Received: 5 February 2020 | Revised: 28 November 2020 | Accepted: 19 March 2021

DOI: 10.1002/cav.1995

RESEARCH ARTICLE

WILEY

Human body reconstruction from limited number of points

Oğuzhan Taştan | Yusuf Sahillioğlu

Computer Engineering Department, Middle East Technical University, Ankara, Turkey

Correspondence
Yusuf Sahillioğlu, Department of Computer Engineering, Middle East Technical University, Ankara, 06800, Turkey.
Email: ys@ceng.metu.edu.tr

Funding information
Türkiye Bilimsel ve Teknolojik Araştirma Kurumu, Grant/Award Number: 215E255

Abstract

We propose a novel approach for reconstructing plausible three-dimensional (3D) human body models from small number of 3D points which represent body parts. We leverage a database of 3D models of humans varying from each other by physical attributes such as age, gender, weight, and height. First we divide the bodies in database into seven semantic regions. Then, for each input region consisting of maximum 40 points, we search the database for the best matching body part. For the matching criterion, we use the distance between novel point-based features of input points and body parts in the database. We then combine the matched parts from different bodies into one body, with the help of Laplacian deformation, which results in a plausible human body. To evaluate our results objectively, we pick points from each part of the ground-truth human body models, then reconstruct them using our method and compare the resulting bodies with the corresponding ground-truths. Also, our results are compared with registration-based results. In addition, we run our algorithm with noisy data to test the robustness of our method and run it with input points whose body parts are manually edited, which produces plausible human bodies that do not even exist in our database. Our experiments verify qualitatively and quantitatively that the proposed approach reconstructs human bodies with different physical attributes from a small number of points using a small database.

KEYWORDS

3D body reconstruction, database-assisted modeling, modeling

1 | INTRODUCTION

Modeling human body digitally is one of the most attractive field in computer graphics as these models have wide-ranging applications. In clothing, size, and shape of the body are important to improve cloth comfort and fit.[1] In medicine, the human model data can be used for arranging the doses of medication[2] and monitoring the posture of human body.[3] There exist some studies which aim to create dynamic digital models that mimic physiological and behavioral realism.[4,5]

This paper considers the problem of generating plausible three-dimensional (3D) human bodies from small number of input points. Generating human body is already a difficult task because human body can be in large number of different shapes and poses. Moreover, our task is even more challenging due to the lack of surface features such as mesh structure and normal vectors in the input. Another challenge is the absence of the correspondence between sparse input points and complete database models.

In real life, the sparse points can be obtained from two-dimensional (2D) images or array of sensors as in the case of the popular motion capture scenario with single[6] or multiple cameras,[7] touch-probe,[8] laser triangulation,[9] wearable sensors,[10] and laser pulse.[11] As another sensor-based technology,[12] makes use of a small set of inertial sensors attached to the body. They deform a template human body based on the measurements from the sensors to capture the motion of human. Although they place six sensors to the wrists, lower legs, back and head, increasing the number of sensor can produce 3D points on human body which can be used as input to our method.

In addition to these so-called active methods where the real-world object is worn markers or hit by emitted beams, there also exist passive acquisition methods based on existing images and videos. To this end, Reference 13 proposes a neural network which learns a nonlinear mapping from a 2D image to a set 3D points that represent a human face. Reference 14 proposes a method which learns salient points on the fingers and pose of hand. Their method can be extended to learn salient points on human body which can be the input to our method, making our method capable of creating 3D human body models from a single image, just like the SMPL model does.[15] In this line of thought, 3D skeleton output of 3D pose estimators that run on single 2D images[16] can be used as input to our generic framework, rendering our method a potentially useful 2D image to 3D model generator.

Among all these active and passive data acquisition possibilities that can be seamlessly incorporated into our framework, we decide to simulate the touch-probe mechanism by manually selecting our sparse input points over an existing 3D model and then try to recover the surface out of this selection. This is deemed appropriate for research purposes as we naturally have a ground-truth 3D model to compare our reconstruction result with (see Section 3).

The main contributions of the paper are as follows:

- **Novel method for reconstructing human body:** A novel method is proposed to reconstruct human body from small number of 3D points by following a data-driven approach which exploits a database of 3D human body models.
- **Novel features for 3D points:** We propose novel features which compactly and distinctively represent the geometry of a set of 3D points. Our experiments show that the proposed features can effectively distinguish between 3D point sets in different geometry and recognize 3D point sets in similar geometry.

This paper is organized as follows: In Section 2, the related literature is reviewed by focusing on the studies which aim to reconstruct human bodies, specifically the ones which accept sparse inputs. In Section 3, the database in use, our proposed geometric features and the method for reconstructing 3D human body are explained. In Section 4, the experimentation of our method and its limitations are given. In Section 5, the conclusion and future work are discussed.

2 | RELATED WORK

Due to the the growing demand for virtual human models, there is a large body of literature on reconstructing human bodies and faces. The paper which is the most similar to ours[17] expects, unlike our simpler sparse input requirement, a dense point cloud as input. As we do, they leverage a database of 3D meshes. Their method semantically divides the meshes into several parts in order to compare the input with the meshes in the database. Differently from our method, their method reconstructs an articulation-free face rather than a full articulated human body and also they require anatomical facial landmark points labeled manually by a trained expert, which we avoid. They first align input points to database by detecting predefined facial landmarks and using them for rigid pose alignment via Procrustes analysis[18] and then perform dense alignment to a generic mesh.[19] After the alignment process, each face part is matched to the database using distance function which is a weighted average of pseudo-landmarks and histograms of azimuth and elevation components of the surface normals.[20,21]

Reference 22 processes several depth frames, which are again dense point clouds, by detecting and cropping the face in each depth frame. Then, these extracted 3D faces are aligned with each other by using Coherent Point Drift algorithm[23] in order to build a cumulated face. Then, the LOWESS nonparametric regression method is used to approximate the face surface from the cumulated face model and remove outliers from the data. The resulting 3D face is still noisy, and in order to get rid of this noise, they first model the face as a 2D manifold embedded in the 3D space. Then, they treat the problem as manifold reconstruction from noisy data and solve this problem by applying a method which extends the one described in Reference 24 based on a combination of dimensionality reduction and local weighted regression.

In order to reconstruct a full body,[25] built a full-body scanner and a separate face scanner, consisting of 40 and 8 DSLR cameras, respectively. This setup creates a huge body scan data with 4M points. After they scan a full body, they try to align and fit to a template model by performing the following steps. In the initialization step, they optimize scaling, rotation, and translation by using manually selected landmarks using[26] and optimize the joint angles using inverse kinematics based on linear blend skinning.[27] After good initialization, fine-scale nonrigid registration is applied by minimizing an energy which a weighted average of the landmark term, fitting term, and regularization term. The landmark term is to minimize the squared distance between the manually selected landmarks, the fitting term penalizes the squared distance between corresponding points and the regularization term penalizes the geometric distortion from the undeformed model. A similar deformation framework is established in Reference 28 to reconstruct the shape by deforming its bounding sphere in the guidance of silhouette data. Other deformation-based studies conform the template model to the input point set via deep learning[29] or energy minimization.[30-32] All these works require an accurate template model which can hardly be obtained in many cases. We overcome this issue by using templates for parts not for the entire inseparable body. Similar to our work, the distributed part-based representations in References 33 and 34 showed more promising results than the classical existing realistic 3D body models as these frameworks facilitate inference by allowing the model to more effectively explore the space of poses. A significant difference of our proposed method from these methods is that ours is a nonparametric-model-based reconstruction method and directly searches for body parts from a small database. Reference 35 bears another similarity to our work in terms of the usage of sparse marker points for 3D reconstruction. The main difference of this method from ours is the usage of a parametric 3D body model, which we avoid.

Reference 36 trains a convolutional neural network with RGB-D face data in order to reconstruct faces. Their network has coarse-to-fine architecture in which a medium-scale CNN to regress a medium-scale face model is followed by a fine-scale CNN to recover the surface details. Both CNN models are trained in a completely self-supervised manner in that the face shape and details are automatically learned from the large-scale unlabeled RGB-D data.

Reference 37 exploits a database of 3D heads of 200 adults which are in a cylindrical representation produced by a commercial laser scanner. First, they put all the head models in the database in full point-to-point correspondence. Then, they create a morphable model from the head models. Finally, using the morphable model, they create a 3D head from 2D input image. To do so, a set of rendering parameters and coefficients of the 3D model are optimized until an image as close as possible to the input image is produced. Their algorithm generates a 3D face from the current parameters, and renders an image, and updates the parameters according to the residual difference. They start with the average head and with rendering parameters roughly estimated by the user.

As we do, Reference 38 follows a data-driven method which reconstructs articulated 3D motion. Their method extracts all shape information from a set of range scans which is formed by 200K points and 50K triangles. They put the range scans in full correspondences by using markers obtained by Correlated Correspondence[39] which computes the consistent embedding of each instance mesh into the template mesh. Then, their model tries to align the template with each mesh in the data set consisting of different poses of a human. The deformations are modeled for each triangle of the template such that the deformations are applied in the local coordinate frame of each triangle by translating a point of the triangle to the origin. A specific transformation matrix is applied to each triangle and then an articulation rotation is applied. They aim to minimize the least-squares error by optimizing transformation and rotation matrices.

As another data-driven algorithm, Reference 40 also uses a database of pre-existing rigs and extracts usable body parts from it in order to construct rigged bodies for given unrigged target meshes. To do so, they fit together pieces from several different source meshes in the database. Differently from ours, their input is a dense mesh.

Point-based registration methods are related to our work as they try to reconstruct a new shape from the existing template model by deforming it toward the input data. As opposed to our method that requires sparse input, point-based registration methods have to deal with models which are as dense as the template model to be deformed.[41a]nd similarly Reference 42 first embeds the shape into isometric representation. Then, they optimize this embedding as a variant of the classical intrinsic distortion[43-45] objective by using Markov Random Field optimization. Nonisometric version of this study achieves reconstruction of skulls by using volumetric Dirichlet energy.[46] Reference 47 tries to align and label two point clouds in 3D in a simultaneous way until they are as similar as possible via Procrustes Matching procedure in a novel semi-definite programming relaxation framework. We state our main difference from these registration-based methods as dealing with sparser data in a part-based fashion, features which bring better usability and flexibility to the process.

3 | METHOD

In order to build a human body from limited number of 3D points, we make use of a database of human bodies. As a preprocess, we compute a set of features for each body part of each body in the database from the manually selected feature points and save them. Given a set of 3D points for each body part, we compute a set of features and find a best-matching part from the parts in the database by comparing the calculated features of the input points and the saved features of the body parts in the database. Then, the selected body parts from different bodies are combined together to build a human body.

3.1 | Database

In this study, a part of the database provided by Reference 48 is used in order to reconstruct human body parts from small number of 3D input points. Each human mesh in this database has 6890 vertices and 13,776 faces. Five of the 10 samples of our database and the segmentation to be used in the following steps are shown in Figure 1. Note that, only one model is manually segmented (shown in Figure 1) and this information is automatically transferred to the other database models via the available one-to-one correspondence through the dataset.

Note also that we keep the database very small in an effort to show the feasibility of our solution with less data which is easier to obtain and store. We use the neutral poses of 10 different characters with varying gender, age, height, and weight. This choice combined with the one-piece arm and leg segments (Figure 1-right) prevents the reconstruction of highly articulated poses. It, however, enables reconstruction of all other poses as well as different features such as height and weight.

