Data-Driven Human Modeling by Sparse Representation

Yiu-Bun Wu[a], Bin Liu[a,b], Xiuping Liu[b], Charlie C.L. Wang[a,*]

[a]*Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, P.R. China*
[b]*School of Mathematical Sciences, Dalian University of Technology, Dalian, P.R. China*

Abstract

Data-driven methods for modeling the realistic shapes of 3D human bodies need to access datasets that contain a large amount of 3D human models. We develop a method based on sparse representation in this paper to represent 3D human models as signals of patches. Unlike the general mesh compression approaches, all mesh models used in a data-driven human modeling framework have the same mesh connectivity. By using this property, we segment a human model into patches containing the same number of vertices. L_0-learning algorithm is selected to train an overcomplete dictionary matrix, which in turn introduces sparse representation of the dataset. Patch signals of individual human models can then be extracted by using the dictionary matrix. With the ease of balance control between sparsity and accuracy that is featured by the chosen learning algorithm, a representation with high compression ratio and low shape-approximation error can be determined. The results have been compared with the widely used statistic representation based on *principal component analysis* (PCA) to verify the effectiveness of our approach. Moreover, the method for using sparse representation in the regression-based statistical modeling of 3D human models has been presented at the end of the paper.

Keywords: Human Models, Sparse Representation, Regression, Data-Driven Modeling

1. Introduction

Modeling 3D human models has become more and more important for the *computer-aided design* (CAD) systems in the applications of designing and fabricating personalized products. Data-driven methods have been proved to be the most effective way to generate realistic 3D human models from the dataset with mesh models captured from real individuals by ranged images [1, 2, 3]. Recent development of dense data acquisition systems can capture dynamic 3D human bodies as triangular meshes in high resolution at tens of frames per second (e.g., [4]). Apparel industry has started to consider using time-variant 3D human models (i.e., 4D data) in the design of functional garments.

The dataset of 3D / 4D human models usually consumes a lot of storage when representing each model by a triangular mesh. For example, a dataset consists of 50 male models with 10 different motion sequences can generate up to 34GB uncompressed data when each motion sequence contains 100 sampled poses and each pose is represented as a triangular mesh with 30k vertices. The efficient access to this kind of datasets by less storage / communication consumption becomes more and more important as demanded.

When applying 3D human models to e-commerce applications (such as virtual fitting or customized clothes design and fabrication) on the platform of mobile phone, the demand on a memory-efficient representation of 3D human models becomes more critical. The typical download speed of 4G is less than 40MB or even 20MB per second in many scenarios. The much smaller memory of a mobile phone comparing to the desktop computer also requires a memory efficient representation of human models. Motivated by this reason, the problem of investigating a new representation for large datasets of 3D human models is investigated in this paper. Apart from the storage size, accuracy of the derived representation to reconstruct the original data is also a crucial factor to be considered.

We propose to conduct sparse representation for data-driven 3D human modeling. In sparse representation, a matrix spanning linear space named as *dictionary* is trained to be composed of a number of clustering vectors called *atoms* [5]. In addition to the corresponding representation coefficients with sparsity, the linear combinations of the weighted atoms can reconstruct every training signal of the original dataset [5, 6]. We propose to derive sparse representation for 3D human model datasets, by considering its capability to represent large size datasets composed of human bodies with wide range of shape features.

Our study finds that L_0-norm dictionary learning is the most appropriate algorithm used to generate sparse representation for human model datasets. To employ this learning algorithm, we develop a signal extraction method to obtain patch signals which are compatible to dictionary learning. Specifically, the extraction method constructs patches cover the whole body of 3D human models by using *centroidal Voronoi tessellation* (CVT) [7]. When all patches having the same number of vertices, the positions of vertices are sorted to form the signals for sparse representation learning. With the help of learning outcome, dictionary of signals can be obtained to provide the

*Corresponding author
Email address: cwang@mae.cuhk.edu.hk (Charlie C.L. Wang)

function of accurately reconstructing the original human models. Experiments are conducted to verify the effectiveness of our approach.

The following contributions are made by our work.

- While the majority of research in sparse representation is based on the datasets of images and audio signals, we extend the use of sparse representation to 3D human model dataset and its application in 3D human modeling.
- An approach is developed to construct patches on the mesh surfaces of 3D human models in a dataset, so that a large amount of training signals can be extracted to fulfill the requirement of dimensions for sparse representation learning.
- The compression level with respect to the data size of the original dataset can be conveniently specified for training with the help of our carefully designed pipeline for signal extraction and learning algorithm selection.

With the help of technology developed in this work, we can use sparse representation to represent more *principal components* (PCs) for statistical human modeling with a compact data size. Moreover, preliminary study of using the dictionary learned from a dataset of human models to synthesize general 3D freeform models is encouraging – models with totally different mesh topology can be successfully reconstructed.

The rest of our paper is organized as follows. After reviewing the most relevant work in Section 2, we explain sparse representation and study different learning algorithms in Section 3. Section 4 focuses on the extraction of patch signals and the synthesis of 3D human models. We then apply the sparse representation in statistical 3D human modeling in Section 5. Experimental results are given in Section 6, where both the effective compression and the modeling by regression will be studied. Lastly, our paper ends with a section of conclusion and discussion.

2. Related Work

In this section, we review the existing research work in algorithms of sparse representation , data-driven geometric modeling and data-driven human modeling. Some relevant applications are also discussed.

2.1. Sparse representation

Generally, the learning process for sparse representation can be divided into two phases: sparse coding and dictionary updating. The sparse coding phase allows the signals to be represented in a higher dimensional space. As a result, it makes the representation of signals more flexible. The dictionary updating phase aims at finding a better dictionary in which the training signals admit a more sparse representation. Solution can be found by applying minimization methods alternatively in these two phases.

Sparse representation has obtained significant success in computer vision applications, such as image restoration [5, 8, 9], face recognition [10, 11], object tracking [12, 13] and abnormal event detection [14]. It is mainly due to the fact that image signals have naturally sparse representations with respect to fixed bases. For example, Aharon et al. [5] generalized K-means clustering process as K-SVD to design an overcomplete dictionary and the sparse coefficient matrix for representing a training dataset. By constraining the number of non-zeros (e.g., by L_0-norm) in each column of the coefficient matrix, the K-SVD algorithm can search for the best representation. In order to consume less memory or obtain faster convergence, more variants of learning algorithms have been developed for sparse representation. Different from handling the whole training dataset at each iteration in the batch approaches, Mairal et al. [6] developed an online learning algorithm that only processes a small subset of the dataset at each time by using stochastic approximation. This makes it possible for dictionary learning to train a dataset with huge size. Instead of the traditional single dictionary learning, Chen et al. [15] proposed to conduct multiple dictionary learning by intrinsic component analysis to detect and remove shadows in images.

2.2. Data-driven geometric modeling

Thanks to the great success of sparse representation in computer vision, researchers have tried to apply sparse representation to geometry processing tasks such as shape retrieval and surface reconstruction. Gal and Cohen-Or [16] applied sparse representation to extract interesting and salient geometric features and then use these features to query similar shape from database. Aiming at the drawback from the unawareness of pooling by the bag-of-feature learning [17, 18], Litman et al. [19] replaced the vector quantization by sparse coding, dictionary learning of which is based on the results of pooling by optimization. Such supervised learning is beneficial by its capability of minimizing reconstruction errors during the retrieval of models by their shapes. Xiong et al. [20] utilized sparse representation to implement surface reconstruction from point cloud. Dictionary is used to store the vertices of triangular mesh and sparse coefficient matrix encodes the connectivity of the mesh. Wei et al. [21] conducted low-rank matrix recovery with kernel based approach to remove noises on scanned 3D mesh surfaces. It identifies regions of irregular shapes on a mesh surface while maintaining the correlation of its normal vectors after denoising.

More applications could be found in the survey paper [22], which analyzes the difficulties of applying sparse representation on geometric modeling caused by the problems in 1) regularity of domain, 2) definition of basis functions and 3) geometric regularization. It also shows the great potential that sparse representation can become an effective tool for geometric modeling. However, less research is investigated in literature to model 3D human bodies in a dataset by sparse representation. The major difficulty is how to design regular signals from those human models and how to reasonably use these signals.

2.3. Data-driven human modeling

Principal component analysis (PCA) is a classical approach which can be used for human database representation based on

statistical analysis (ref. [1, 2, 23, 24, 25, 26]). Starting from a range of 3D body scans with markers, Allen et al. [1] developed a parameterization method based on PCA to fit template meshes to the scans by non-linear optimization with constraints by the proximity between the template vertices and scan markers. Afterwards, researchers [2, 27, 25] took more attention to the strategy that the shape-related and pose-related deformations are learned separately and combined later in the synthesis step. The SCAPE method proposed in [2] learned the difference in body shape by PCA and the change in poses among human models by articulated deformation model. The multi-person linear model [25] is designed to represent shape-and-posture variations in the vertex space. PCA based statistic representation with normalization of poses can also be learned from human model dataset with massive amount of raw body scans [26]. However, due to the basic nature of PCA that every principal component has to be orthogonal to each other in the feature space, reconstruction accuracy may be limited by the selected number of principal components.

