surface triangles. The datasets we used include: low-quality point cloud data captured by three Kinect cameras, the raw SCAPE dataset [10], and the synthetic TOSCA dataset [36]. The results of using different datasets are

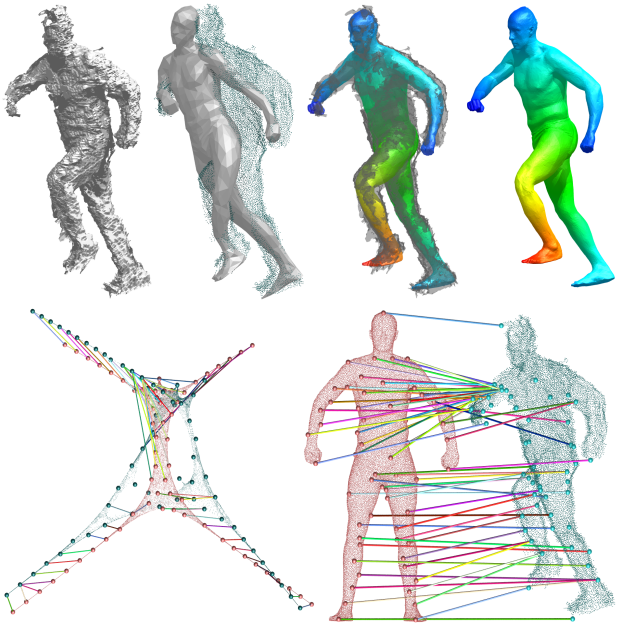


Figure 7: Mesh posed by depth camera data, with intermediate result shown. Top row, from left to right: input scans, control mesh after initial alignment, posed template overlapped with input, posed template. Bottom row: extrema correspondence in both squeezed GPS space and model space.

shown in Figure 1. See the supplementary video for more detailed results.

Figure 6 shows the process of posing by data captured by depth camera such as Kinect. We used three cameras with 120 degree separation to capture poses, then fed down sampled input scans to our method. Figure 7 shows another posed mesh and its intermediate result. Note that the five extrema correspondence is reliable even in the presence of noise and occlusion. Although extended correspondence is disturbed by noise and occlusion, and limbs do not have the same number of extremum, the deformation framework works robustly against this anomaly. Our method works well even under the rough guidance of correspondence. Figure 8 shows more meshes posed by depth camera data.

On average, it took 10 to 30 seconds for our system to run the whole process, including eigenfunction computation. The exact timing of each test case can vary, depending on the size of the point cloud and the magnitude of the deformation. We notice that mesh alignment is usually the performance bottleneck of our system, which can be accelerated by using implicit co-rotational FEM simulation [37] in future.

Although our method can handle noisy and incomplete point clouds, it cannot handle point clouds with self body contacts, i.e., topological changes. Based on this condition, we tested more than 200 point clouds (without topological changes) from the three datasets.

Given sufficiently large alignment stiffness, the average difference between the deformed mesh and the point cloud can become small (less than 10mm) in most cases. However, we did notice large difference occasionally in our experiment, due to a twisting problem as the right hand shown in Figure 5. Fortu-