3.2 | Preprocessing of the database

We semantically divide each body into seven segments where the segments are represented by at most 40 ordered 3D representative points. These points are manually selected from a single base model shown in Figure 1, and then transferred to the others by using the existing full correspondence between the models. Single manual segmentation takes merely 5 s thanks to the the simplicity of the model's pose. After segments are made ready, we calculate a feature vector for each segment based on their representative points. Since our feature vector is computed from exactly 40 points,

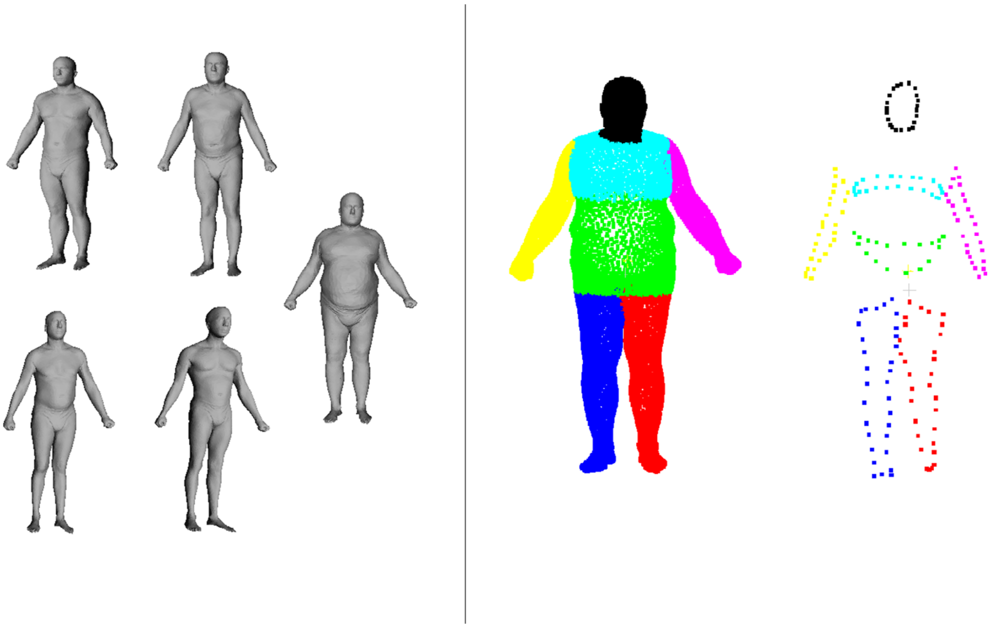

FIGURE 1 (Left) Five samples from our database. (Right) Human body segments and feature points for each body part

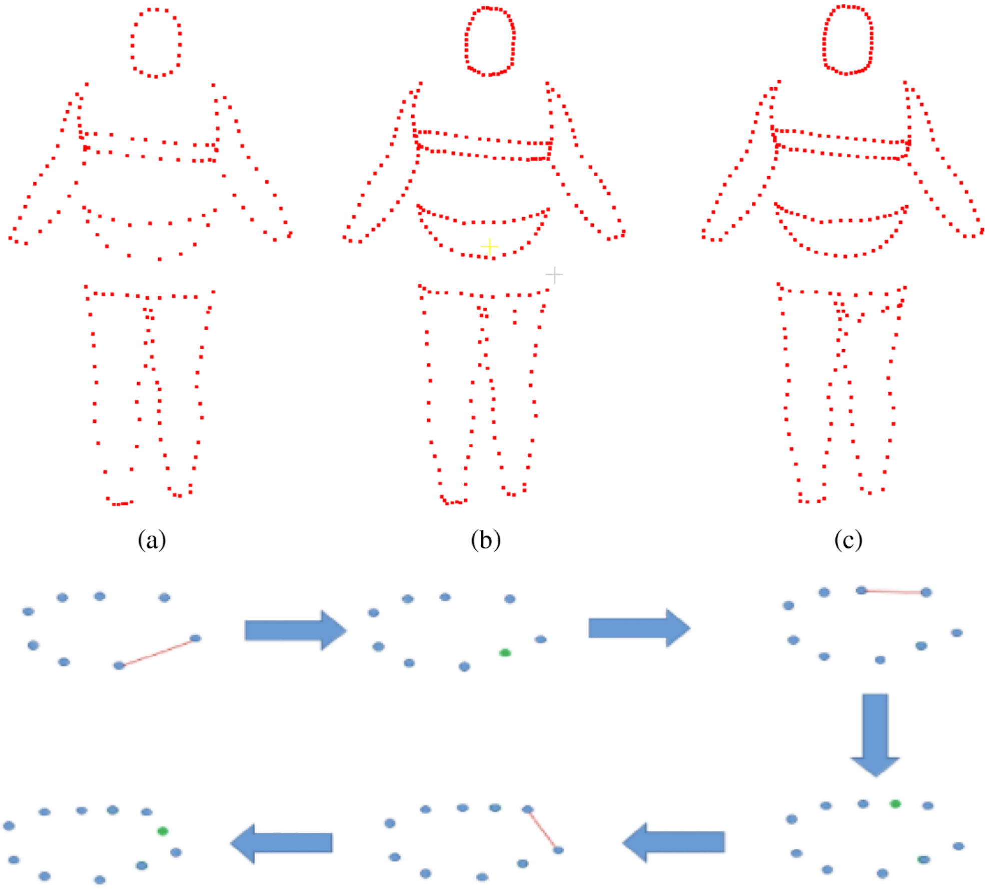

FIGURE 2 Comparison of point completion methods. (a) Input points, (b) Completed points by the linear method, (c) Completed points by the nonlinear method. Three iterations of the linear method is displayed at bottom

we first complete the number of segment points to 40 (user is allowed to select less points for less effort) in linear fashion by adding a point at the middle of the most widely separated consecutive points until it reaches to 40 points. The process is visualized on a didactic example in Figure 2-bottom. Alternatively, we first fit a nonlinear parametric B-spline curve to the input points and then produce 40 uniform points from the fitted curve by changing the curve-defining one-dimensional parameter uniformly. The results of both point completion methods are shown in Figure 2. As we can see, the nonlinear method produces smoother and more uniform points than the linear method does, and hence it is promoted.

3.3 | 3D Features

After completing the segment representative points, a histogram of Euclidean distances between each point to the other representative points, called distance histogram features (DHF), is calculated and shown in Figure 3-left. The representative points are normalized so that they have zero-mean and unit *SD*. In addition, we calculate a vector for each point to the others and look at the angles between those vectors and the *x*-, *y*-, *z*-axes. Then, the histogram of those angles, called angle histogram feature (AHF), for each axis is calculated (shown in Figure 3-right). Since each histogram consists of 20 bins, the feature vector for each body part has a size of 80 (20 for DHF and 20+20+20 for AHF that uses three axes).

3.4 | Body part selection

For each body part, we search for the best match in the database by comparing the feature vector of querying body part and that of each of corresponding body part in database. The body part in the database with minimum feature distance to the querying feature vector is selected as the best match. In order to calculate distances between feature vectors, we use the L2 metric.

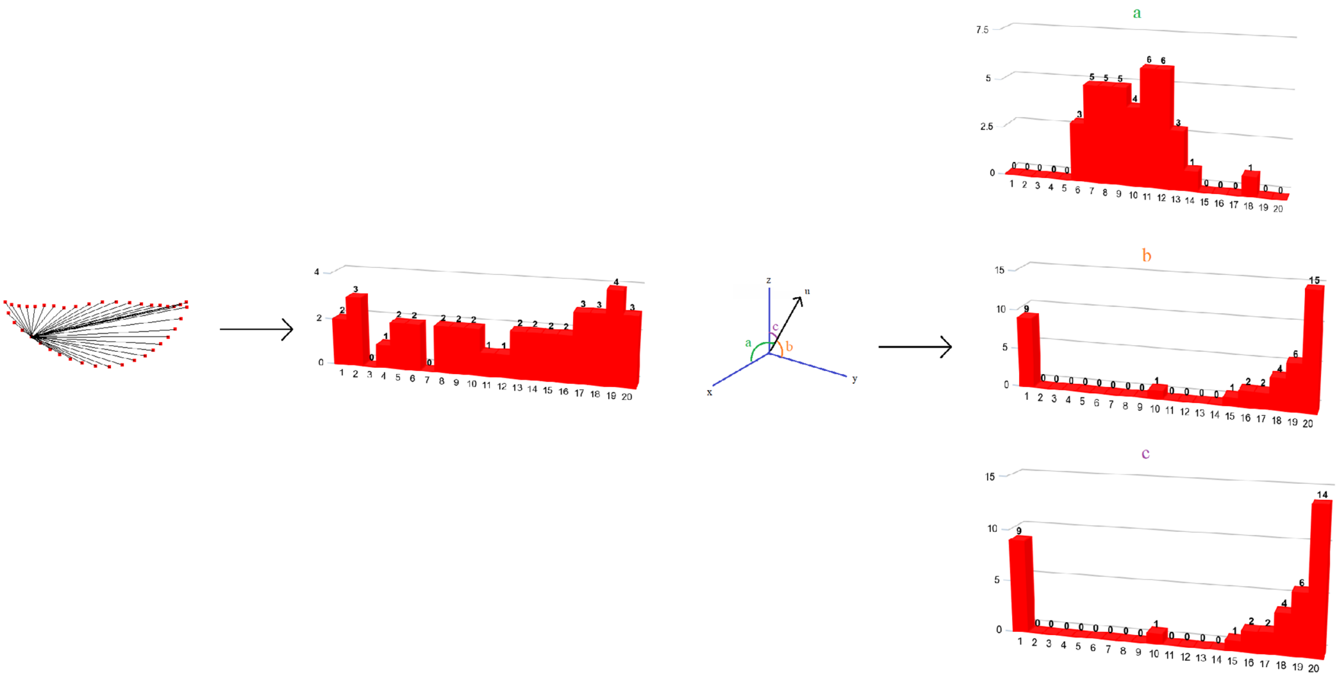

FIGURE 3 Illustration of distance histogram features (DHF) for one seed point is shown on the second column and angle histogram feature (AHF) on the fourth column. The same point set of size 40 is used (first column) for both feature computations. **u** represents the vector between any two points in the point set. DHF is an histogram of ||**u**|| and AHF is for the angles between **u** and *x*-, *y*-, *z*-axes namely, *a*, *b*, *c*, respectively

3.5 | Merging selected body parts

Since each body part is possibly taken from different human bodies, some unpleasant appearances come to existence when the selected parts are merged together. In order to get rid of these problematic results, we follow a four-step procedure which is described in the following subsections. The effect of each step can be seen Figure 4-right. The relative improvements of each of these steps are also visualized in Figure 5.

3.5.1 | Moving base points

We first select one of the models from our database and call it the base model. We will essentially deform this base model in order to get our smooth output. To this end, we first move the vertices of the base model to the corresponding positions of the parts that we selected in Section 3.4. Recall that all of the models in our database are in fixed connectivity, revealing the correspondences immediately.

Since the body parts belong to different bodies there exist some mismatches. In addition, since the range of rotations of the body parts in the database is limited, the selected parts do not fully match with the input points in terms of orientation. As a solution to this problems, we continue with the next steps.