The other type of methods intend to perform simultaneous analysis of shape and posture variations (ref. [28, 29, 30]). These methods learned skinning weights for corrective enveloping of posture-related shape variations, which allows to explore both shape and posture variations using a single shape space. The approach in [28] captured the correlations between identity and the pose of human body's shape variation. It leads to a hybrid representation to include a latent variable model with keys of interpolations among different human models. Hasler et al. [29] introduced a model that is learned from a large dataset of 3D human scans with different shapes and poses. Muscle deformation can be simulated in this approach.

Researchers also studied the correlation between semantic parameters such as body measurements and the statistical representation with respect to the shape space of human models [31, 32, 33, 34, 35, 36]. Yang et al. [31] used linear regression to learn the mapping. It is found that overfitting can easily occur while working with PCA based representation. Regarding the inadequacy of PCA approach for representing the variation of shapes in high-dimensional space, Huang et al. [35] proposed to tackle the problem in local patches by deep neural network. The regression model has also been employed in a hardware device different from camera system – by a smart vest to capture the features of a human body [36]. The 3D human models are reconstructed from the captured features by a multi-layer neural network.

In the community of computer-aided design, techniques of parametric human modeling have been developed to support the design of human-centered products (e.g., [3, 37]). When the data acquisition can be conducted more and more easily on portable devices (e.g., depth cameras on mobile phone), data-driven 3D human modeling becomes a key technique to bridge the personalized demand of consumers and the manufacturing suppliers with the help of machine learning [38, 35]. Moreover, statistics-based modeling is also used for the automatic meshing in the finite element analysis [39, 40] and the computation re-use in isogeometric analysis [41]. However, no prior work is found in literature to conduct sparse representation for data-driven 3D human modeling.

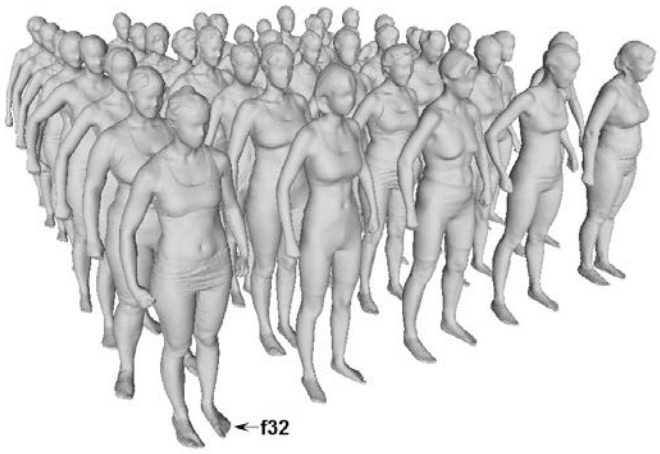


Figure 1: A training dataset consists of 40 female models – all represented as triangular meshes with the same connectivity, which however are different from the male models.

3. Preliminary

This preliminary section first presents the dataset of human models. Then, the sparse representation is briefly introduced together with its requirements on signals. After that, different dictionary learning algorithms are compared and the most appropriate one is selected based on the demand of our application.

3.1. Human model representation

Given a dataset of human models denoted by $\mathcal{H}$, we define the representation of human models in $\mathcal{H}$ as:

- All models in $\mathcal{H}$ have the same mesh connectivity;
- $\mathcal{H}$ consists of m_D models and the i-th model is denoted by $\mathbf{h}_i$, that is $\mathcal{H} = \{\mathbf{h}_i\}_{i=1,\ldots,m_D}$;
- Each model has n_D vertices and the j-th vertex of $\mathbf{h}_i$ is denoted by $\mathbf{v}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \in \mathfrak{R}^3$.

When using the full body of every human model $\mathbf{h}_i$ in $\mathcal{H}$ to represent a single training signal, totally m_D signals with length $3n_D$ can be extracted from $\mathcal{H}$ for representation. In practice, the number of 3D human models that can be obtained is very limited (e.g., $m_D = 40/50$ in our experiments – see the training set of 40 female models as shown in Fig. 1). The signal length $3n_D$ is much larger than the number of signals m_D (i.e., $n_D = 33$k in our datasets), which does not satisfy the requirements of dictionary learning. More sophisticated signal extraction method needs to be developed.

3.2. Sparse representation

Given m training signals $\mathbf{y}_i$ stored in a matrix $\mathbf{Y}$ as

$$\mathbf{Y} = \{\mathbf{y}_i \,|\, \mathbf{y}_i \in \mathfrak{R}^n,\ i = 1, \ldots, m\} \in \mathfrak{R}^{n \times m},$$

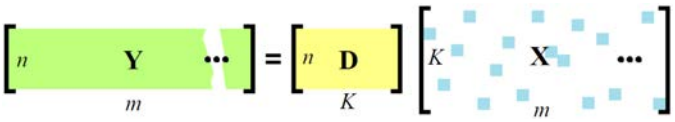


Figure 2: A schematic illustration of matrix sizes in dictionary learning for sparse representation.

each signal can be effectively represented by K coefficients in a vector $\mathbf{x}_i \in \Re^K$ with the help of a dictionary matrix $\mathbf{D} \in \Re^{n\times K}$ as a linear system

$$[\mathbf{Y}_{n\times m}] = [\mathbf{D}_{n\times K}][\mathbf{X}_{K\times m}], \tag{1}$$

where

$$\mathbf{X} = \{\mathbf{x}_i \,|\, \mathbf{x}_i \in \Re^K,\ i = 1, \ldots, m\} \in \Re^{K\times m}.$$

All m training signals in $\mathbf{Y}$, which means everything from the original dataset, can be re-assembled by $\mathbf{D}$ and $\mathbf{X}$ when $\mathbf{D}$ is an overcomplete dictionary (i.e., $K > n$). Finding a sparse representation is to determine a new overcomplete dictionary $\mathbf{D}$ with sparsity enhanced in $\mathbf{X}$. This can be achieved when the number of training signals is much larger than their lengths (i.e., $m \gg K > n$). As shown in Fig.2, simply forming each human model into a single signal vector is hard to meet the requirement on the number of training signals.

3.3. Learning algorithm

Because of the overcompleteness ($K > n$) required on the dictionary $\mathbf{D}$, $\mathbf{X}$ can have infinite number of solutions to approach $\mathbf{Y}$ as an underdetermined linear system. To help the system converging to a desired solution, constraints will be imposed in an optimization framework to determine the optimally sparse coefficient matrix with a desired accuracy. Given a dictionary $\mathbf{D}$, that is

$$\min_{\mathbf{X}} \|\mathbf{X}\|_0 \quad s.t.\ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2 \le \epsilon \tag{2}$$

where L_0-norm indicates the number of non-zero elements in the matrix $\mathbf{X}$ and ϵ is a threshold to control the accuracy of approximation error.

To effectively learn the dictionary matrix $\mathbf{D}$ with a controlled compression ratio (i.e., the number of non-zero elements in $\mathbf{X}$), an objective function is introduced in [5] as

$$\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 \quad s.t.\ \|\mathbf{x}_i\|_0 \le \lambda\ (i = 1, \ldots, m) \tag{3}$$

where λ is a user-specified constant for constraining the sparsity in the coefficient matrix. By this formulation, each column of $\mathbf{X}$ can only have λ non-zero elements at most. It implies that every training signal in $\mathbf{Y}$ can only be approximated by maximal λ atoms in $\mathbf{D}$. A batch-training algorithm named as K-SVD is introduced in [5] to determine $\mathbf{X}$ and $\mathbf{D}$ (see the flow-chat shown in the left of Fig.3).

Being an alternative to the batch process in K-SVD, an online dictionary learning is proposed in [6]. By online style, only a small subset of the training dataset (e.g., one training signal) is considered during every iteration of the learning process. Therefore, datasets with large size can be handled and the previous training results can be re-used when more training samples are fed into the learning algorithm. The objective function for online dictionary learning is

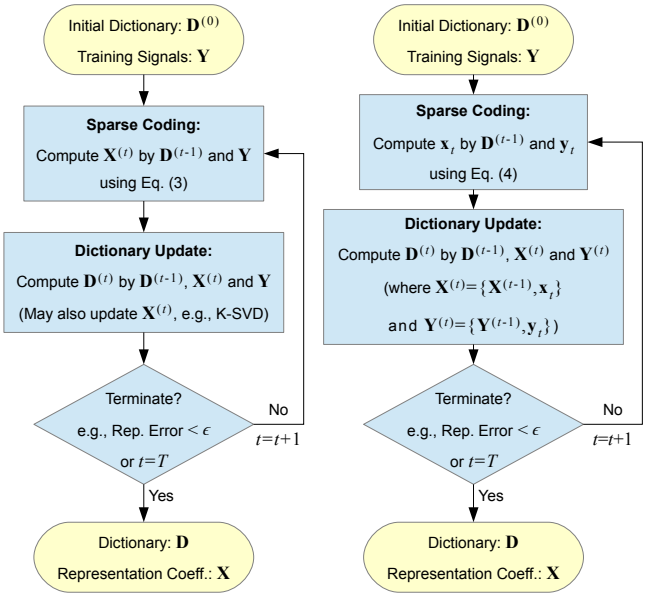


Figure 3: A schematic illustration of different dictionary learning styles – (left) batch learning [5] and (right) online learning [6].

$$\min_{\mathbf{D}\in C,\,\mathbf{X}} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 \quad s.t.\ \|\mathbf{x}_i\|_0 \le \lambda\ (i = 1, \ldots, m) \tag{4}$$

with $\mathbf{D} \in C$, where $\lambda/3$ is specified for x-, y- and z-components separately. C is a convex set to bound the normal of every atom in $\mathbf{D}$ as

$$C = \{\mathbf{d}_k \,|\, \mathbf{d}_k \in \Re^n,\ \|\mathbf{d}_k\|_2 \le 1,\ k = 1, \ldots, K\}.$$

The algorithm's flow chart is given in the right of Fig.3.