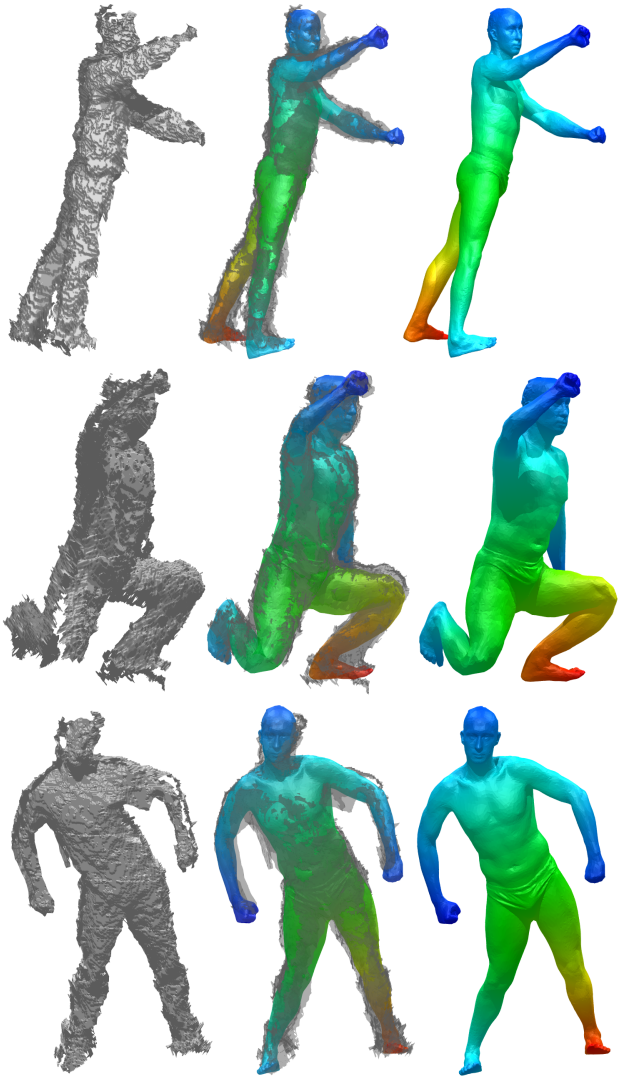


Figure 8: Mesh posed by scans captured by depth cameras. For each row, from left to right: input scans, posed template overlapped with input, posed template.

nately, such problems are not common (less than 5 percent) and most of them can be fixed by adjusting alignment parameters.

To quantitatively understand the influence of the point cloud noise on our results, we create a synthetic example by randomly perturbing the point positions in a point cloud with a certain radius. In this example, we use a small elastic energy term to reduce the error caused by body size variation. Figure 9 shows one deformed result from this test, with noise magnitude of 8mm. Figure 10 shows that the average distance between the body and the point cloud is proportional to the noise magnitude. This is not surprising, since the alignment process tends to deform the body mesh to the middle, when the noises cause the points to form a layer on the body surface. However, large maximum distances (due to the twisting problem mentioned previously) are more likely to occur. In our experiment, we noticed 2 out of ten such cases, when the noise magnitude is above 20mm.

We compared our method with SCAPE [10] and observed

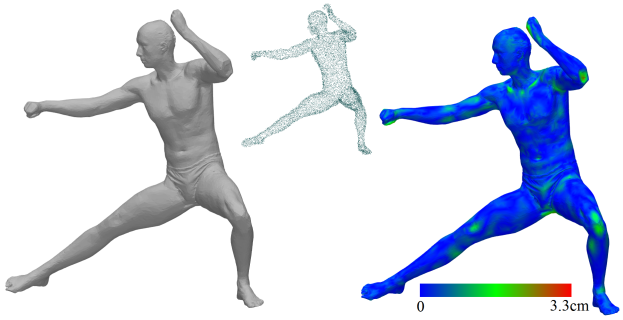


Figure 9: Mesh deformed by a noisy 26K point cloud sampled from a male TOSCA model. Left: deformed template mesh. Middle: noisy input. Right: deformed template mesh colored by its distance to the model.

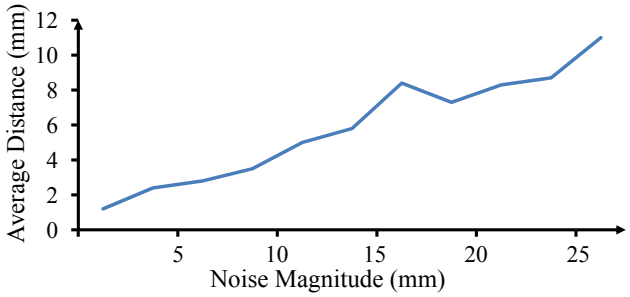


Figure 10: Error plot. This plot visualizes the relationship between the random noises added on a synthetic human body example and the average point-body distance.

our result is competitive with SCAPE. Figure 11 compares the result of our method with the registration result provided by SCAPE, which is based on a probabilistic correspondence model [19] and non-rigid ICP [38]. It shows that when the input point cloud is incomplete, our method can faithfully retain the template mesh details, even if they are not well aligned with the data. In contrast, the method used by SCAPE is more sensitive to this issue as these three examples show.

Limitations. Like other non-rigid registration approaches, our method relies on the point cloud quality. If the point cloud has severe occlusions or if the human point cloud is mixed with other points from the environment, our method cannot handle it. For the same reason, our method cannot accurately align the body details, such as fingers and faces, if the point cloud does not contain such information. Our method cannot robustly handle topological change either.

5. Conclusions and Future Work

In summary, we proposed an automatic approach to align a detailed human body template mesh with human point clouds in different poses. This approach can robustly handle noise and partial occlusion issues in the point cloud data, but not topological change. A recent result in [39] addressing the topological noise may turn out to be useful.

Our immediate plan in the future is to make the system more efficient. We expect 5× to 20× speedups from our system optimization. Although default parameters in our system work well most of the time, we would like to study whether they can be automatically determined based on the alignment process. We are also interested in making our method more robust against severe occlusion issues, for example, the absence of the whole upper or lower body. Finally, topological change is a challenging yet interesting problem that we would like to address.