3.5.2 | Rigid body part alignment

In order to get the selected part (**S**) and the input points for that part (**I**) in the same orientation, we calculate and apply a transformation matrix which transform the selected part into input points. This provides individual alignment of each body part separate from each other. To do so, we follow the steps described in Reference 26, first both point sets are moved to origin ($\mathbf{S}'$ and $\mathbf{I}'$) and then, calculate the co-variance matrix as follows:

$$\mathbf{C} = \mathbf{S}'(\mathbf{I}')^T. \tag{1}$$

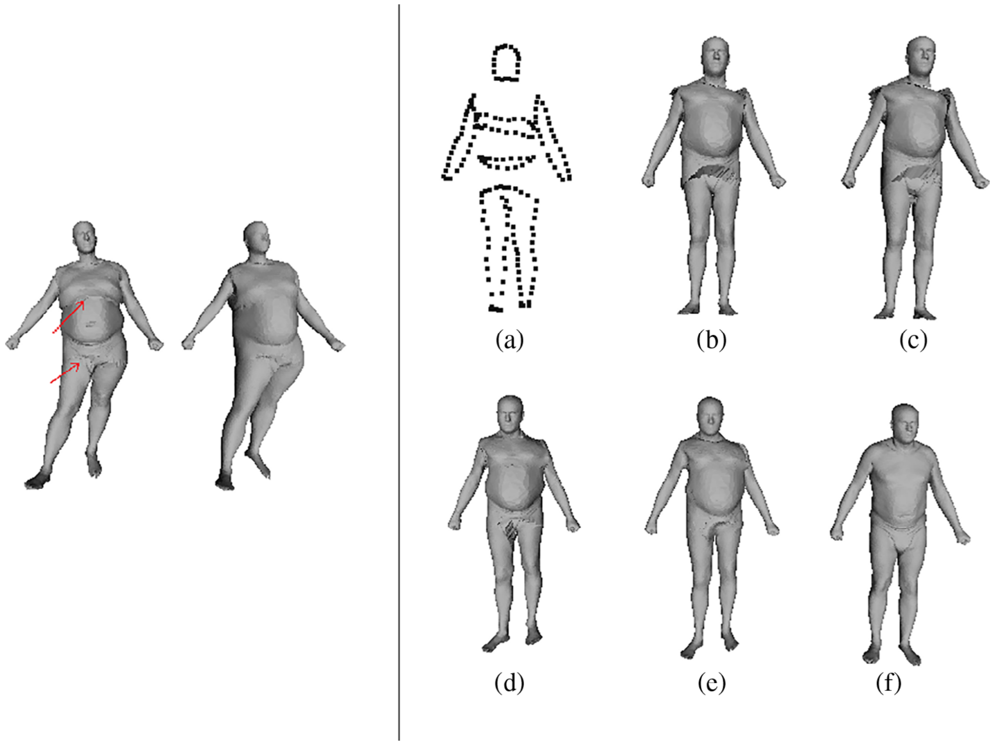

FIGURE 4 (Left) Output of the Interparts Alignment step after Laplacian smoothing does not yield a perfectly smooth mesh (problems pointed by arrows) (left). Smoother and more natural result is obtained when the output of the Inter-parts Alignment goes directly to our Border Smoothing step. This is our final resulting model (right). (Right) All steps are visualized for a different input. (a) Input points, (b) Result after moving base points (3.5.1), (c) Result after Rigid Body Part Alignment (3.5.2), (d) Result after Interparts Alignment (3.5.3), (e) Result after Border Smoothing (3.5.4), (f) Ground-truth result

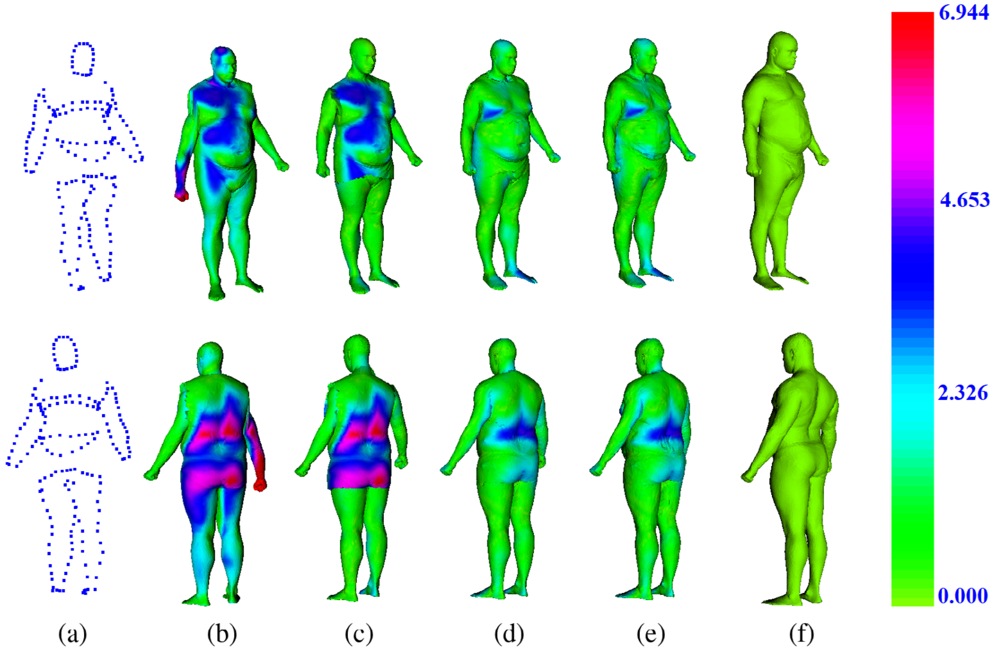

FIGURE 5 The comparison of outputs of our algorithmic steps with color-coded errors. Top row is the front view, bottom row is the back view. (a) input points, (b) output of moving base points, (c) output of rigid body part alignment, (d) output of inter-parts alignment, (e) output of border smoothing, (f) ground-truth. The average errors from column B to column E are 2.120, 1.291, 1.088, 1.076, respectively

Using this co-variance matrix, the following matrix is constructed

$$\begin{vmatrix} C_{00}+C_{11}+C_{22} & C_{12}-C_{21} & C_{20}-C_{02} & C_{01}-C_{10} \\ C_{12}-C_{21} & C_{00}-C_{11}-C_{22} & C_{01}+C_{10} & C_{20}+C_{02} \\ C_{20}-C_{02} & C_{01}+C_{10} & -C_{00}+C_{11}-C_{22} & C_{12}+C_{21} \\ C_{01}-C_{10} & C_{20}+C_{02} & C_{12}+C_{21} & -C_{00}-C_{11}+C_{22} \end{vmatrix}. \tag{2}$$

As Reference 26 suggested, in order to find the best orientation, we calculate eigen-vectors of the matrix. Using these vectors, now we can construct the rigid alignment matrix.

3.5.3 | Interparts alignment

To create a consistent human body, the body parts taken from different bodies should be aligned. To do so, we fix the position of the chest part (light blue colored in Figure 1-right) and add a translation to the head, arms, and belly parts; black-, yellow-, pink-, and green-colored parts in that figure, respectively, so that the borders of those parts are matched. Then, the same operation is followed for legs, blue and red colored in Figure 1-right), to match them with the belly part.

In order to calculate the translation between two parts, we first find the neighboring edges, that is, edges which has one vertex at one part and the other vertex at the other part. Then, an average displacement between the vertices of the neighboring edges is calculated. This displacement is the translation to be needed to align the borders of the body parts from different bodies.

After this operation, the average points of the borders are matched. Since the parts can have different sizes, for example, width of the borders, rough and uneven surfaces appears at the borders of parts. To alleviate this problem, we apply our border smoothing algorithm.

3.5.4 | Border smoothing

Since the selected parts come from different human bodies, there may exist “surgical scars” between neighboring body parts. In order to get rid of these abnormalities, we first simply try Laplacian smoothing[49] with several iterations as well as surface fairing.[50] However, the defects around the joint points are so large that they cannot be cleared by a simple smoothing algorithm as shown in Figure 4-left.

In order to remove the defects more robustly, we perform Laplacian deformation of the base model. In particular, the base model moves under Laplacian energy where we use the selected part points as handles. Consequently, the base model goes to the shape and pose implied by the selected part points. For this deformation, we make use of the differential coordinates by Laplacian of the mesh, which encodes each vertex relative to its neighbors, as defined in Reference 51. For the construction of the Laplacian of the mesh, the cotangent weights[52] are used. Since a differential coordinate is a linear combination of a vertex and its neighbors, the process of constructing differential coordinates for all vertices is as follows:

$$\mathbf{L}\mathbf{v} = \boldsymbol{\delta}, \tag{3}$$

where $\mathbf{L}$ is n-by-n Laplacian matrix, $\mathbf{v}$ is n-by-3 matrix in which each row contains a vertex and $\boldsymbol{\delta}$ is n-by-3 matrix storing differential coordinates.

The regularization term of the deformation energy is the least-squares difference between differential coordinates of the rest-pose of the base model mesh (v_0) and the deformed pose of the base model mesh ($\mathbf{v}$):

$$E_{\text{reg}} = ||\mathbf{L}\mathbf{v_0} - \mathbf{L}\mathbf{v}||^2. \tag{4}$$

On the other hand, the match term is the sum of the squares of the distance between the corresponding points from the deformed pose mesh and the fixed handles h_k designated by the selected parts from Section 3.4:

$$E_{\text{match}} = \sum_{k=1}^{N} ||\mathbf{v_k} - \mathbf{h_k}||^2, \tag{5}$$

where N is the number of handle points, that is, 40 from each of the seven parts, hence 280 for our experiments.

In our case, weighted regularization term and matching term are used to construct total deformation energy (Equation (6)), then a set of vertices is searched for in order to minimize the total deformation energy.

$$E_{\text{tot}} = \alpha E_{\text{reg}} + E_{\text{match}}. \tag{6}$$

To minimize E_{tot}, we first take the derivative of it with respect to $\mathbf{v}$ and then equate it to zero. The resulting sparse linear system to be solved instantly for $\mathbf{v}$ is as follows:

$$(\alpha \mathbf{L}^T \mathbf{L} + \mathbf{I})\mathbf{v} = \alpha \mathbf{L}^T \mathbf{L} \mathbf{v_0} + \mathbf{h}, \tag{7}$$

where α controls the weight of the regularization energy.

The result of each step is shown on the right part of Figure 4. The results of the improvements made in each step, on the other hand, can be shown in Figure 5. The largest improvement has been made by the Rigid Body Part alignment as each part which is independent from each other is put together to build a human body. After Inter-part Alignment, the error is decreased by significant amount. Finally, border smoothing reduces the error slightly but improves the appearance of the body considerably.

4 | EXPERIMENTS AND RESULTS

The proposed approach is evaluated with input body points which are taken from differently posed and shaped models. In order to make the evaluation easy and accurate, the input points are selected from the ground-truth body meshes. However, the real-world applications can be possible such as obtaining the input points directly from 2D images and feed them to our method to produce a 3D model from a 2D image. To do so, there exists various studies in the literature such as Reference 13. Although Reference 13 works with faces, it can be possible to extend their work so that the method produces 3D body points by training their deep learning model with 3D body models. In addition, Reference 53 design a cascaded coupled-regressor approach by integrating a 3D point distribution model in order to estimate the 3D landmarks. Reference 10 also estimates 3D landmark on human bodies by using sensor arrays which are combined with deep learning method which maps measured capacitance from the sensors to the deformed geometry. Another source of data is the sparse markers placed on human bodies for purposes such as motion capture.