To keep the nice property of progressive enhancement, the online algorithm is employed in our implementation to determine the dictionary $\mathbf{D}$. Note that, given $\mathbf{y}_i$ and $\mathbf{D}$, the sparse coding coefficient $\mathbf{x}_i$ can be determined by an approach called *order recursive matchng pursuit* (ORMP) via Cholesky decomposition [42]. ORMP is also the method used in our approach to obtain the sparse representation of a new signal based on a given dictionary $\mathbf{D}$.

4. Signal Extraction and Synthesis

This section presents the details of signal generation from a limited number of 3D human models. After that, the method for synthesizing the signals back to the triangular mesh of a human body is presented.

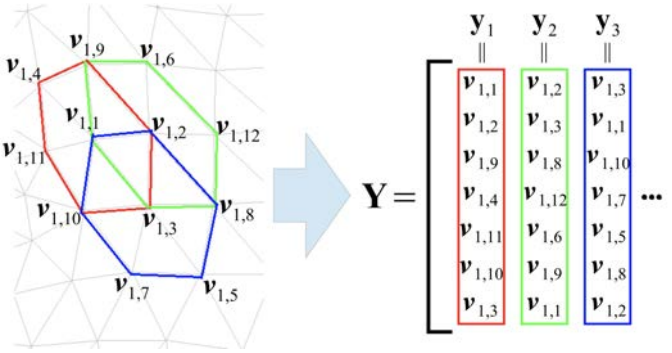


Figure 4: Converting a patch of vertices into a column of training signal.

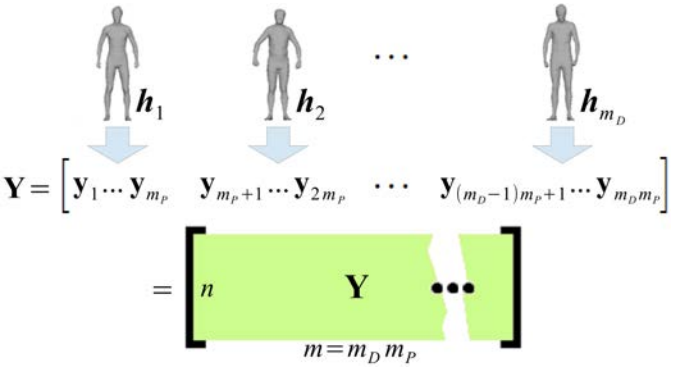


Figure 5: An illustration of extracting patch signals from m_D human models to form the training matrix $\mathbf{Y}$.

4.1. Patch signals

To benefit from the sparsity of dictionary learning, we need to organize a large number of training signals with rather short signal length – i.e., $n \ll m$ as shown in Fig.2. Being inspired by the image processing examples like [6], in which patches of pixels from photos are sampled as training signals, patch signals are proposed in our approach. To develop a signal sampling method on surface meshes which is analogous to the patch signals on images, the connectivity between mesh vertices on a model's mesh surface can help.

We explain the idea by using the example shown in Fig.4 by assuming the shown mesh is a portion of the model $\mathbf{h}_1$ from the dataset $\mathcal{H}$ with the vertices $\{\mathbf{v}_{1,j}\}$. The first twelve mesh vertices are indicated in the figure so that $j = 1, \ldots, 12$. For simplicity at this stage, let's further assume all vertices in this mesh have the valence of 6 (i.e., every vertex has six neighboring vertices). Starting from the first vertex $\mathbf{v}_{1,1}$, its one ring of neighbors bounded by the red hexagon will be collected to form a patch vertex by vertex. By unfolding this sequence of patch vertices as indicated by the red rectangle in the right of Fig.4, the first signal $\mathbf{y}_1$ in the training matrix $\mathbf{Y}$ can be formed by the vertices' 3D coordinates. When necessary, other patches can be constructed and converted to signals by the similar grouping process. For example, the green and blue patches centered at the vertices $\mathbf{v}_{1,2}$ and $\mathbf{v}_{1,3}$ respectively are converted to signals $\mathbf{y}_2$ and $\mathbf{y}_3$. The signal extraction from $\mathbf{h}_1$ is completed as long as the group of patches are constructed such that their union covers all vertices. That means the total number of patches m_p can be much smaller than the number of vertices on $\mathbf{h}_1$.

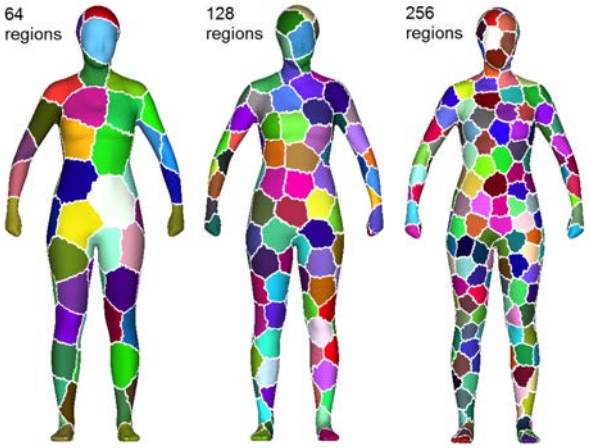


Figure 6: The segmentation results of CVT with different numbers of regions.

By making use of the consistent connectivity among all models in the dataset, the same set of patches constructed on $\mathbf{h}_1$ can be extracted from every other model $\mathbf{h}_{i,i\neq 1}$ in $\mathcal{H}$ with different shapes. As a result, a total number of $m = m_D m_P$ training signals can be extracted from the dataset. A schematic illustration can be found in Fig.5.

Each patch signal is scaled into a bounding box with unit size, and the corresponding ratio of scaling and the center of all vertices are also stored (i.e., 4 coefficients). In summary, for a human model segmented into m_P patches, the total number of coefficients as floating number to be stored are

$$(4 + \lambda)m + Kn$$

Moreover, we need to spend $((K + 1) + m\lambda)$ integers to store the sparse matrix $\mathbf{X}$ in the way of *compressed sparse row* (CSR). The memory consumption of these integers equals to $((K + 1) + m\lambda)/2$ floating point numbers. Note that, the x-, y- and z-coordinates are stored separately in three $\mathbf{Y}$, $\mathbf{D}$ and $\mathbf{X}$ matrices. Three sparse matrices of $\mathbf{X}$ can share the same CSR representation.

4.2. Segmentation and Patch Generation

Now we try to solve the problem of segmenting the triangular mesh of a human model $\mathbf{h}_i$ into m_P regions, and present the scheme to convert each region into a patch signal.

A segmentation algorithm that can divide the mesh model into regions with equal number of vertices is demanded for the purpose of generating less redundant information. For this purpose, we conduct the *centroidal Voronoi tessellation* (CVT) with L_2-norm distance metric [7]. Instead of using the triangular area in the computation, we assign each triangle's area as one with the motivation to obtain regions containing similar numbers of vertices. The results of segmentation can be found in Fig.6. The regions generated by CVT can only have nearly similar numbers of vertices. A scheme is needed to generate patches with a given number of vertices, n_P, to cover the whole model.

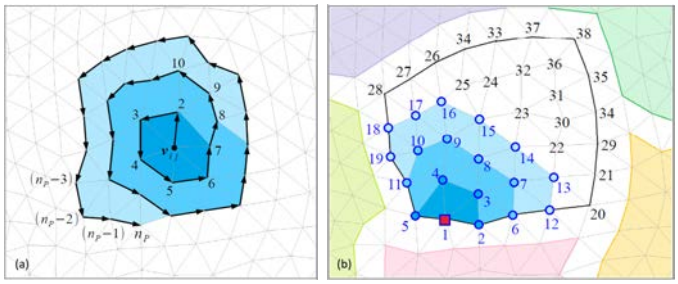


Figure 7: The illustration of forming a patch signal around a base vertex (a) by a spiral path in an anti-clockwise order and (b) inside a segmented region.

The following steps are used to construct a patch signal with n_P vertices. As shown in Fig.7, we select one vertex on the boundary of a region as a seed and then apply the following flooding based method to generate a new patch and the ordered vertices.

- *Rule I*: Start collecting the neighbors of the base vertex in an anti-clockwise order from the one with smallest index (e.g., '2' in Fig.7(a)).
- *Rule II*: Vertices are collected ring by ring in a spiral path until reaching the total number of n_P (see the illustration shown in Fig.7(a)).
- *Rule III*: Vertices are firstly collected in the same region as the base vertex (see Fig.7(b)).
- *Rule IV*: In case if the total number of vertices is still less than n_P after collecting all vertices in a region, the collection is allowed to conduct in neighboring regions.