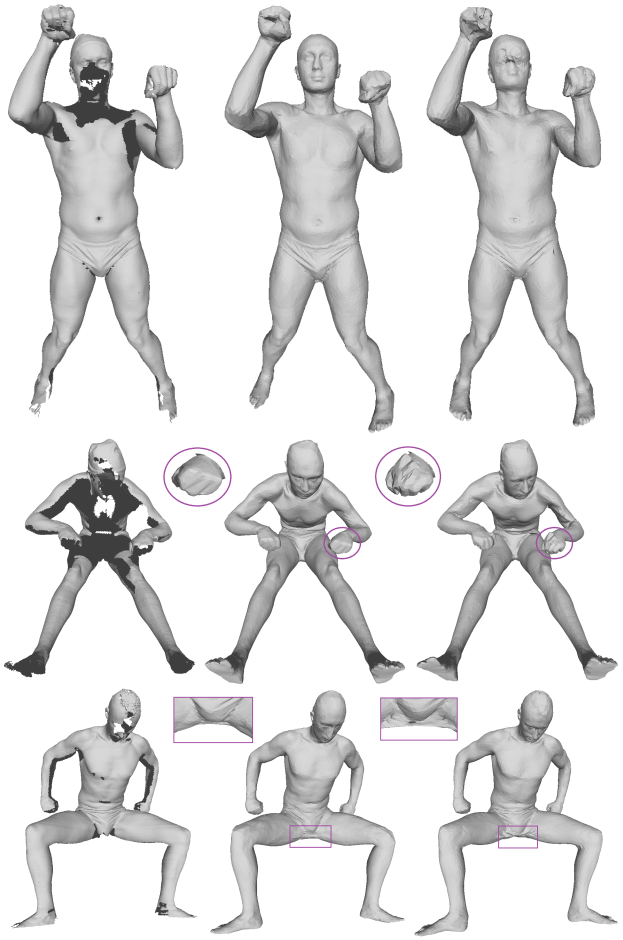


Figure 11: Comparison of our results with SCAPE data. Left column: incomplete scans; middle column: deformed template mesh based on our method; right column: morphed template by SCAPE.

Acknowledgement

The authors thank the anonymous reviewers for their helpful comments. Most of the models used in this paper are courtesy of SCAPE and TOSCA datasets. We acknowledge the support of the NSF grants CCF-1116258 and CCF-1318595.

References

[1] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, S. Thrun, Performance capture from sparse multi-view video, ACM Trans. Graph. (SIGGRAPH) 27 (3) (2008) 98:1–98:10.

[2] M. Liao, Q. Zhang, H. Wang, R. Yang, M. Gong, Modeling deformable objects from a single depth camera, in: IEEE Int'l Conf. Computer Vision, 2009, pp. 167–174.