Input points, the ground-truth result and outputs of each step are shown in Figure 4-right. The final result of our algorithm has been compared with the results of as-rigid-as-possible (ARAP)-based shape registration algorithm introduced in Reference 54. This algorithm essentially uses ARAP energy to regularize the mesh as it is being pulled toward the data points. ARAP energy first fixes the positions and finds the optimal rotation for each mesh face. It then fixes the rotations and finds the optimal positions. These two steps are alternated until mesh does not move any further. Here a base model is deformed toward the sparse input point cloud. The obvious problem here is the pose and shape of the starting base model for the ARAP method. To keep this competitor method more accurate, we manually select a good start, but note that it is rather a difficult initialization problem, rendering our competitor less preferable than our method. The error metric for comparison is the average Euclidean distance between the output points and the corresponding ground truth points, formulated as follows:

$$\varepsilon = \frac{\sum_{i=1}^{n} ||\mathbf{I}_{\mathrm{i}} - \mathbf{G}_{\mathrm{i}}||}{n\tau}, \tag{8}$$

where n is the number of points in meshes, τ is the average edge length of the ground-truth mesh (in order to interpret the results easily in terms of the average edge lengths), $\mathbf{I}_i$ is ith point of the output mesh and $\mathbf{G}_i$ is ith point of the ground-truth mesh.

The constrained points are initialized with the ground-truth matches of the input points for the ARAP-based method. Some comparisons have been shown in Figures 6–15 with the corresponding error quantification and timing values in Tables 1 and 2, respectively. Note that, for space limitations we only provide output samples for a total of 10 different runs. We, in fact, have tested our algorithm comprehensively using a total of 50 different input configurations. In the following figures, the order of the bodies from left to right is as follows; input points, ARAP-based result, our Laplacian-based result, the ground-truth result. In addition, each body mesh in each figure is colored by the error with respect to the the ground-truth. The error for each point of the mesh is the distance to the closest point in the ground truth mesh after the output mesh is rigidly aligned to ground-truth mesh using Iterative Closest point (ICP) algorithm. The average error for each result produced by the ARAP-based method and two different variants of the proposed method are compared at Table 1. One of the variants interpolates the input points by linear method and the other

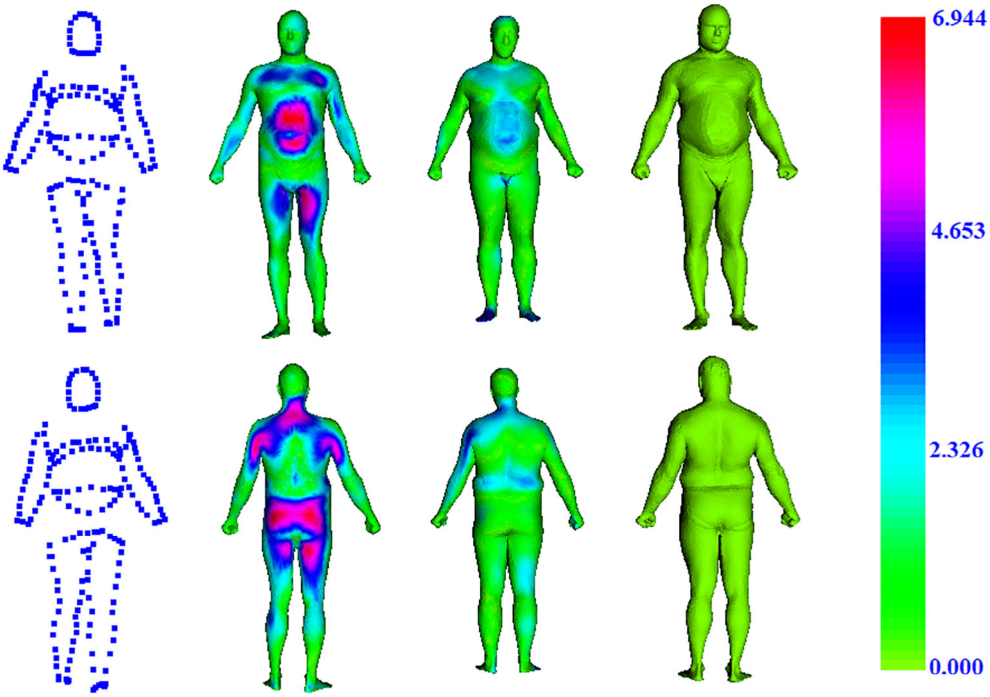

FIGURE 6 Comparison of the results. Top row is the front view, bottom row is the back view. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.736 and 1.125, respectively. The average edge length of the ground-truth mesh is 0.0144

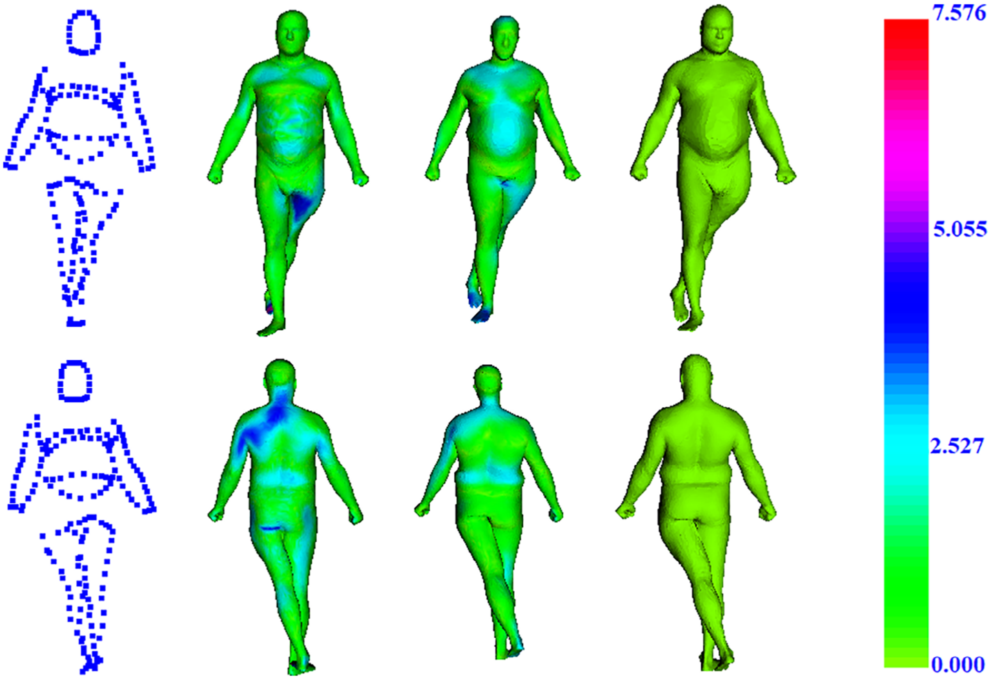

FIGURE 7 Comparison of the results. Top row is the front view, bottom row is the back view. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.23 and 1.103, respectively. The average edge length of the ground-truth mesh is 0.0165

does this by nonlinear spline interpolation. The results show that the proposed method with the spline interpolation performs better than ARAP-based method. In addition, nonlinear point completion slightly improves the reconstruction accuracy compared to the linear point completion method. All errors in the tables and the figures are calculated using Equation (8) which is the relative error with respect to average edge length in order to make the errors more understandable. The execution times obtained on an Intel Core i7-6700K 16 GB RAM PC for each input are shown in Table 2.

As shown in Figures 6–9, our method produces more accurate reconstructions than ARAP-based method does for the most of the body parts. Although the deformation is higher in Figure 10 compared to the other samples,

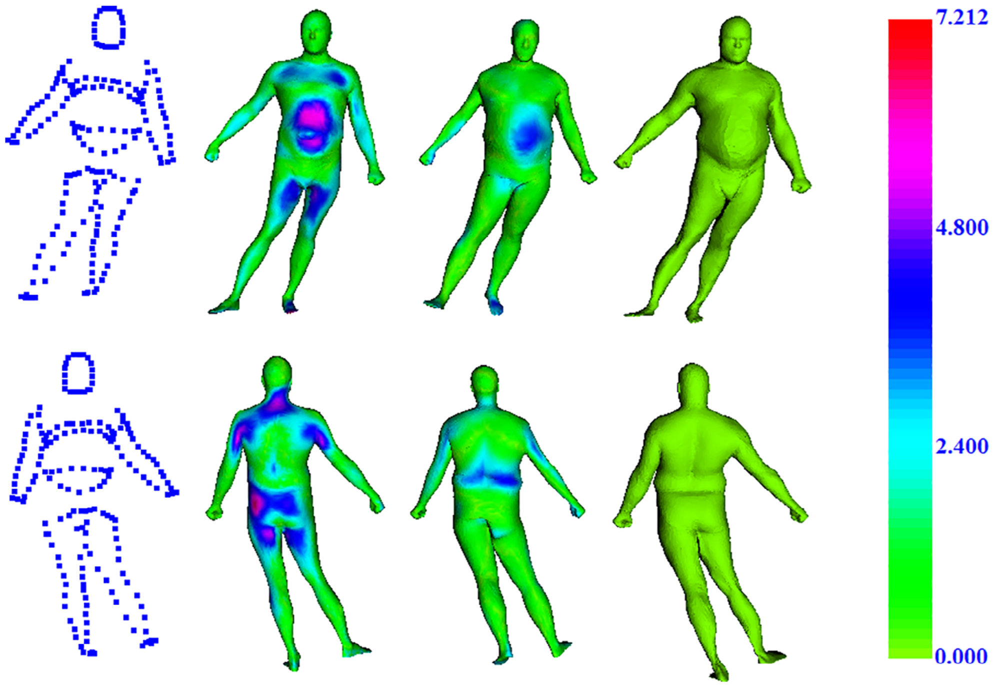

FIGURE 8 Comparison of the results. Top row is the front view, bottom row is the back view. From left to right, input points, ARAP-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.70 and 1.121, respectively. The average edge length of the ground-truth mesh is 0.0165

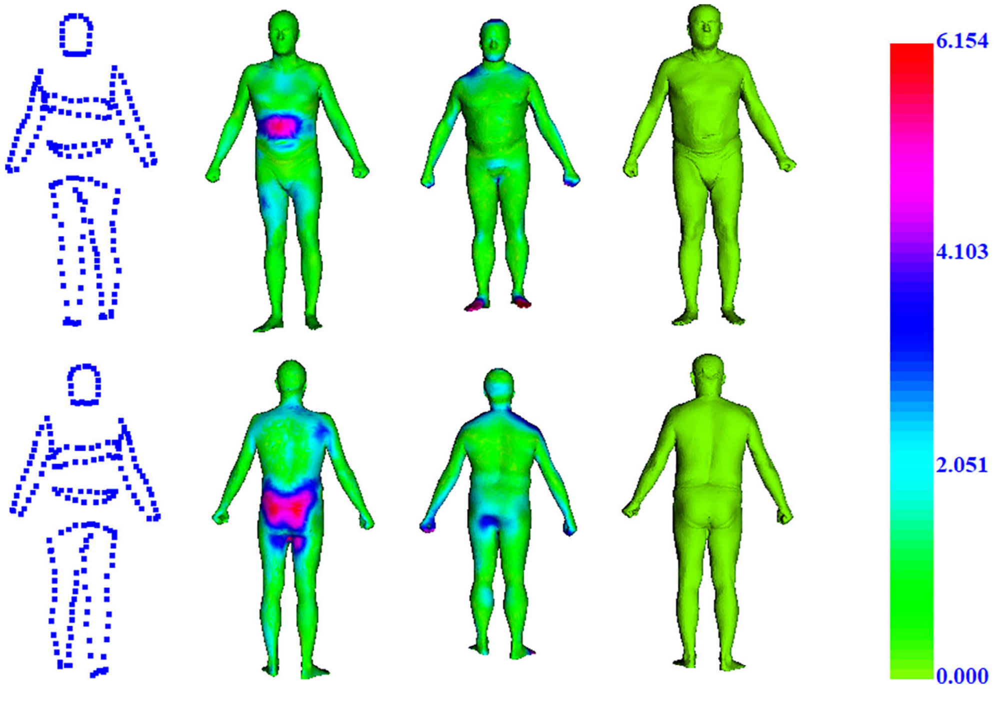

FIGURE 9 Comparison of the results. Top row is the front view, bottom row is the back view. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.154 and 1.133, respectively. The average edge length of the ground-truth mesh is 0.0156

both methods reconstructs similar outputs. This suggests that both methods are affected by large deformations in similar ways.