The last rule is defined to solve the problem that the regions generated by the CVT-based segmentation have different numbers of vertices. For example, the result of 128 regions shown in Fig.6, the numbers of vertices in different regions are in the range between 200 and 342. We choose $n_P = 342$ and can then generate patch signals with the same number of vertices by applying Rule IV. Therefore, the signal matrix $\mathbf{Y}$ can be obtained for the dictionary learning of sparse representation.

4.3. Model synthesis

After the learning process of sparse representation is completed by L_0-norm minimization, a dictionary $\mathbf{D}$ filled by the computed prototype signal atoms and a coefficient matrix $\mathbf{X}$ will be output as a sparse representation of the original dataset $\mathcal{H}$. Conceptually, multiplying the derived dictionary $\mathbf{D}$ by the sparse coefficient matrix $\mathbf{X}$, all training signals can be synthesized by $\tilde{\mathbf{Y}} = \mathbf{DX} \simeq \mathbf{Y}$ to re-assemble their original signals from $\mathbf{Y}$ and therefore also the human models of the original dataset $\mathcal{H}$.

The detailed steps of synthesizing an approximate model $\tilde{\mathbf{h}}_i$ to reconstruct the i-th model $\mathbf{h}_i$ in the dataset are as follows.

1. As schematically shown in Fig.8, the m-th column in $\mathbf{X}$ is eligible to reconstruct the m-th training patch signal in $\tilde{\mathbf{Y}}$. From the $(m_P(i-1)+1)$-th column to the $m_P i$-th column of $\mathbf{X}$ are picked up for synthesizing the m patches of $\mathbf{h}_i$.

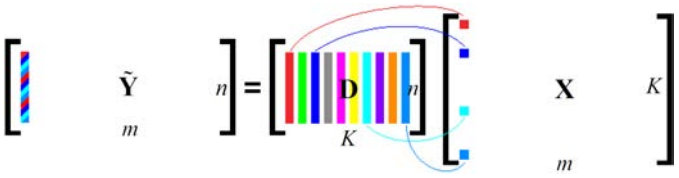


Figure 8: Signal synthesis by linear combination of dictionary atoms with respective sparse coefficients.

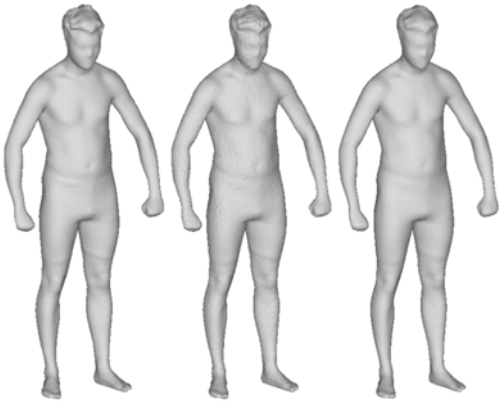


Figure 9: The resultant 3D shape of synthesis with high-frequency noises embedded, which can be removed by applying the Laplacian-based filter – (from left to right) the original model, the synthesized model and the smooth model obtained from Laplacian denoising.

2. The atoms in the dictionary will be multiplied with the corresponding columns of sparse coefficients in the way of linear combination to obtain the following column vectors.

$$[\tilde{\mathbf{y}}_{m_P(i-1)+1} \;\; \tilde{\mathbf{y}}_{m_P(i-1)+2} \;\; \cdots \;\; \tilde{\mathbf{y}}_{m_P i-1} \;\; \tilde{\mathbf{y}}_{m_P i}] \in \Re^{n \times m_p}$$

   with $\tilde{\mathbf{y}}_{m_P(i-1)+1} \simeq \mathbf{y}_{m_P(i-1)+1}$ and $n = n_p$.
3. Each patch of a human model can then be obtained by scaling back to its original size and transformed to its original position.
4. For the overlapped regions of patches, inconsistent positions could be determined by different patches for the same vertex. We apply the average position for such a vertex.

Our experiments have shown that very accurate results of 3D human models can be synthesized by applying the above steps although the synthesized 3D shape in general embedded some high-frequency noises (i.e., not as smooth as the original model – see the middle of Fig.9 for an example).

To have an idea on why the synthesized patches and thus the surfaces of models are not smooth, we study the dictionary atoms which compose the patch signals by linear combination. As shown in Fig.10, the patch signal of a region on a human model (Fig.10(a)) is formed by linear combination of the dictionary atoms (Fig.10(b)). Even by applying the mesh topology of the patch on these atoms, none of them can solely represent a patch with proper shape. It is because the dictionary learning is only constrained by optimization to minimize the error between the signals and their representation as formulated in Eq.(4), disregarding the continuity or connectivity between the patch vertices. Hence, although their linear combination can

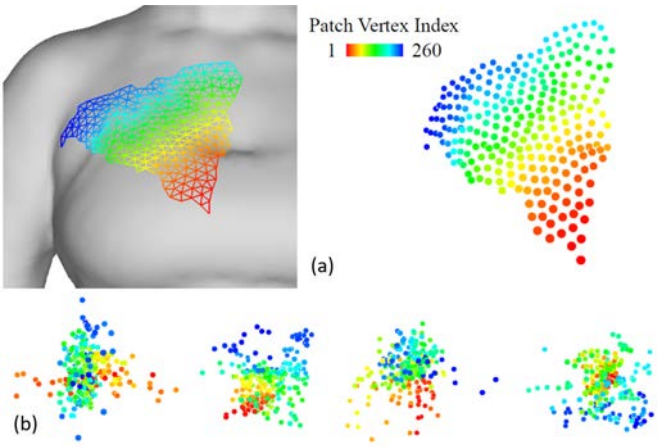


Figure 10: A patch signal (a) and some of its dictionary atoms (b). It can be found that the dictionary atoms are not smooth surfaces in general.

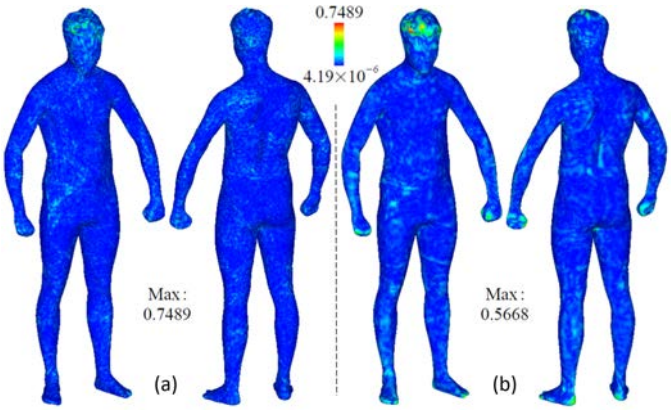


Figure 11: Geometric error analysis by vertex-to-vertex distances (unit: centimetre) of (a) the result of 3D shape synthesis (RMS=0.0428) and (b) the result after Laplacian denoising (RMS=0.0204) with RMS denoting the root mean square.

indeed reconstruct the corresponding signal, each of the dictionary atoms is not necessary to be interpreted as any meaningful geometric feature. This leads to non-smoothness on the synthesized patches and thus the reconstructed 3D human models.

As the errors all occur in the band of high-frequency, they can be easily removed by applying the Laplacian smoothing filter presented in [43]. The co-tangential weights are employed to consider the irregularity of triangular meshes. The result of denoising can be found in the right of Fig.9. The geometric error is given in Fig.11. It can be observed that both the average error and the maximal error on the denoising result are slightly reduced.

5. Statistical 3D Human Modeling

This section presents the method of statistical modeling for 3D human models. We first introduce the statistical method by using the conventional PCA-based representation. After that, the benefit from sparse representation will be studied to enable the usage of more PCs in the regression. Therefore, 3D models

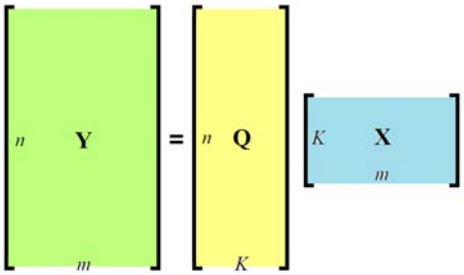
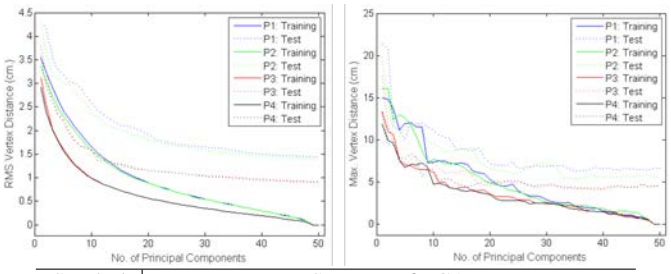


Figure 12: A schematic illustration of PCA-based representation of a human dataset in reduced size.



Symbol	Strategy of PCA	
P1	Original body height	Joint dimensions
P2	Normalized body height	Joint dimensions
P3	Original body height	Separate dimensions
P4	Normalized body height	Separate dimensions

Figure 13: A geometric error study of using different strategies in PCA-based statistical modeling of human models. The analysis has been conducted on both the training dataset and the test dataset.

in higher accuracy can be generated when consuming the same amount of memory.

