

COMPM080 - Report  
**PCA on Human shape models**

Fen Jin

May 2019

## 1 Introduction

This project has been done in collaboration with Elliot Thompson, both of us have worked on the implementation equally and in parallel.

The topic of this project is to propose a reconstruction method based on Principal Component Analysis (PCA). Our main inspiration has been the paper "The space of human body shapes: reconstruction and parameterization from range scans" [2]. We wanted to create new realistic models from already existing 3D models, catching the main features of the human shape. Modifying the magnitude of the principal components corresponding to the directions of highest variance allow us to change the 3D mesh along the directions of the main features. This operation can result in meshes that are not present in the dataset, but in models that resemble the original ones.

The data we used belongs to a pre-processed dataset [1] of the original CAESAR<sup>1</sup> (Civilian American and European Surface Anthropometry Resource). Each model has 12500 vertices, and they are in correspondence between models.

The UI that the user is going to interact with is shown in Fig.1. It shows also the default settings. The general procedure is to follow the modes in order:

1. Mode 1: loads the first n number of models (Models to load slider below) from the data folder and calculates the eigenvectors.
2. Mode 2: shows the mean shape.
3. Mode 3: calculates the transformation of the sampled points and shows them as point cloud.
4. Mode 4: applies upscaling part and shows reconstructed surface.
5. Mode 5: chooses a target model (selectable from Target model) to apply PCA reconstruction to.
6. Mode 7: optional test of PCA of a single model (loading and finding the eigenvectors).
7. Mode 8: optional test of PCA of a single model (reconstruction).

## 2 Paper summary

The goal of the paper "The space of human body shapes: reconstruction and parameterization from range scans" is to fit a template mesh to human body scans such that it results in a completely registered model, which can be used in applications such as PCA.

---

<sup>1</sup><http://store.sae.org/caesar/>

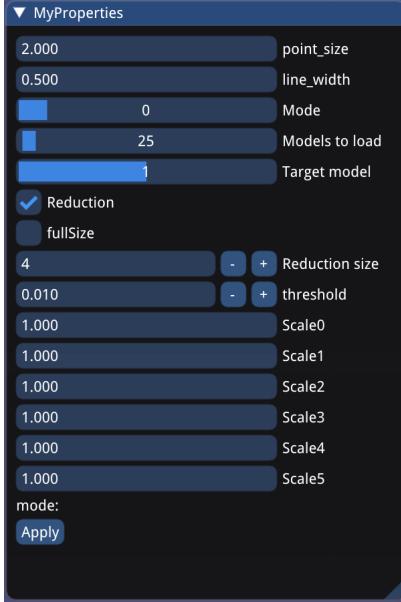


Figure 1: Main user interface and changeable settings.

The dataset used is the original CAESAR dataset. This one contains high resolution scans of a good range of human body shapes, varying in most of the demographic characteristics.

The main difficulty the paper tried to face is the hole-filling problem. Because of the difficulties in the scanning process that does not allow to have a fully complete model, some body parts are missing; for this reason the paper aims to take a complete template model and move each vertex towards the scan model. The expected result is that the template resembles the scan and at the same time fills the holes in the most natural way.

The objective function they modeled is composed by three types of error that are to be minimized: data error, smoothness error, marker error. The data error aims to bring the template model as close as possible to the scan model, by minimizing the distances between vertices. The objective function term is defined as:

$$E_d = \sum_{i=1}^n w_i \text{dist}^2(T_i v_i, D)$$

where  $n$  is the total number of vertices,  $w_i$  is a weighting factor depending on the region of the vertex on the body,  $T_i$  is affine transformation applied to the corresponding vertex  $v_i$ ,  $D$  is the scan model and the function  $\text{dist}()$  computes the sum squared distances (SSD) to the closest compatible vertex.

The smoothness error aims to constrain the deformations that the data error term can find, by forcing neighboring vertices to have similar transformation. This will result in smoother surfaces of the template model. The objective function term is defined as:

$$E_s = \sum_{\{i,j | \{v_i, v_j\} \in \text{edges}(Template)\}} \|T_i - T_j\|_F^2$$

where  $\|\cdot\|_F$  is the Frobenius norm.

The marker error aims to avoid convergence in local minima when the template and scan models are initialized not close enough. To achieve this, pairs of corresponding vertices have been identified and considered as markers which distance shall be minimized. The objective function term is defined as:

$$E_m = \sum_{i=1}^M \|T_{k_i}v_{k_i} - m_i\|^2$$

where  $m$  is the marker on the scan and the subscript  $k_i$  indicates the corresponding marker on the template.

The final objective function is a weighted combination of the three terms:

$$E = \alpha E_d + \beta E_s + \gamma E_m$$

where  $\alpha, \beta, \gamma$  are scalars optimized for a multi-resolution fitting. Results are shown in Fig.2.

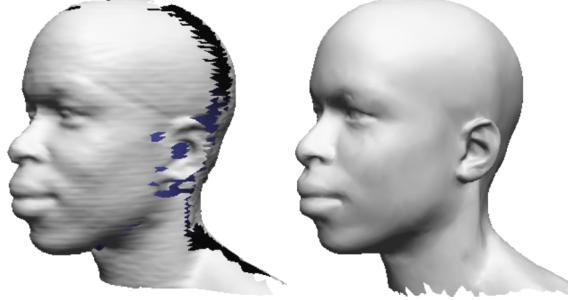


Figure 2: Adapted example of hole filling from [2]. On the left, the scans miss some body parts leaving a hole; on the right the result of the optimization proposed in the paper.