The number of points for each body part is also experimented in this study. The input points are completed up to 10, 20, 30, 40, 50, and 100 points by starting from different numbers of points and the requested bodies are generated from these points. The starting points are the different percentages (viz, 30%, 40%, 50%, 60%, and 70%) of the target number of points. The comparison of the different number of completed points is shown in Table 3. Most of the cases, the error decreases as the number of points increases up to 40 points. After 40 points, the error almost remains the same even if the number of points increases. Furthermore, after 50%, increasing the starting points percentage

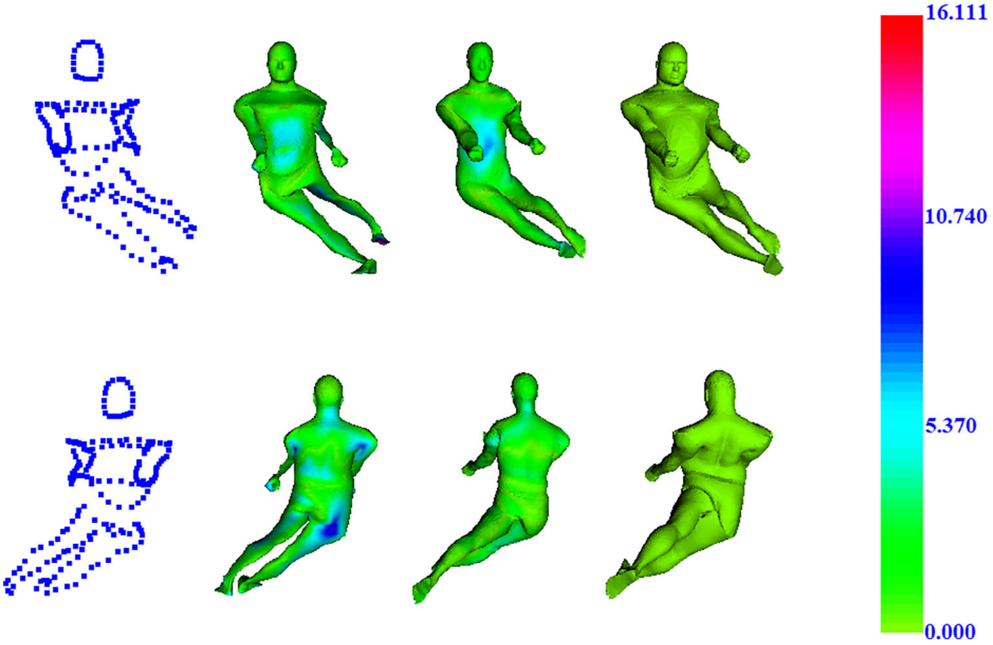

FIGURE 10 Comparison of results. Top row is the front view, bottom row is the back view. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 2.284 and 1.617, respectively. The average edge length of the ground-truth mesh is 0.0162

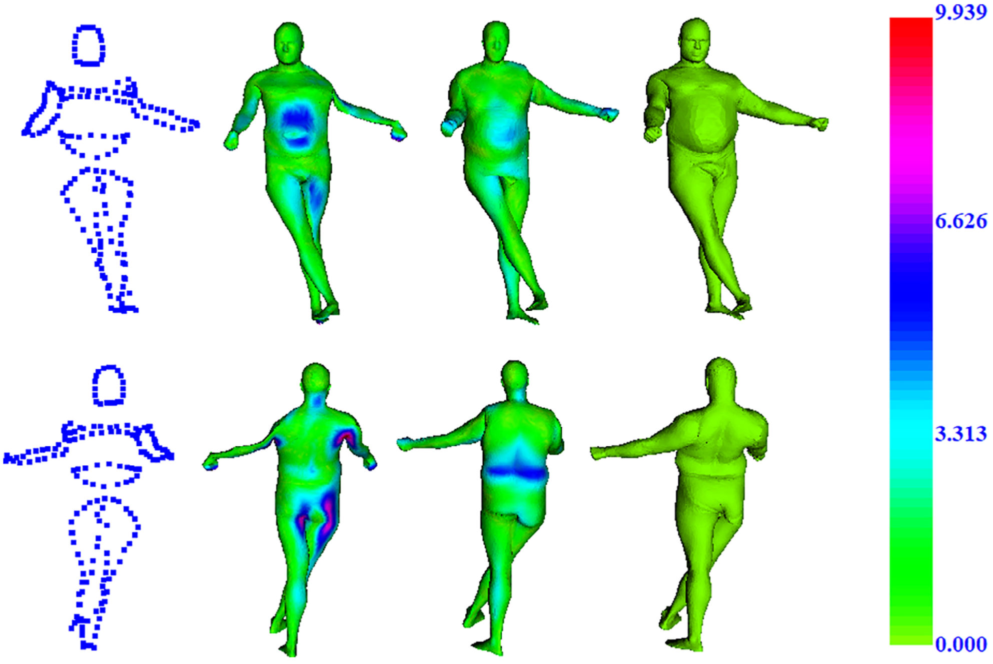

FIGURE 11 Comparison of the results for a different input. Top row is the front view, and bottom row is the back. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.853 and 1.429, respectively. The average edge length of the ground-truth mesh is 0.0163

does not change the error significantly. Considering the performance issues, we select 40 points as the most efficient number of input points. We also come to conclusion that starting from merely 20 user input points (50% of 40) is sufficient.

We also test our method with noisy data. To do so, each input point is moved in a random direction with random amount of distance which can be at most the maximum edge length of the ground-truth mesh. Sample outputs with noisy input are shown in Figures 16–18. As shown in the figures, our method is robust to noise in the input as it produces body models which are very similar to the ones produced from noiseless input points. As the noise level increases, the average error of the output increases.

Our method can also generate body models that do not even exist in our database by processing an input point set. Some parts of this set are edited manually while other parts remain the same. As shown in Figure 19, points on the belly

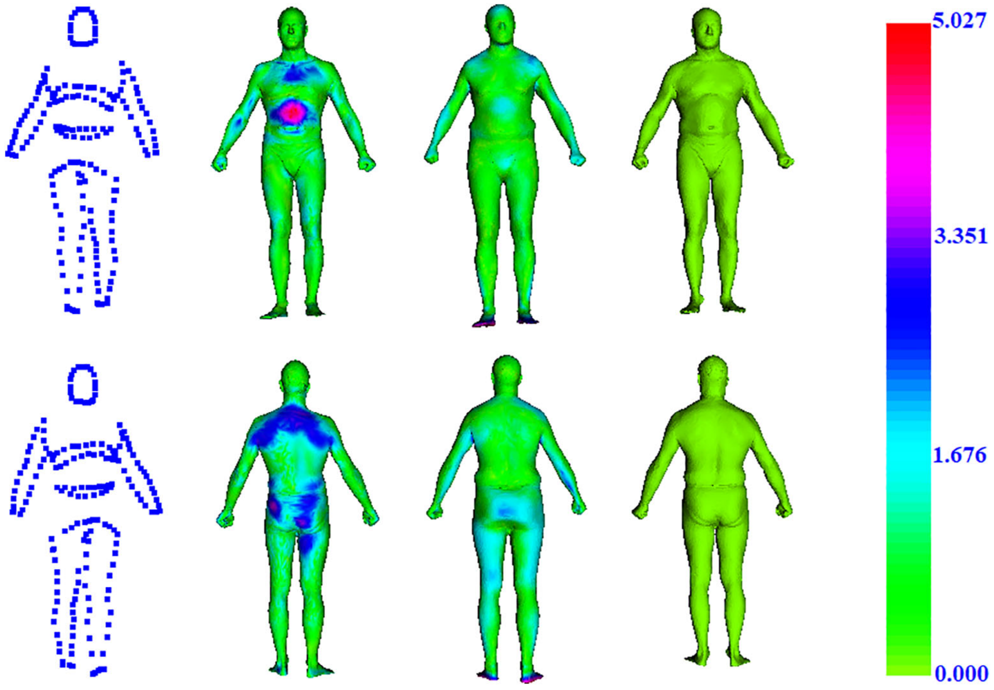

FIGURE 12 Comparison of the results for a different input. Top row is the front view, and bottom row is the back. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.01 and 1.82, respectively. The average edge length of the ground-truth mesh is 0.0148

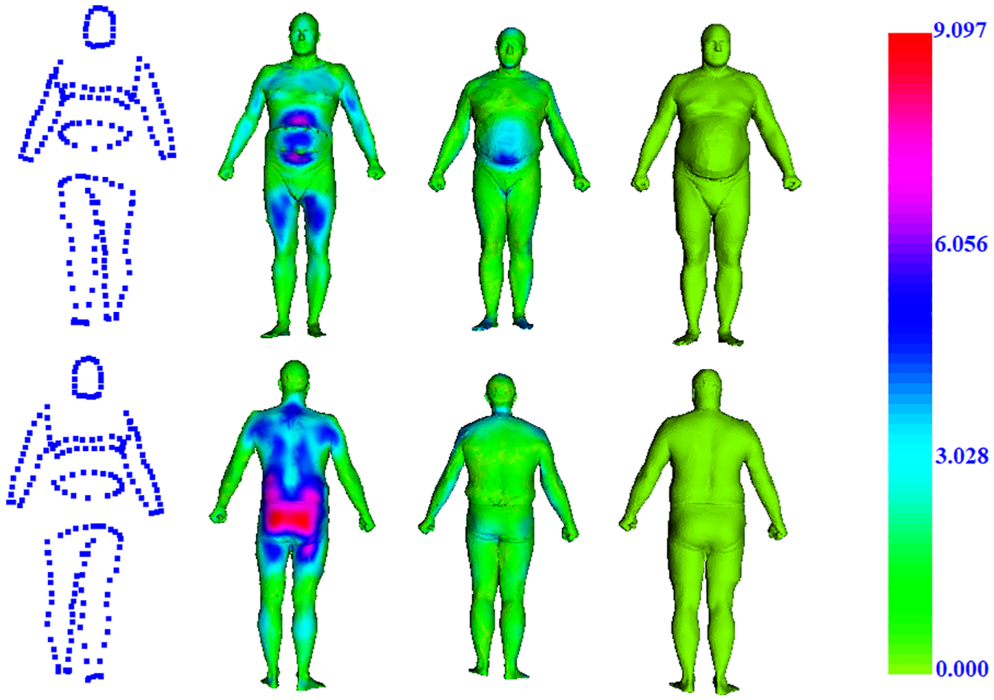

FIGURE 13 Comparison of the results for a different input. Top row is the front view, and bottom row is the back. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 0.98 and 1.403, respectively. The average edge length of the ground-truth mesh is 0.0144

and chest parts are slightly moved outward from its center so as to create a human body which has a wider belly and chest than the original one has.