5.1. PCA-based representation

Without loss of the generality, we can convert each model represented by a triangular mesh with n_D vertices into a column vector with dimension $3n_D$. The matrix $\mathbf{Y} \in \Re^{3n_D \times m_D}$ spans the whole dataset $\mathcal{H}$ can be reduced by applying the PCA to extract K most significant PCs with the largest absolute eigenvalues, where $K \ll m_D$. Storing all the K PCs in $\mathbf{Q} \in \Re^{3n_D \times K}$ and the corresponding coefficients in $\mathbf{X} \in \Re^{K \times m_D}$ leads to a representation of the original m_D models in a reduced size (see the schematic illustration in Fig.12).

As discussed in [24], when using normalized / original body height and joint / separate dimensions in PCA, results in different accuracy can be obtained. We conduct a study as shown in Fig.13, where the *root mean square* (RMS) and the maximal vertex-to-vertex distances are employed as the metrics for analysis. The strategy with normalized height and separate dimensions (P4) gives the most accurate synthesis and is employed in the rest of this paper.

5.2. Linear regression for parametric design

With the help of PCs, we can conduct a linear regression to map a set of semantic parameters to the coefficients of PCs [23].

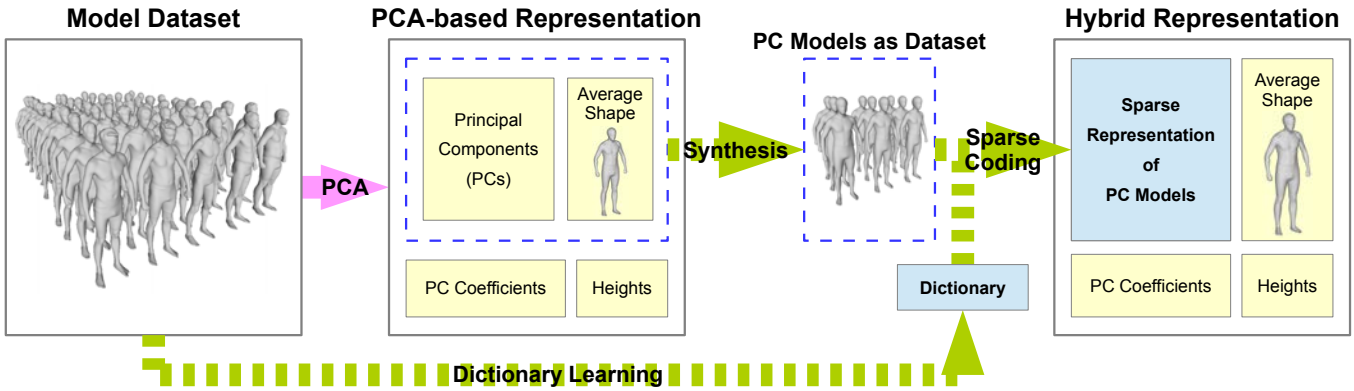


Figure 14: A flow-chart to illustrate the computational pipeline of hybrid representation. Starting from the original dataset $\mathcal{H}$ of human models, the PCA-based representation is first constructed. After that, every remained PC is converted into a 3D human model to be encoded by sparse coefficients w.r.t. the dictionary learned from $\mathcal{H}$. With the help of this hybrid representation, more PCs can be involved in the regression-based parametric design of 3D human models when having the same memory consumption.

We define the semantic parameters as p body measurements corresponding to every represented model in the dataset $\mathcal{H}$ as

$$\mathbf{L} = \{\mathbf{l}_i \,|\, \mathbf{l}_i \in \Re^{p\times 1},\ i = 1, \ldots, m_D\} \in \Re^{p\times m_D}. \tag{5}$$

The goal of linear regression is to determine the relationship between $\mathbf{X}$ and $\mathbf{L}$, such that

$$\mathbf{x}_i = \mathbf{R}\mathbf{l}_i + \mu, \quad i = 1, \ldots, m_D \tag{6}$$

where $\mathbf{R} \in \Re^{K\times p}$ is the regression matrix and $\mu \in \Re^{K\times 1}$ is the vector of intercept terms. When $m_D \gg ((K+1)\times p)$, the above linear system becomes overdetermined. We can then compute the least-square solution of $\mathbf{R}$ and μ.

When a new vector of semantic parameters $\tilde{\mathbf{l}}$ is given, we can conduct the following formula to determine the shape of a new human model.

$$\tilde{\mathbf{y}} = \mathbf{Q}\tilde{\mathbf{x}} = \mathbf{Q}(\mathbf{R}\tilde{\mathbf{l}} + \mu) \tag{7}$$

Note that, the shape information $\tilde{\mathbf{y}}$ here is based on the subtraction to the average shape of models in a dataset. The function of parametric design is realized for generating new human models.

5.3. Hybrid representation for parametric design

As will be shown in the result section below, the sparse representation proposed in our paper can achieve a high compression ratio – i.e., by introducing a lot of zeros. We are able to store more PCs in sparse representation than the original matrix $\mathbf{Q}$ when consuming the same amount of memory. With utilization of more PCs by a PCA based representation, its linear regression with semantic parameters is expected to be more accurate. We propose to conduct the sparse representation to store the PCs in semantic modeling of 3D human bodies – named as *hybrid representation*. Details are given below (see also a flow-chart in Fig.14).

After constructing the PCA-based representation (as $\mathbf{Q}$ and $\mathbf{X}$) of a human dataset $\mathbf{Y}$, each column vector $\mathbf{q}_k$ in $\mathbf{Q}$ ($\mathbf{q}_k \in \Re^{3n_D}$) can be considered as a 3D human model when applying the same mesh topology of models in the dataset $\mathcal{H}$. The shape of $\mathbf{q}_k$ is a variant of the average shape of all human models in $\mathcal{H}$. A dataset can be formed by K such models corresponding to the K PC vectors. We can project each PC model $\mathbf{q}_k$ onto the dictionary $\mathbf{D}$ to obtain its sparse representation. As a result, all PC vectors can be represented as an assembly of atoms in $\mathbf{D}$ with the number of non-zero coefficients much smaller than $3n_D$. Hence, the parametric design tool with hybrid representation can generate more accurate results because that more PCs can be employed when consuming the same size of memory.

The dictionary $\mathbf{D}$ learned from the original training dataset of human models $\mathcal{H}$ can still be used here as the shapes of features encoded in $\mathbf{D}$ are closely related to those PC models, which are indeed variants of the average shape of models in $\mathcal{H}$. When being applied to a remote-access based application (e.g., after the hybrid representation is downloaded to a mobile device), the PC models will be synthesized from the sparse representation and re-assembled into PCs for the PCA based representation at running time. Together with the regressed PCA coefficients, 3D human models can be reconstructed from semantic parameters. More results and study will be given in the result section below.

6. Experiment Results

This section presents the experimental tests conducted in our research to demonstrate the performance of our data-driven human modeling approach based on sparse representation. We will first study the effectiveness of compression provided by sparse representation, and then give the results of semantic regression by hybrid representation. Lastly, preliminary trial is conducted to use the dictionary learned from human models to reconstruct general 3D models.

6.1. Effective compression

Experiments have been conducted to compare the results of models compressed at the similar level of compression ratio. CVT segmentation with 128 and 256 regions will be employed