[3] D. Vlasic, I. Baran, W. Matusik, J. Popović, Articulated mesh animation from multi-view silhouettes, ACM Trans. Graph. (SIGGRAPH) 27 (3) (2008) 97:1–97:9.
[4] H. Li, B. Adams, L. J. Guibas, M. Pauly, Robust single-view geometry and motion reconstruction, ACM Trans. Graph. (SIGGRAPH Asia) 28 (5) (2009) 175:1–175:10.
[5] D. Bradley, W. Heidrich, T. Popa, A. Sheffer, High resolution passive facial performance capture, ACM Trans. Graph. (SIGGRAPH) 29 (4) (2010) 41:1–41:10.
[6] T. Weise, S. Bouaziz, H. Li, M. Pauly, Realtime performance-based facial animation, ACM Trans. Graph. (SIGGRAPH) 30 (4) (2011) 77:1–77:10.
[7] S. Bouaziz, Y. Wang, M. Pauly, Online modeling for realtime facial animation, ACM Trans. Graph. (SIGGRAPH) 32 (4) (2013) 40:1–40:10.
[8] H. Li, J. Yu, Y. Ye, C. Bregler, Realtime facial animation with on-the-fly correctives, ACM Trans. Graph. (SIGGRAPH) 32 (4) (2013) 42:1–42:10.
[9] C. Cao, Y. Weng, S. Lin, K. Zhou, 3D shape regression for real-time facial animation, ACM Trans. Graph. (SIGGRAPH) 32 (4) (2013) 41:1–41:10.
[10] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, J. Davis, SCAPE: Shape completion and animation of people, ACM Trans. Graph. (SIGGRAPH) 24 (3) (2005) 408–416.
[11] B. Allen, B. Curless, Z. Popović, A. Hertzmann, Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis, in: Symp. Computer Animation, 2006, pp. 147–156.
[12] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, H.-P. Seidel, A statistical model of human pose and body shape, Symp. Geometry Processing 28 (2) (2009) 337–346.
[13] A. Weiss, D. Hirshberg, M. J. Black, Home 3D body scans from noisy image and range data, in: IEEE Int'l Conf. Computer Vision, 2011, pp. 1951–1958.
[14] D. A. Hirshberg, M. Loper, E. Rachlin, M. J. Black, Coregistration: Simultaneous alignment and modeling of articulated 3D shape, in: European Conf. Computer Vision, 2012, pp. 242–255.
[15] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, L. J. Guibas, Example-based 3D scan completion, in: Symp. Geometry Processing, 2005, pp. 23–32.
[16] B. Amberg, S. Romdhani, T. Vetter, Optimal step nonrigid ICP algorithms for surface registration, in: IEEE Conf. Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[17] R. W. Sumner, J. Schmid, M. Pauly, Embedded deformation for shape manipulation, ACM Trans. Graph. (SIGGRAPH) 26 (3).
[18] H. Li, R. W. Sumner, M. Pauly, Global correspondence optimization for non-rigid registration of depth scans, in: Symp. Geometry Processing, 2008, pp. 1421–1430.
[19] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, J. Davis, The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces, in: Neural Info. Processing Systems, 2004.
[20] A. M. Bronstein, M. M. Bronstein, R. Kimmel, Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching, Proc. the National Academy of Sciences of the United States of America 103 (5) (2006) 1168–1172.
[21] Q.-X. Huang, B. Adams, M. Wicke, L. J. Guibas, Non-rigid registration under isometric deformations, in: Symp. Geometry Processing, 2008, pp. 1449–1457.
[22] H. Zhang, O. van Kaick, R. Dyer, Spectral mesh processing, Computer Graphics Forum 29 (6) (2010) 1865–1894.
[23] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, Computer Graph. Forum 28 (5) (2009) 1383–1392.
[24] M. Ovsjanikov, Q. Mérigot, F. Mémoli, L. J. Guibas, One point isometric matching with the heat kernel, Computer Graph. Forum 29 (5) (2010) 1555–1564.
[25] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, L. Guibas, Functional maps: A flexible representation of maps between shapes, ACM Trans. Graph. 31 (4) (2012) 30:1–30:11.
[26] R. Litman, A. M. Bronstein, M. M. Bronstein, Diffusion-geometric maximally stable component detection in deformable shapes, Computers & Graphics 35 (3) (2011) 549–560.
[27] A. Dubrovina, R. Kimmel, Approximately isometric shape correspondence by matching pointwise spectral features and global geodesic structures, Advances in Adaptive Data Analysis 03 (2011) 203–228.
[28] R. M. Rustamov, Laplace-Beltrami eigenfunctions for deformation invariant shape representation, in: Symp. Geometry Processing, 2007, pp. 225–233.
[29] T. K. Dey, P. Ranjan, Y. Wang, Convergence, stability, and discrete approximation of laplace spectra, in: Symp. Discrete Algorithms, 2010, pp. 650–663.
[30] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: 3-D Digital Imaging and Modeling, 2001, pp. 145–152.
[31] T. Ju, S. Schaefer, J. Warren, Mean value coordinates for closed triangular meshes, ACM Trans. Graph. (SIGGRAPH) 24 (3) (2005) 561–566.
[32] H. Si, Tetgen.
URL `http://wias-berlin.de/software/tetgen/`
[33] G. Rong, Y. Cao, X. Guo, Spectral mesh deformation, Visual Computer 24 (7) (2008) 787–796.
[34] G. Irving, J. M. Teran, R. P. Fedkiw, Invertible finite elements for robust simulation of large deformation, in: Symp. Computer Animation, 2004, pp. 131–140.
[35] I. Eckstein, J.-P. Pons, Y. Tong, C.-C. J. Kuo, M. Desbrun, Generalized surface flows for mesh processing, in: Symp. Geometry Processing, 2007, pp. 183–192.
[36] A. M. Bronstein, M. M. Bronstein, R. Kimmel, Numerical Geometry of Non-Rigid Shapes, Springer, 2009.
[37] M. Müller, L. McMillan, J. Dorsey, R. Jagnow, Real-time simulation of deformation and fracture of stiff materials, in: Symp. Computer Animation, 2001, pp. 113–124.
[38] D. Hahnel, S. Thrun, W. Burgard, An extension of the ICP algorithm for modeling nonrigid objects with mobile robots, in: Intl. Joint Conf. AI, 2003, pp. 915–920.
[39] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, R. Kimmel, Coupled quasi-harmonic bases, Computer Graph. Forum 32 (2pt4) (2013) 439–448.