4.1 | Limitations

A limitation of the proposed method is that the input points need to be on the contours of the corresponding body part. As the input points move away from the contours, e.g., due to geometric noise, the resulting reconstructed body

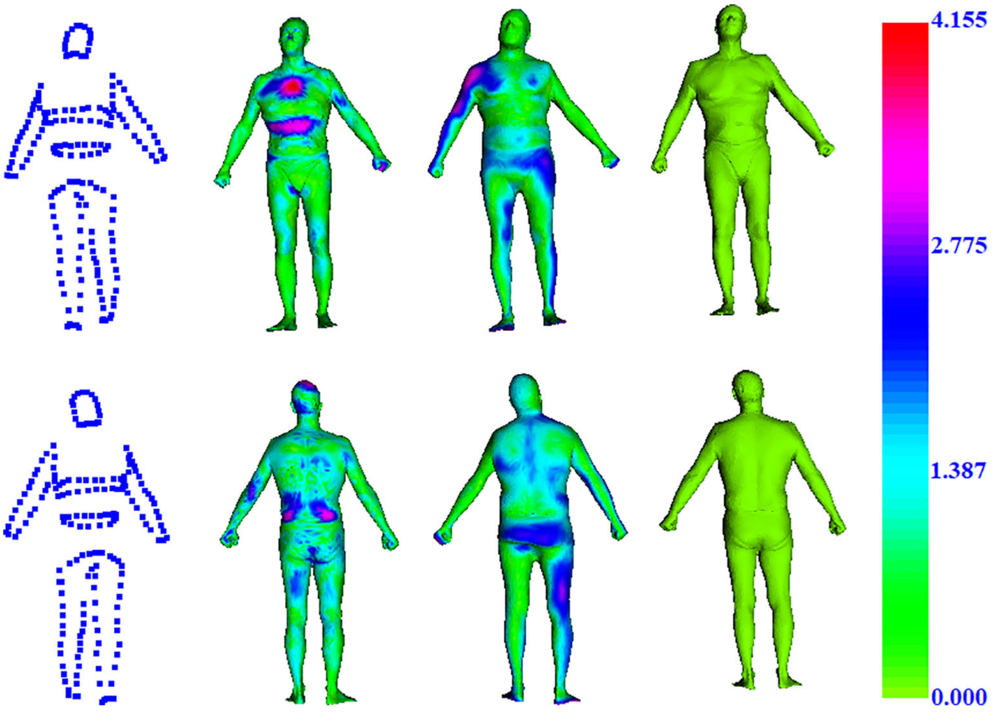

FIGURE 14 Comparison of the results for a different input. Top row is the front view, and bottom row is the back. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.303 and 1.254, respectively. The average edge length of the ground-truth mesh is 0.0142

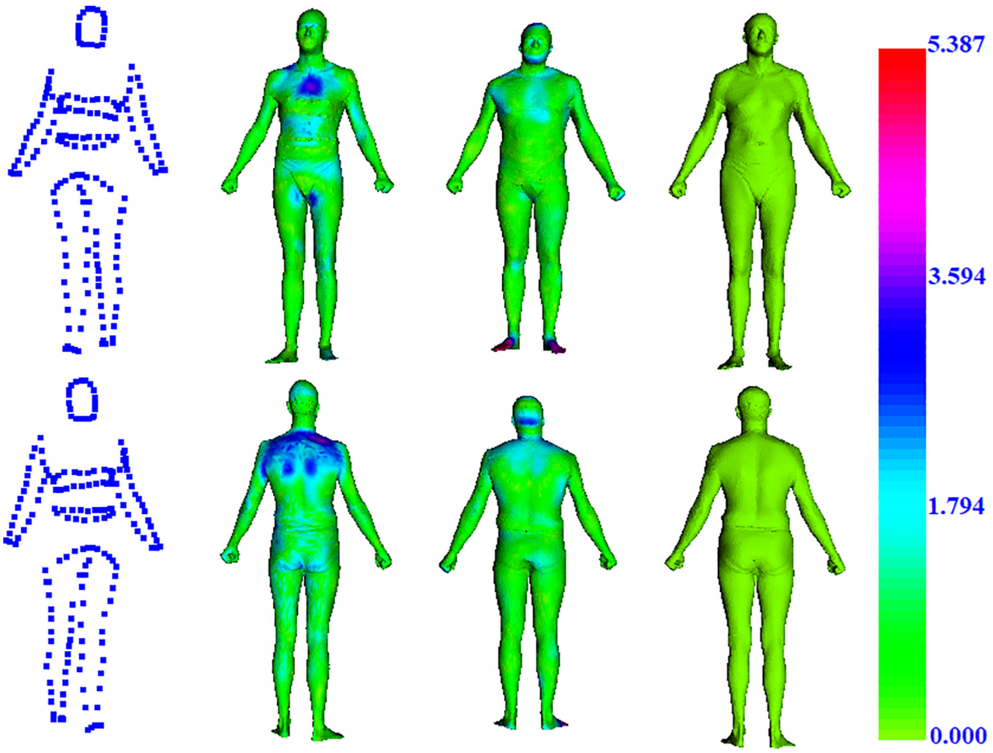

FIGURE 15 Comparison of the results for a different input. Top row is the front view, and bottom row is the back. From left to right, input points, as-rigid-as-possible-based result, result of our method and ground-truth are shown, respectively. The average errors of the second and third columns are 1.281 and 1.176, respectively. The average edge length of the ground-truth mesh is 0.0155

becomes different from the ground-truth result. This is caused by the fact that the features are designed for defining a body part as a contour, therefore, given a non-contour point set, our method produces erroneous result. The erroneous results produced from the input points which are far from the contours of the body parts are shown in Figures 20 and 21.

Another limitation is the inability to reconstruct highly articulated poses due to the one-piece arm and leg segmentations of our database. All other poses, including significant drifts without bending, can still be obtained with varying features such as height and weight.

TABLE 1 Errors produced by different methods. Our method with spline interpolation turns out to be the best performer on average taken over 50 different executions. Only 10 of them are shown in this table with pointers to their corresponding figures (first column)

Figure numbers	Our method with linear interpolation	Our method with spline interpolation	As-rigid-as-possible-based method
6	1.146	1.125	1.736
7	1.146	1.103	1.23
8	1.170	1.121	1.70
9	1.359	1.133	1.154
10	1.759	1.617	2.284
11	1.552	1.429	1.853
12	1.19	1.01	1.82
13	1.126	0.98	1.403
14	1.299	1.254	1.303
15	1.204	1.176	1.281
Average	1.198	1.152	1.607

TABLE 2 Execution times of each of our steps and our total as well as the competitor method′s total (in s)

Figure numbers	Feature matching (Section 3.4)	Rotation alignment (Section 3.5.2)	Inter-part alignment (Section 3.5.3)	Border smoothing (Section 3.5.4)	Our method total	ARAP-based total
6	1.197	2.043	0.47	5.02	8.73	9.3
7	1.164	2.314	0.469	4.97	8.917	9.65
8	1.172	2.145	0.468	4.95	8.735	9.72
9	1.189	2.095	0.472	5.05	8.806	9.47
10	1.175	2.243	0.471	4.98	8.869	9.82
11	1.166	2.145	0.467	5.01	8.78	9.52
12	1.159	2.211	0.420	4.36	8.15	9.16
13	1.163	2.144	0.408	4.61	8.325	9.97
14	1.188	2.042	0.451	4.99	8.671	9.17
15	1.178	2.255	0.759	4.56	8.752	9.44

TABLE 3 Errors by number of points after point completion

	10 Points	20 Points	30 Points	40 Points	50 Points	100 Points
30%	2.035	1.326	1.264	1.222	1.222	1.222
40%	1.39	1.354	1.25	1.18	1.183	1.181
50%	1.27	1.319	1.243	1.18	1.18	1.18
60%	1.25	1.25	1.243	1.153	1.151	1.151
70%	1.243	1.229	1.243	1.153	1.153	1.153

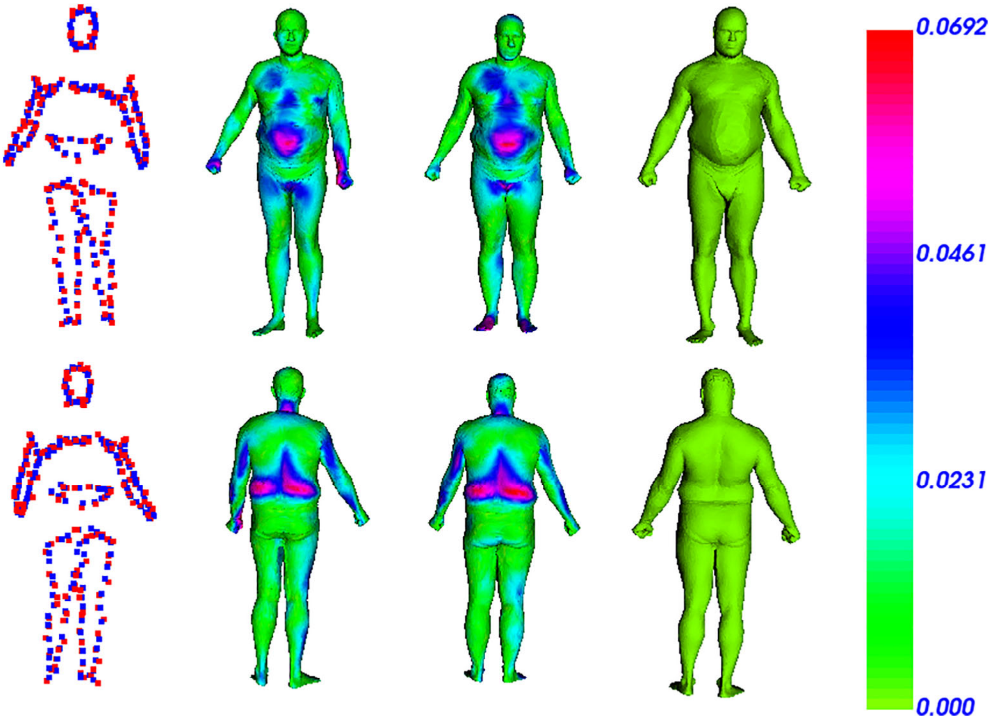

FIGURE 16 Comparison of results under geometric noise. Top row is the front view, bottom row is the back view. From left to right, input points (noisy input colored as red, original input colored as blue), result of noisy input, result of original input and ground-truth are shown, respectively. Noise is generated from Gaussian distribution with standard deviation of 0.8. The average errors of the second and third columns are 1.243 and 1.125, respectively. The average edge length of the ground-truth mesh is 0.0144

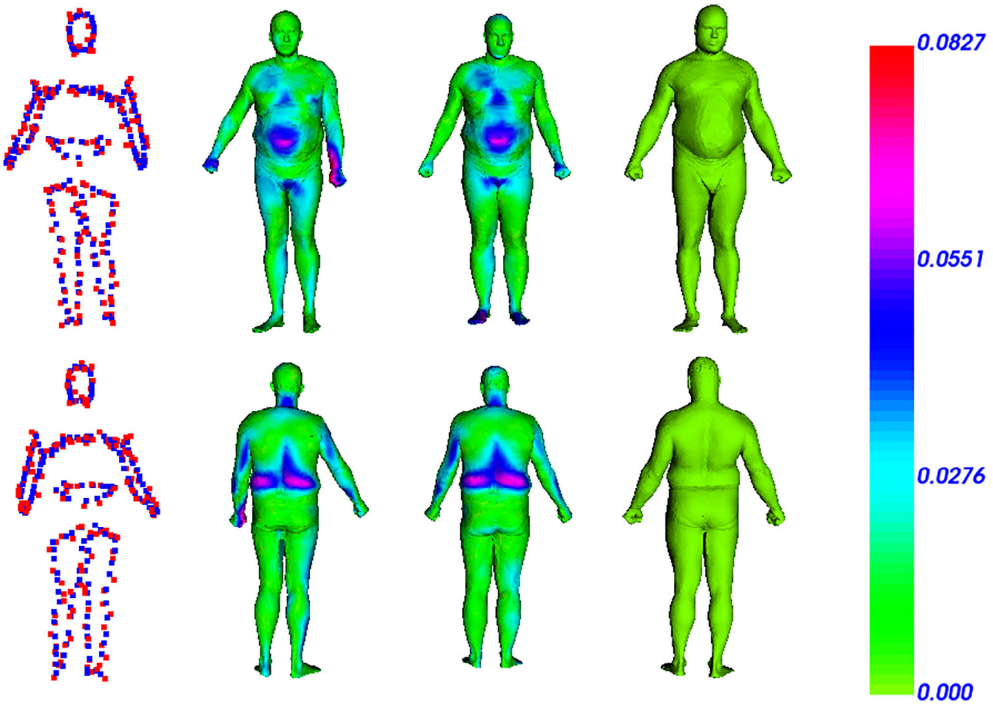

FIGURE 17 Comparison of results under geometric noise. Top row is the front view, bottom row is the back view. From left to right, input points (noisy input colored as red, original input colored as blue), result of noisy input, result of original input and ground-truth are shown, respectively. Noise is generated from Gaussian distribution with *SD* of 1. The average errors of the second and third columns are 1.285 and 1.125, respectively. The average edge length of the ground-truth mesh is 0.0144

5 | CONCLUSION

In computer graphics, one of the most studied field is human body modeling. In order to create human body models, different approaches use different inputs such as single RGB-D depth frames,[17] multiple depth frames,[22] morphable template models and landmarks,[25] and dense 3D point clouds.[41] Differently from previous studies, in this paper, a data-driven method which reconstructs human body from sparse input points is proposed.