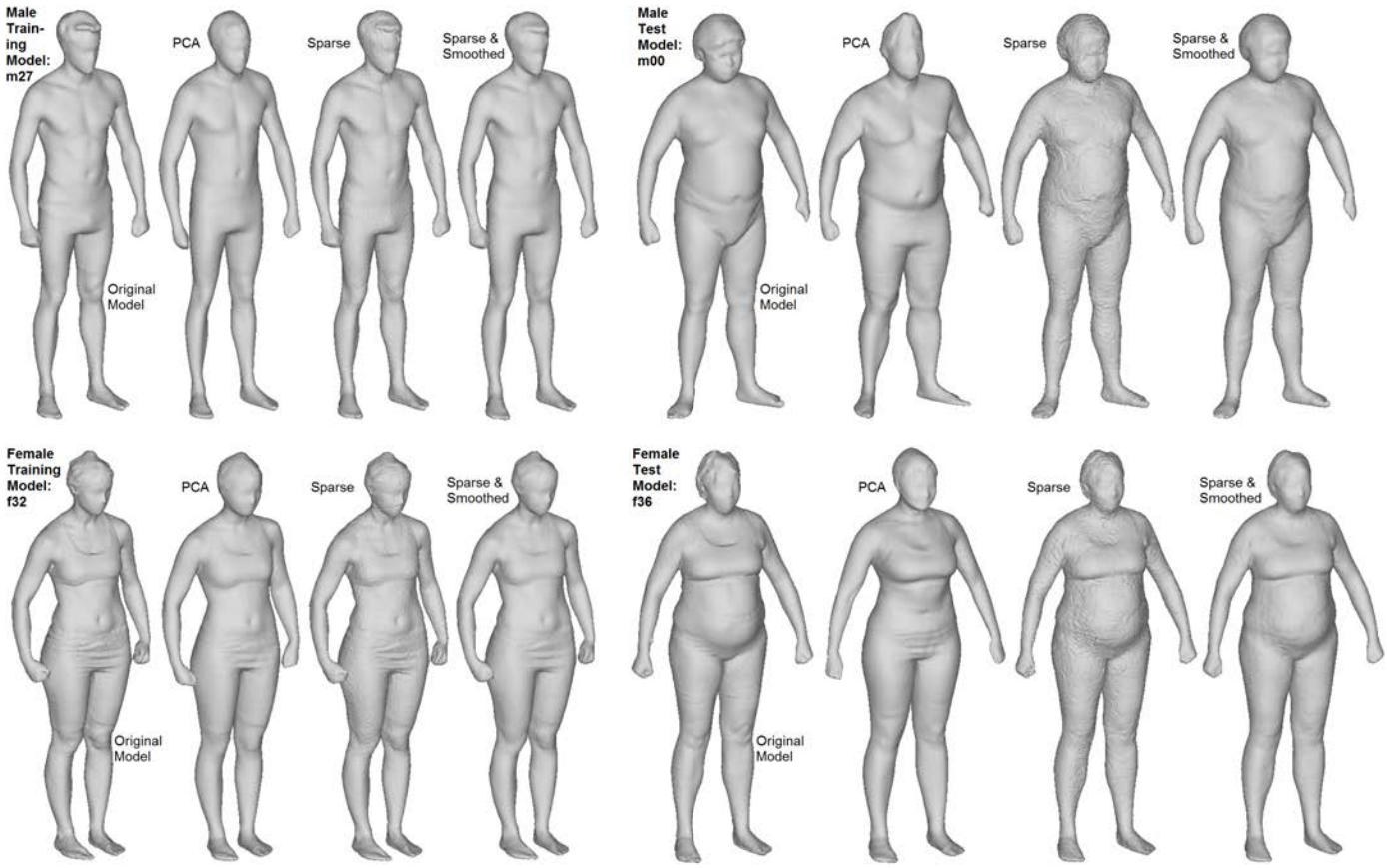


Figure 15: Results comparison of models synthesized from PCA-based representation vs. sparse representation (128 patches for each model) while using the same compression ratio – with around 74% and 70% data size reduced for male and female datasets respectively.

		# of	Reduced	Vert. Dist. Err. (cm)	
Male	Var. (%)	PCs	(%)	RMS	Max.
Training	94.85%	12 (50)	73.96	.86465	4.1872
Test				1.3528	5.3714
		# of	Reduced	Vert. Dist. Err. (cm)	
Female	Var. (%)	PCs	(%)	RMS	Max.
Training	94.89%	11 (40)	69.97	.79865	4.0975
Test				1.4387	9.3089

Table 1: Statistics of PCA-based representation with about 95% of the total variance remained.

		$n =$		Reduced	Vert. Dist. Err. (cm)	
Male	m_P	n_P	λ	(%)	RMS	Max.
Training	128	355	75	73.97	.076343	1.6629
Test					.099708	1.4271
Training	256	195	47	74.31	.094716	1.5838
Test					.11254	1.6081
		$n =$		Reduced	Vert. Dist. Err. (cm)	
Female	m_P	n_P	λ	(%)	RMS	Max.
Training	128	342	82	70.02	.067677	.76960
Test					.10144	1.0705
Training	256	198	52	70.34	.081345	1.0261
Test					.10508	1.1546

Table 2: Statistics of sparse representation when keeping the compression ratio at the same level.

to generate the sparse representation respectively. When comparing with the conventional compression method for statistical human modeling, we choose the results of PCA with 95% of the total variance remained – i.e., 12 out of 50 PCs for the male training dataset and 11 out of 40 PCs for the female training dataset when P4 strategy for PCA is employed. The geometric errors of PCA-based representation on both the training and the test sets are summarized in Table 1.

We now study the results of sparse representation while keeping the same level of reduced percentage – i.e., 74% for the male dataset and 70% for the female dataset. Selected models (from both the training and the test sets) are given in Figs.15 and 16 to visualize the shape difference comparing to the original model. Both the synthesized model and the smoothed model are shown. It can be clearly observed that the sparse representation preserves the shape of an original model better when using the same compression ratio as PCA-based representation. The statistical results on both the training dataset and the test dataset are given in Table 2, where the vertex-to-vertex distance errors are evaluated on the synthesized model. Errors on the models synthesized from sparse representation are only about $1/10$ (RMS) and $1/3$ (Max.) of the models obtained from PCA-based representation.

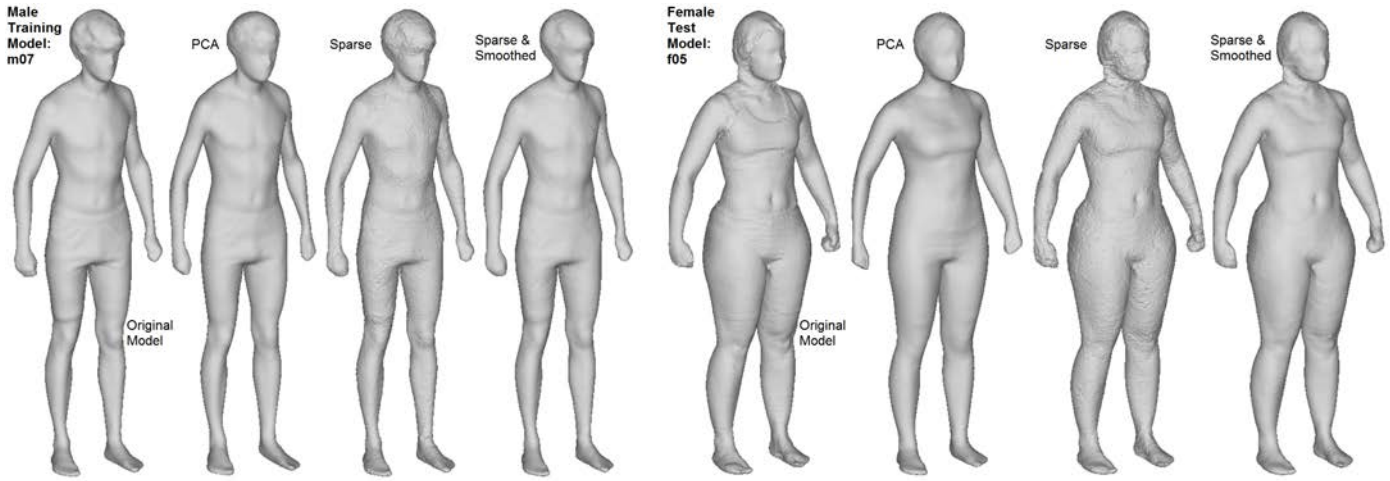


Figure 16: Results comparison of models synthesized from PCA-based representation vs. sparse representation (256 patches for each model) while using the same compression ratio – with around 74% and 70% data size reduced for male and female datasets respectively.

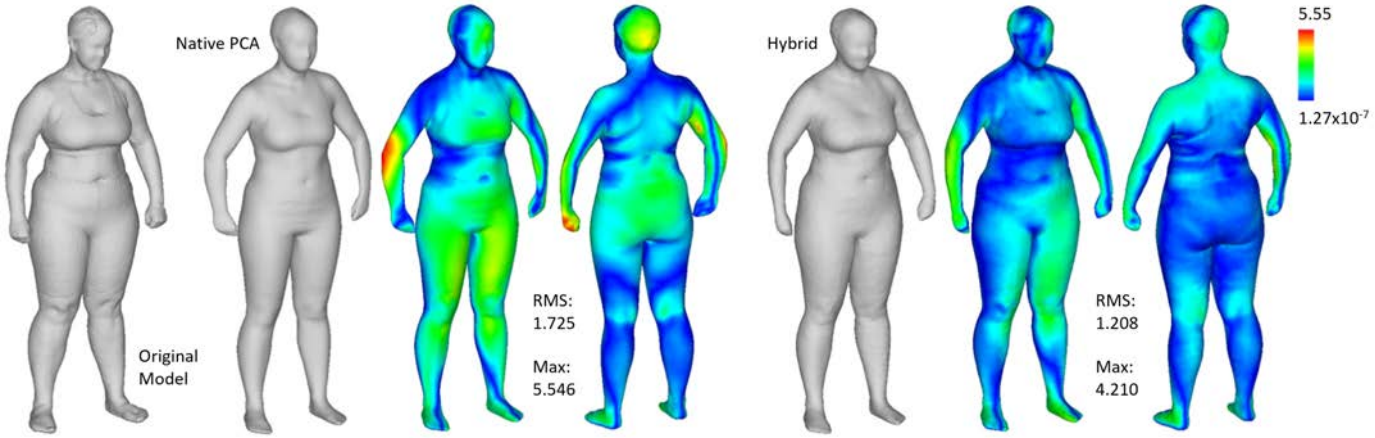


Figure 17: A female model reconstructed by linear regression with 8 parameters – from left to right are the original model, the model generated by native PCA-based representation with 5 PCs and the model generated by hybrid representation with 10 PCs having the same memory consumption. The color maps give the distribution of vertex-to-vertex distance errors (unit: centimetre).

6.2. Hybrid representation for regression

The capability of sparse representation in presenting 3D shapes also benefits the hybrid representation for regression. Table 3 shows the data size information of PCA and hybrid representations with the male training dataset as source. For the native PCA-based representations with $K = 5$ and $K = 10$ chosen PCs, the data sizes of their components can be calculated by the dimension of $\mathbf{Q}$. For the hybrid representation in Table 3, $m_P = 128$ and $n_P = 264$ are employed. Similarly, we also generate the comparison by using the female training dataset as source – see Table 4 for details. It can be found that the memory consumption of hybrid representation is only half of the native PCA-based representation. As a result, the number of PCs employed in the regression of statistical human modeling is doubled.