The comparative evaluations have shown that proposed method is fast and successful at generating human body from limited number of points. The proposed method is able to reconstruct bodies in different shapes and different poses,

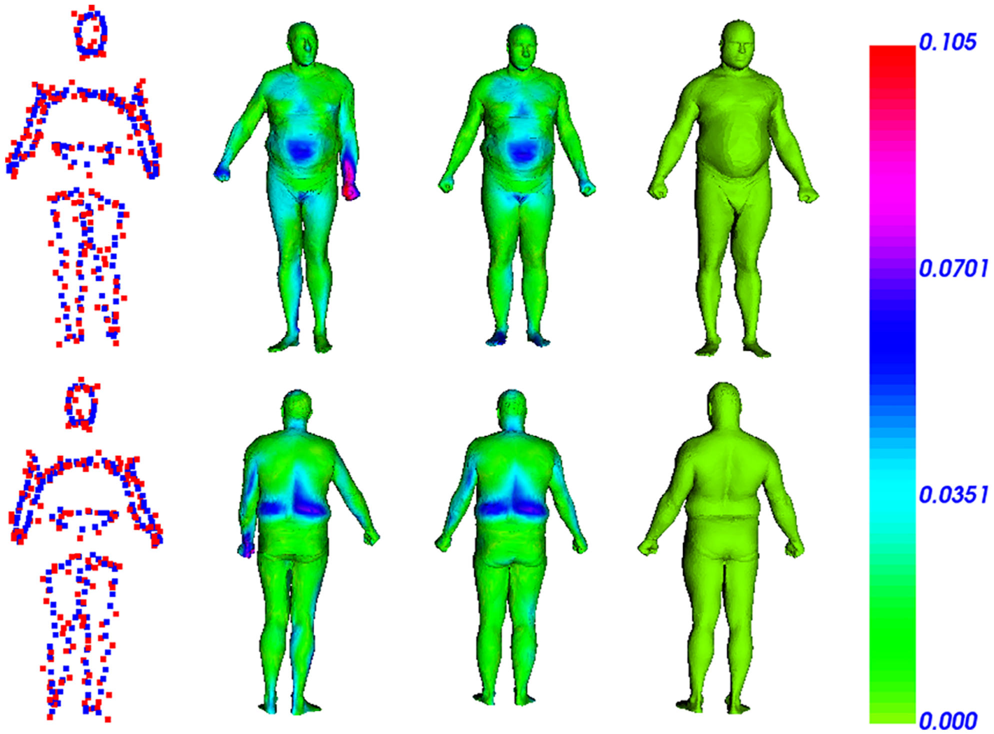

FIGURE 18 Comparison of results under geometric noise. Top row is the front view, bottom row is the back view. From left to right, input points (noisy input colored as red, original input colored as blue), result of noisy input, result of original input and ground-truth are shown, respectively. Noise is generated from Gaussian distribution with *SD* of 1.5. The average errors of the second and third columns are 1.458 and 1.125, respectively. The average edge length of the ground-truth mesh is 0.0144

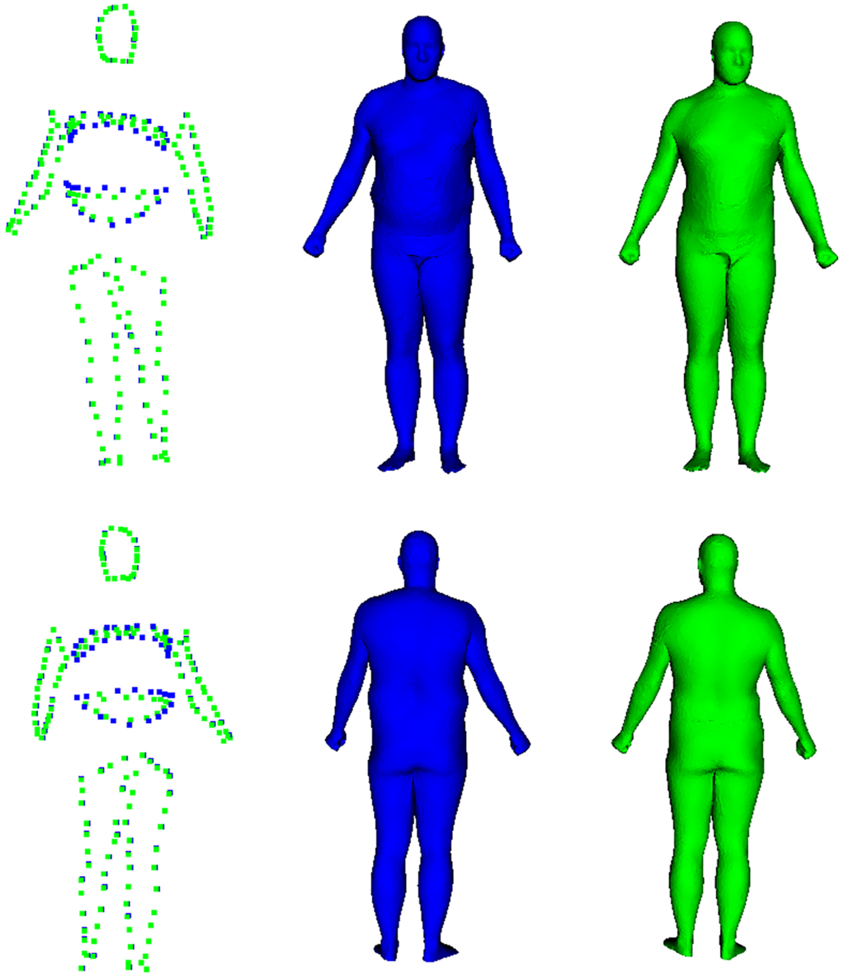

FIGURE 19 Illustration of generating a new body by manual editing. Top row is the front view, bottom row is the back view. From left to right, input points (manually edited input colored as blue, original input colored as green), result of manually edited input and result of original input shown, respectively. Notice the slight expansion in belly and chest of the blue model

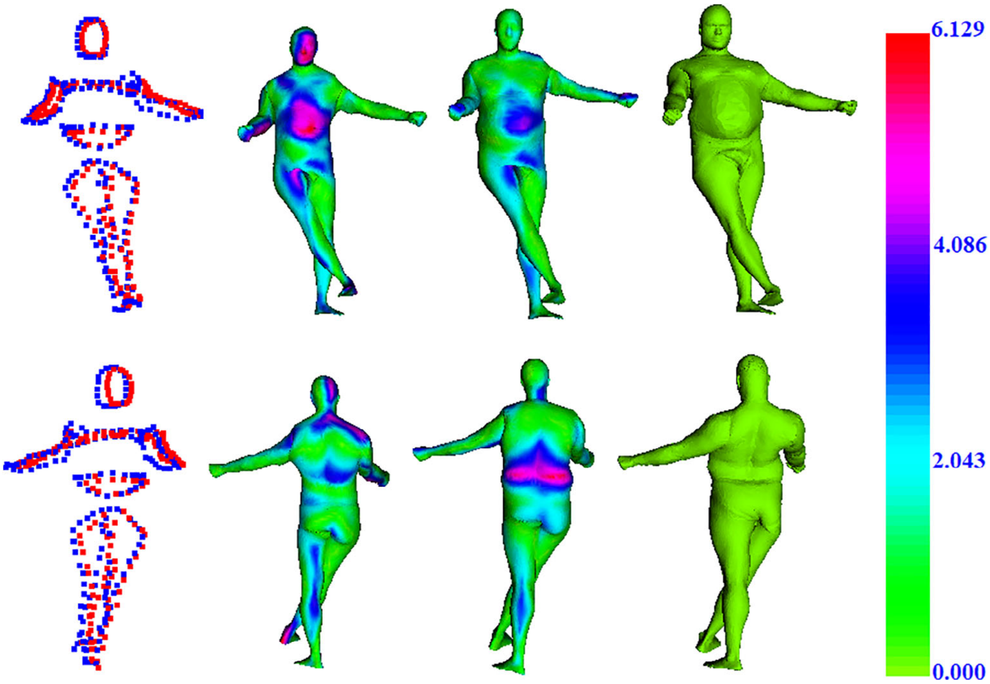

FIGURE 20 Comparison of the results produced from the input points which are far from the contours and from the input points which are around the contours. Top row is the front view, bottom row is the back view. From left to right, input points (red one represents the input points which are far from the contours and blue one represents the input points which are around the contours), result of red input points, result of blue input points and ground-truth are shown, respectively. The average errors for model in the second and third column are 2.083 and 1.569, respectively

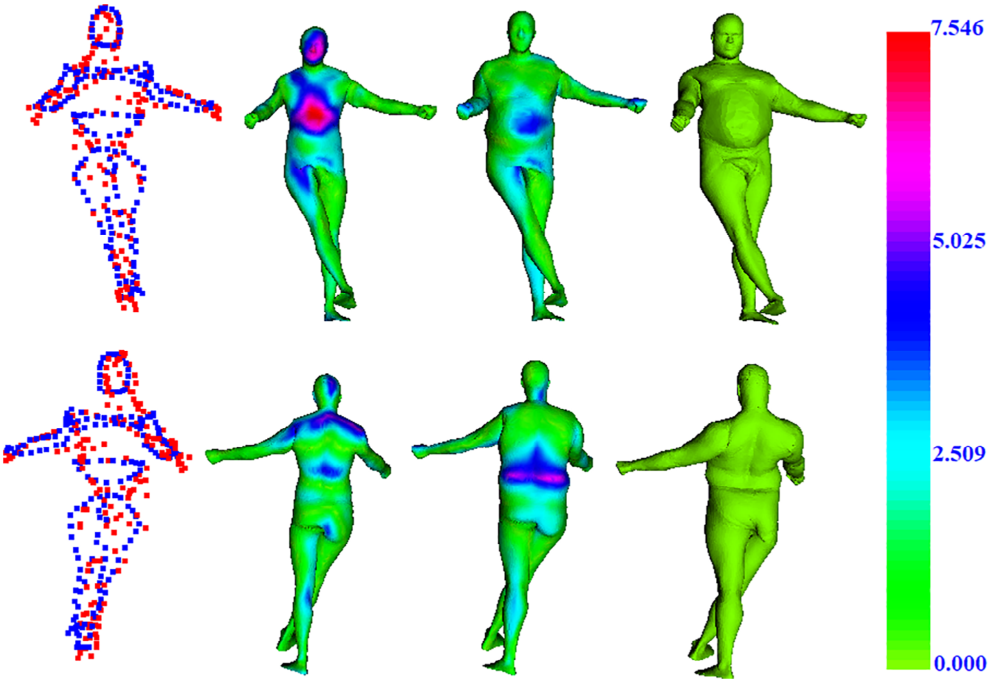

FIGURE 21 Comparison of the results produced from the input points which are far from the contours and from the input points which are around the contours. Top row is the front view, bottom row is the back view. From left to right, input points (red one represents the input points which are far from the contours and blue one represents the input points which are around the contours), result of red input points, result of blue input points and ground-truth are shown, respectively. The average errors for model in the second and third column are 2.639 and 1.433, respectively.

including non-rigid bending, thanks to the novel point set features and deformation support. It has shown resilience to geometric noise and potential for the synthesis of brand new models.