We now demonstrate the benefit of shape accuracy gained by using hybrid representation. As shown in Fig.17, the linear regression of female human body is generated by using 8 semantic parameters – including body height, neck girth, shoulder width, bust girth, under-bust girth, waist girth, hip girth and inside-leg length. The vertex-to-vertex distances between a generated model and the original model are conducted as the metric of geometric error. The RMS and maximal values on the results of native PCA-based and hybrid representations are given. In short, the hybrid representation can use more PCs in regression to generate more accurate results.

6.3. Reconstructing general freeform models

In this sub-section, we conduct more interesting tests to reconstruct general freeform models by the atoms of $\mathbf{D}$, which are learned from the original dataset of human models $\mathcal{H}$. Specifically, we re-assemble an armadillo model with $43,245$ vertices the mesh topology of which is totally different from the human models in $\mathcal{H}$. The dictionary trained from male models with $m_P = 256$ patches where each patch contains $n_P = 195$ vertices (see Table 2) is employed for the experimental tests here. To reconstruct the armadillo model by atoms stored in this dictionary, each patch should contain at least 195 vertices. Therefore,

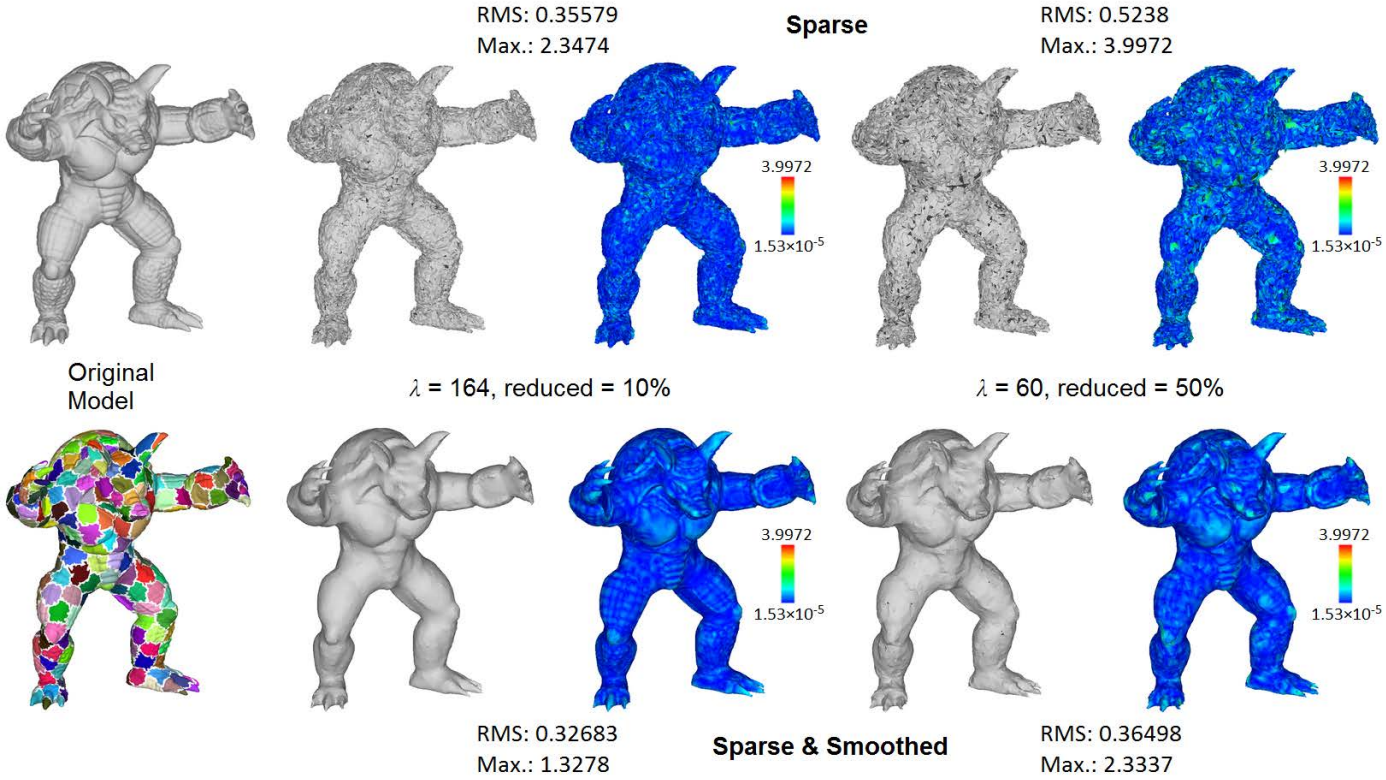


Figure 18: Experimental results of reconstructing an Armadillo model with the height 180cm by using the dictionary learned on the dataset of male human models each having 256 patches. The most left column shows the original model and the segmented patches for reconstruction. By using different λs, models with different reduced percentage – 10% and 50% here – can be reconstructed successfully. The distributions of vertex-to-vertex distances to the original model are shown by the color maps, and both RMS and the maximal values are reported for each result (unit: centimetre).

Rep.		**Native PCA**	**Hybrid**		
# of PCs		5	10		
Data Size	PCs	505,020	1,010,040	Dictionary	396,000
				Sparse Coeff.	83,311.5 ($\lambda = 43$)
				Means	3,840
				Norms	3,840
				Ver. Map	16,896
				Sub-Total	503,887.5
	Coeff.	750	1,500		
	Avg. S.	101,004			
	Height	50			
	Total	606,824	1,112,594	606,441.5	

Means: the centers of patch signals;
Norms: the original scales of normalized signals;
Ver. Map: mapping vertices on the original model to the patch signals;
Avg. S.: the average shape of all models in the dataset.

Table 3: Data size of native PCA-based and hybrid representations for male training dataset with $m_D = 50$ and $n_D = 33,668$.

Rep.		**Native PCA**	**Hybrid**		
# of PCs		5	10		
Data Size	PCs	498,750	997,500	Dictionary	390,000
				Sparse Coeff.	83,311.5 ($\lambda = 43$)
				Means	3,840
				Norms	3,840
				Ver. Map	16,640
				Sub-Total	497,631.5
	Coeff.	600	1,200		
	Avg. S.	99,750			
	Height	40			
	Total	599,140	1,098,490	598,621.5	

Table 4: Data size of native PCA-based and hybrid representations for female training dataset with $m_D = 40$ and $n_D = 33,250$.

we first apply the CVT based method to construct 333 patches where each patch contains 91 to 193 vertices (see the left bottom of Fig.18). Each patch is extended to have 195 vertices to use the atoms already learned in the dictionary. After that, as mentioned in Section 3.3, the ORMP method [42] is employed to determine the sparse coefficients for each patch on the armadillo model.

Experimental results of our tests are given in Fig.18, where tests with two different values of λ thus different reduced percentages are given. We also provide the results both before and after smoothing. The RMS and maximal values of the vertex-to-vertex distances between the original armadillo model and the reconstructed ones are also provided to evaluate the results quantitatively. The results are encouraging. Models with high accuracy can be re-assembled when using a large value of λ. Models with acceptable quality are still able to be obtained even

after increasing the reduced percentage to 50%. Although we mainly focus on 3D human modeling, the method proposed in this paper can be generalized to enable the sparse representation (therefore compression) of 3D freeform models.

7. Conclusion and Discussion

We present a method to generate sparse representation for data-driven 3D human modeling in this paper. All human models are represented as a collection of patch signals. L_0-learning algorithm is selected to train an overcomplete dictionary matrix for sparse representation. As a result, the patch signals can be presented with a controlled small number of coefficients together with the dictionary matrix. Experimental tests have been conducted on a variety of models to demonstrate the performance of our approach. In summary, the representation of 3D human models introduced in this paper has high compression ratio and low shape-approximation error. Moreover, we also present the benefit of using this sparse representation in the application of regression-based 3D human body modeling.

The major problem of current approach is the non-smoothness of synthesized patch signals as also discussed in Section 4.3, which is caused by the non-smooth atoms extracted from dictionary learning (see Fig.10). It is worthy to incorporate the geometric smoothness into the consideration. The spectrum coefficients (such as Laplacian coordinates [44] or manifold harmonics [45]) instead of vertex positions are planned to be considered in our future research. With the help of smooth patch signals, we could be able to directly compute the regression on sparse representation instead of the hybrid representation.

Acknowledgement

This work is partially supported by HKSAR Innovation and Technology Commission (ITC) Innovation and Technology Fund (Project Ref. No.: ITT/032/18GP), and the Natural Science Foundation of China (61976040, 61702079). The authors also would like to thank Xiaoting Zhang for providing the program of CVT segmentation and the valuable comment given by Zishun Liu in private communications.

References

[1] B. Allen, B. Curless, Z. Popović, The space of human body shapes: reconstruction and parameterization from range scans, ACM Transactions on Graphics (TOG) 22 (3) (2003) 587–594. doi:10.1145/1201775.882311.

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, J. Davis, SCAPE: Shape completion and animation of people, ACM Transactions on Graphics (TOG) 24 (3) (2005) 408–416. doi:10.1145/1186822.1073207.

[3] C. C. Wang, Parameterization and parametric design of mannequins, Computer-Aided Design 37 (1) (2005) 83–98. doi:10.1016/j.cad.2004.05.001.

[4] G. Pons-Moll, J. Romero, N. Mahmood, M. J. Black, Dyna: A model of dynamic human shape in motion, ACM Transactions on Graphics 34 (4). doi:10.1145/2766993.

[5] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322. doi:10.1109/TSP.2006.881199.

[6] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 689–696. doi:10.1145/1553374.1553463.

[7] B. Lévy, Y. Liu, Lp centroidal voronoi tessellation and its applications, ACM Transactions on Graphics (TOG) 29 (4) (2010) 119:1–119:11. doi:10.1145/1778765.1778856.

[8] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Transactions on Image Processing 17 (1) (2007) 53–69. doi:10.1109/TIP.2007.911828.

[9] B. Yang, S. Li, Multifocus image fusion and restoration with sparse representation, IEEE Transactions on Instrumentation and Measurement 59 (4) (2009) 884–892. doi:10.1109/TIM.2009.2026612.

[10] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2008) 210–227. doi:10.1109/TPAMI.2008.79.

[11] Y. Xu, D. Zhang, J. Yang, J.-Y. Yang, A two-phase test sample sparse representation method for use with face recognition, IEEE Transactions on Circuits and Systems for Video Technology 21 (9) (2011) 1255–1262. doi:10.1109/TCSVT.2011.2138790.

[12] Q. Wang, F. Chen, W. Xu, M.-H. Yang, Online discriminative object tracking with local sparse representation, in: 2012 IEEE Workshop on the Applications of Computer Vision (WACV), 2012, pp. 425–432. doi:10.1109/WACV.2012.6162999.

[13] Z. He, S. Yi, Y.-M. Cheung, X. You, Y. Y. Tang, Robust object tracking via key patch sparse representation, IEEE Transactions on Cybernetics 47 (2) (2016) 354–364. doi:10.1109/TCYB.2016.2514714.

[14] Y. Cong, J. Yuan, J. Liu, Abnormal event detection in crowded scenes using sparse representation, Pattern Recognition 46 (7) (2013) 1851–1864. doi:10.1016/j.patcog.2012.11.021.

[15] Q. Chen, G. Zhang, X. Yang, S. Li, Y. Li, H. H. Wang, Single image shadow detection and removal based on feature fusion and multiple dictionary learning, Multimedia Tools and Applications 77 (14) (2018) 18601–18624. doi:10.1007/s11042-017-5299-0.

[16] R. Gal, D. Cohen-Or, Salient geometric features for partial shape matching and similarity, ACM Transactions on Graphics (TOG) 25 (1) (2006) 130–150. doi:10.1145/1122501.1122507.

[17] G. Lavoué, Combination of bag-of-words descriptors for robust partial shape retrieval, The Visual Computer 28 (9) (2012) 931–942. doi:10.1007/s00371-012-0724-x.

[18] Z. Lian, A. Godil, X. Sun, J. Xiao, CM-BOF: Visual similarity-based 3d shape retrieval using clock matching and bag-of-features, Machine Vision and Applications 24 (8) (2013) 1685–1704. doi:10.1007/s00138-013-0501-5.

[19] R. Litman, A. Bronstein, M. Bronstein, U. Castellani, Supervised learning of bag-of-features shape descriptors using sparse coding, Computer Graphics Forum 33 (5) (2014) 127–136. doi:10.1111/cgf.12438.

[20] S. Xiong, J. Zhang, J. Zheng, J. Cai, L. Liu, Robust surface reconstruction via dictionary learning, ACM Transactions on Graphics (TOG) 33 (6) (2014) 201:1–201:12. doi:10.1145/2661229.2661263.

[21] M. Wei, J. Huang, X. Xie, L. Liu, J. Wang, J. Qin, Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery, IEEE Transactions on Visualization and Computer Graphics 25 (10) (2018) 2910–2926. doi:10.1109/TVCG.2018.2865363.

[22] L. Xu, R. Wang, J. Zhang, Z. Yang, J. Deng, F. Chen, L. Liu, Survey on sparsity in geometric modeling and processing, Graphical Models 82 (2015) 160–180. doi:10.1016/j.gmod.2015.06.012.

[23] C.-H. Chu, Y.-T. Tsai, C. C. Wang, T.-H. Kwok, Exemplar-based statistical model for semantic parametric design of human body, Computers in Industry 61 (6) (2010) 541–549. doi:10.1016/j.compind.2010.03.004.

[24] T.-H. Kwok, K.-Y. Yeung, C. C. Wang, Volumetric template fitting for human body reconstruction from incomplete data, Journal of Manufacturing Systems 33 (4) (2014) 678–689. doi:10.1016/j.jmsy.2014.05.009.

[25] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, M. J. Black, SMPL: A skinned multi-person linear model, ACM Transactions on Graphics

(TOG) 34 (6) (2015) 248:1–248:16. doi:10.1145/2816795.2818013.

[26] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, B. Schiele, Building statistical shape spaces for 3d human modeling, Pattern Recognition 67 (2017) 276–286. doi:10.1016/j.patcog.2017.02.018.

[27] Y. Chen, Z. Liu, Z. Zhang, Tensor-based human body modeling, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 105–112. doi:10.1109/CVPR.2013.21.

[28] 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: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006, pp. 147–156.

[29] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, H.-P. Seidel, A statistical model of human pose and body shape, Computer Graphics Forum 28 (2) (2009) 337–346. doi:10.1111/j.1467-8659.2009.01373.x.

[30] N. Hasler, T. Thormählen, B. Rosenhahn, H.-P. Seidel, Learning skeletons for shape and pose, in: Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, 2010, pp. 23–30. doi:10.1145/1730804.1730809.

[31] Y. Yang, Y. Yu, Y. Zhou, S. Du, J. Davis, R. Yang, Semantic parametric reshaping of human body models, in: Proceedings of the 2014 Second International Conference on 3D Vision, Vol. 2, 2014, pp. 41–48. doi:10.1109/3DV.2014.47.

[32] Y. Zhang, J. Zheng, N. Magnenat-Thalmann, Example-guided anthropometric human body modeling, The Visual Computer: International Journal of Computer Graphics 31 (12) (2015) 1615–1631. doi:10.1007/s00371-014-1043-1.

[33] B.-Y. Koo, E.-J. Park, D.-K. Choi, J. J. Kim, M.-H. Choi, Example-based statistical framework for parametric modeling of human body shapes, Computers in Industry 73 (2015) 23–38. doi:10.1016/j.compind.2015.07.007.

[34] Y. Shi, J. Ondřej, H. Wang, C. O'Sullivan, Shape up! perception based body shape variation for data-driven crowds, in: 2017 IEEE Virtual Humans and Crowds for Immersive Environments (VHCIE), 2017, pp. 1–7. doi:10.1109/VHCIE.2017.7935623.

[35] J. Huang, T.-H. Kwok, C. Zhou, Parametric design for human body modeling by wireframe-assisted deep learning, Computer-Aided Design 108 (2018) 19–29. doi:10.1016/j.cad.2018.10.004.

[36] H. Xu, J. Li, G. Lu, H. Deng, D. Zhang, J. Ye, Modeling 3D human body with a smart vest, Computers & Graphics 75 (2018) 44–58. doi:10.1016/j.cag.2018.07.005.

[37] S.-Y. Baek, K. Lee, Parametric human body shape modeling framework for human-centered product design, Computer-Aided Design 44 (1) (2012) 56–67. doi:https://doi.org/10.1016/j.cad.2010.12.006.

[38] B. Liu, X. Liu, Z. Yang, C. C. L. Wang, Concise and effective network for 3d human modeling from orthogonal silhouettes (2019). arXiv:1912.11616.

[39] X. Wang, X. Qian, A statistical atlas based approach to automated subject-specific FE modeling, Computer-Aided Design 70 (2016) 67–77. doi:https://doi.org/10.1016/j.cad.2015.07.003.

[40] K. Zhang, L. Cao, A. Fanta, M. P. Reed, M. Neal, J.-T. Wang, C.-H. Lin, J. Hu, An automated method to morph finite element whole-body human models with a wide range of stature and body shape for both men and women, Journal of Biomechanics 60 (2017) 253–260. doi:https://doi.org/10.1016/j.jbiomech.2017.06.015.

[41] G. Xu, T.-H. Kwok, C. C. Wang, Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization, Computer-Aided Design 91 (2017) 1–13. doi:https://doi.org/10.1016/j.cad.2017.04.002.

[42] S. F. Cotter, R. Adler, R. D. Rao, K. Kreutz-Delgado, Forward sequential algorithms for best basis selection, IEEE Proceedings of Vision Image and Signal Processing 146 (5) (1999) 235–244. doi:10.1049/ip-vis:19990445.

[43] M. Desbrun, M. Meyer, P. Schröder, A. H. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999, pp. 317–324. doi:10.1145/311535.311576.

[44] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, H.-P. Seidel, Laplacian surface editing, in: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 2004, pp. 175–184. doi:10.1145/1057432.1057456.

[45] B. Vallet, B. Lévy, Spectral geometry processing with manifold harmonics, Computer Graphics Forum 27 (2) (2008) 251–260. doi:10.1111/j.1467-8659.2008.01122.x.