As a future work, our method can be extended so that the bodies are reconstructed from 2D images by combining our method and the methods like the one proposed in.[13] Manual editing that lead to brand new human body generations can also be replaced by an automated process, for example, allow movement of input points in a particular pattern.

DATA AVAILABILITY STATEMENT

There is no data available.

ORCID

Yusuf Sahillioğlu https://orcid.org/0000-0002-7997-4232

REFERENCES

1. Jones PR, Rioux M. Three-dimensional surface anthropometry: applications to the human body. Opt Lasers Eng. 1997;28(2):89–117.
2. Fuller N, Jebb S, Laskey M, Coward W, Elia M. Four-component model for the assessment of body composition in humans: comparison with alternative methods, and evaluation of the density and hydration of fat-free mass. Clin Sci. 1992;82(6):687–93.
3. Batouche M. A knowledge based system for diagnosing spinal deformations: moiré pattern analysis and interpretation. Proceedings of the 1992 11th IAPR International Conference on Pattern Recognition. The Hague, Netherlands: IEEE; 1992. p. 591–4.
4. Hodgins JK, Wooten WL, Brogan DC, O'Brien JF. Animating Human Athletics. Atlanta, GA: Georgia Institute of Technology; 1995.
5. Fua P, Gruen A, Plänkers R, D'Apuzzo N, Thalmann D. Human body modeling and motion analysis from video sequences. Proceedings of the ISPRS Commission V Symposium Swiss Federal Institute of Technology, Institute of Geodesy and Photogrammetry, Hakodate, Japan: 1998.
6. Mehta D, Sotnychenko O, Mueller F, Xu W, Elgharib M, Fua P, et al. XNect: real-time multi-person 3D motion capture with a single RGB camera. ACM Trans Graph. 2020;39(4):82:1–82:17.
7. De Aguiar E, Stoll C, Theobalt C, Ahmed N, Seidel HP, Thrun S. Performance capture from sparse multi-view video. ACM SIGGRAPH 2008 papers; New York City, United States: Association for Computing Machinery; 2008. p. 1–10.
8. Persson A, Andersson M, Oden A, Sandborgh-Englund G. A three-dimensional evaluation of a laser scanner and a touch-probe scanner. J Prosthet Dent. 2006;95(3):194–200.
9. Sahillioğlu Y. Coarse-to-fine surface reconstruction from silhouettes and range data using mesh deformation. Vis Comput. 2020;36:1705–21.
10. Glauser O, Panozzo D, Hilliges O, Sorkine-Hornung O. Deformation capture via soft and stretchable sensor arrays. ACM Trans Graph. 2019;38(2):16.
11. Yang KJ, Sun CT. High precision 3D point cloud with modulated pulses for LiDAR system. Advances in intelligent information hiding and multimedia signal processing. New York, NY: Springer; 2017. p. 217–24.
12. von Marcard T, Rosenhahn B, Black MJ, Pons-Moll G. Sparse inertial poser: automatic 3d human pose estimation from sparse imus. Comput Graph Forum. 2017;36:349–60.
13. Zhao R, Wang Y, Benitez-Quiroz CF, Liu Y, Martinez AM. Fast and precise face alignment and 3d shape reconstruction from a single 2d image. Proceedings of the European conference on computer vision. New York, NY: Springer; 2016. p. 590–603.
14. Ballan L, Taneja A, Gall J, Van Gool L, Pollefeys M. Motion capture of hands in action using discriminative salient points. European conference on computer vision. New York, NY: Springer; 2012. p. 640–53.
15. Bogo F, Kanazawa A, Lassner C, Gehler P, Romero J, Black MJ. Keep it SMPL: automatic estimation of 3D human pose and shape from a single image. Proceedings of the European conference on computer vision New York, NY: Springer; 2016. p. 561–578.
16. Kocabas M, Karagoz S, Akbas E. Self-supervised learning of 3d human pose using multi-view geometry. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, United States: 2019. p. 1077–86.
17. Liang S, Kemelmacher-Shlizerman I, Shapiro LG. 3d face hallucination from a single depth frame. Proceedings of the 2014 2nd International Conference on 3D Vision, vol. 1 IEEE, Washington DC, United States; 2014. p. 31–38.
18. Gower JC. Generalized procrustes analysis. Psychometrika. 1975;40(1):33–51.
19. Allen B, Curless B, Curless B, Popović Z. The space of human body shapes: reconstruction and parameterization from range scans. ACM Trans Graph. 2003;22:587–94.
20. Atmosukarto I, Shapiro LG, Heike C. The use of genetic programming for learning 3d craniofacial shape quantifications. Proceedings of the 20th International Conference on Pattern Recognition. IEEE, Washington DC, United States; 2010. p. 2444–7.
21. Mercan E, Shapiro LG, Weinberg SM, Lee SI. The use of pseudo-landmarks for craniofacial analysis: a comparative study with l 1-regularized logistic regression. Proceedings of the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, Osaka, Japan; 2013. p. 6083–6.
22. Bondi E, Pala P, Berretti S, Del Bimbo A. Reconstructing high-resolution face models from depth sequences acquired in uncooperative contexts. Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), vol. 7 IEEE, Ljubljana, Slovenia; 2015. p. 1–6.
23. Myronenko A, Song X. Point set registration: coherent point drift. IEEE Trans Pattern Anal Mach Intell. 2010;32(12):2262–75.
24. Frasconi P, Silvestri L, Soda P, Cortini R, Pavone FS, Iannello G. Large-scale automated identification of mouse brain cells in confocal light sheet microscopy images. Bioinformatics. 2014;30(17):i587–93.
25. Achenbach J, Waltemate T, Latoschik ME, Botsch M. Fast generation of realistic virtual humans. Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology. New York, NY: ACM; 2017. p. 12.
26. Horn BK. Closed-form solution of absolute orientation using unit quaternions. Josa a. 1987;4(4):629–42.
27. Buss SR. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. IEEE J Robot Automat. 2004;17(1–19):16.
28. Yemez Y, Sahillioğlu Y. Shape from silhouette using topology-adaptive mesh deformation. Pattern Recogn Lett. 2009;30(13):1198–207.
29. Groueix T, Fisher M, Kim VG, Russell BC, Aubry M. 3d-coded: 3d correspondences by deep deformation. Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany; 2018. p. 230–46.

30. Tong J, Zhou J, Liu L, Pan Z, Yan H. Scanning 3d full human bodies using kinects. IEEE Trans Vis Comput Graph. 2012;18(4):643–50.
31. Li H, Adams B, Guibas LJ, Pauly M. Robust single-view geometry and motion reconstruction. ACM Trans Graph. 2009;28(5):1–10.
32. Chang W, Zwicker M. Global registration of dynamic range scans for articulated model reconstruction. ACM Trans Graph. 2011;30(3):1–15.
33. Zuffi S, Black MJ. The stitched puppet: a graphical model of 3d human shape and pose. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, United States; 2015. p. 3537–46.
34. Joo H, Simon T, Sheikh Y. Total capture: a 3d deformation model for tracking faces, hands, and bodies. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, United States; 2018. p. 8320–9.
35. Loper M, Mahmood N, Black MJ. MoSh: motion and shape capture from sparse markers. ACM Trans Graph. 2014;33(6):1–13.
36. Guo Y, Zhang J, Cai L, Cai J, Zheng J. *Self-supervised CNN for unconstrained 3D facial performance capture from an RGB-D camera*. arXiv preprint arXiv:180805323; 2018.
37. Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. SIGGRAPH'99. Association for Computing Machinery: New York City, United States; 1999;99:187–94.
38. Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J, Davis J. SCAPE: shape completion and animation of people. ACM Trans Graph. 2005;24:408–16.
39. Anguelov D, Srinivasan P, Pang HC, Koller D, Thrun S, Davis J. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. Advances in neural information processing systems; San Francisco, United States: Morgan Kaufmann Publishers Inc; 2005. p. 33–40.
40. Miller C, Arikan O, Fussell D. Frankenrigs: building character rigs from multiple sources. Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. New York, NY: ACM; 2010. p. 31–8.
41. Chen Q, Koltun V. Robust nonrigid registration by convex optimization. Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile; 2015. p. 2039–47.
42. Huang QX, Adams B, Wicke M, Guibas LJ. Non-rigid registration under isometric deformations. Comput Graph Forum. 2008;27:1449–57.
43. Bronstein AM, Bronstein MM, Kimmel R. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. Proc Natl Acad Sci. 2006;103(5):1168–72.
44. Sahillioğlu Y. A genetic isometric shape correspondence algorithm with adaptive sampling. ACM Trans Graph. 2018;37(5):175.
45. Sahillioğlu Y, Yemez Y. Recent advances in shape correspondence. Comput Vis Image Understand. 2010;114(3):334–48.
46. Sahillioğlu Y, Kavan L. Skuller: a volumetric shape registration algorithm for modeling skull deformities. Med Image Anal. 2015;23(1):15–27.
47. Maron H, Dym N, Kezurer I, Kovalsky S, Lipman Y. Point registration via efficient convex relaxation. ACM Trans Graph. 2016;35(4):73.
48. Bogo F, Romero J, Loper M, Black MJ. FAUST: dataset and evaluation for 3D mesh registration. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, United States; 2014. p. 3794–801.
49. Nealen A, Igarashi T, Sorkine O, Alexa M. Laplacian mesh optimization. Proceedings of the 4th international conference on computer graphics and interactive techniques in Australasia and Southeast Asia; 2006. p. 381–9.
50. Taubin G. A signal processing approach to fair surface design. Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. New York, NY: ACM; 1995. p. 351–8.
51. Sorkine O. Differential representations for mesh processing. Comput Graph Forum. 2006;25(4):789–807.
52. Meyer M, Desbrun M, Schröder P, Barr AH. Discrete differential-geometry operators for triangulated 2-manifolds. Visualization and mathematics III. New York, NY: Springer; 2003. p. 35–57.
53. Jourabloo A, Liu X. Pose-invariant 3D face alignment. Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile; 2015. p. 3694–702.
54. Igarashi T, Moscovich T, Hughes JF. As-rigid-as-possible shape manipulation. ACM Trans Graph. 2005;24(3):1134–41.

How to cite this article: Taştan O, Sahillioğlu Y. Human body reconstruction from limited number of points. *Comput Anim Virtual Worlds*. 2021;1–20. https://doi.org/10.1002/cav.1995