The main application of this paper, and the main focus of our implementation, is the Principal Component Analysis applied on the models. This will be addressed in the next section along with our implementation.

The paper suggests also other applications such as transfer of textures and morphing, markerless and marker-only matching and instrumentation transfer. Morphing can be achieved by linearly combine vertices from two models; markerless matching is obtained from a PCA-reconstructed surface while marker-only matching approximates the entire surface with just few points. Then instrumentation transfer refers to the ability to animate a model once a skeleton has been applied on it and skinning information are recovered.

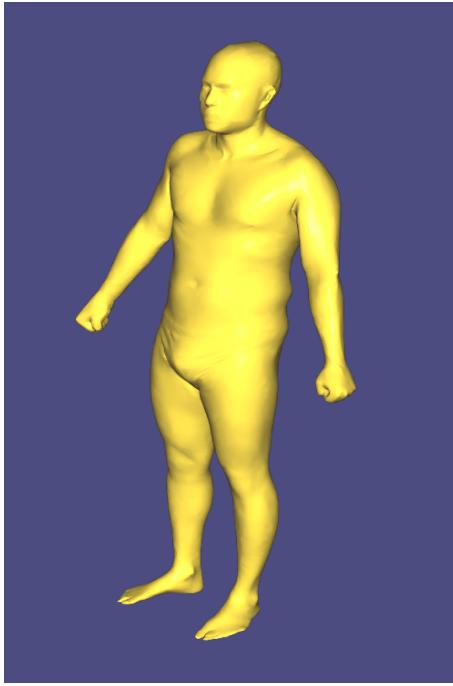
### 3 Implementation

The implementation follows generally the Principal Component Analysis section of the paper.

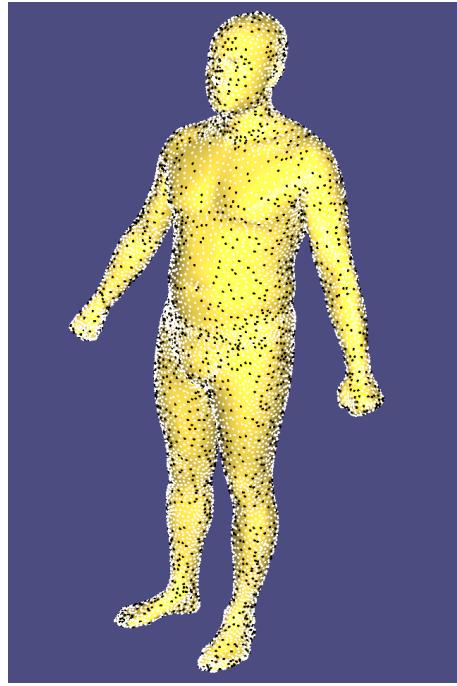
Once we have selected the  $k$  number of meshes to use, each one of them is loaded and stacked in vectors of size  $3n \times 1$ , where  $n$  is the number of vertices of the model. Since the vertices have already correspondence between models, is safe to stack them in a single vector and keep the correspondence. For performance reasons we decided to take a subset of vertices instead of the whole (so from now on  $n$  will indicate the subsampled number of vertices), but a version of the full size models is stored as well for later. These vertices then form the  $S$  matrix of size  $3n \times k$ . We compute the row-wise mean of  $S$  and subtract it from each column of  $S$ , this will result in the matrix  $U$ . This mean is denoted as mean shape ( $s\_mean$ ). Fig.3a shows the mean shape resulting from the first 25 models (not subsampled in this case). For all the examples in this report we have used  $k=25$  models.

From the matrix  $U$  we apply Singular Value Decomposition (SVD), to retrieve the eigenvectors that form the matrix  $C$ .

$$U = C\Sigma V^T$$



(a) Full mean shape.



(b) Black points are the sampled ones, white points compose the full mean shape.

Figure 3: Mean shapes from  $k=25$  models. On the right the full mean, on the left sampled (black) points compared to the full size (white) points.

Each column from the  $3n \times 3n$  matrix  $C$ , is one of the eigenvectors, ordered from the largest singular value to the smallest.

We have preferred this method instead of computing the covariance matrix for performance reasons. We tried to compute the covariance matrix:

$$Cov = \frac{1}{n-1} UU^T$$

and then apply eigendecomposition on it to retrieve the eigenvectors.

In reconstruction phase, we perform a weighted sum of the eigenvectors. First, we project the mean shape onto each eigenvector,

$$a_i = \langle s\_mean^T, c_i \rangle$$

where  $a_i$  is scalar and  $c_i$  is a column vector from  $C$ . Then, from interface we scale arbitrarily the first eigenvectors. The resulting output mesh  $O$  is:

$$O = \sum_{i=1}^{3n} \alpha_i a_i c_i$$

where  $\alpha_i$  corresponds to the scaling factor the user chooses, for the rest of them its values is equal to 1.

## 4 Extension

Because we have downsampled some points and applied PCA only on those to speed up the process, we need to upsample to the original number of vertices.

The method we thought to use is barycentric coordinates. The basic idea is to maintain the same relation the vertices have between the full size mean shape and the sampled version of it, even after the new configuration given by the PCA-reconstruction. We now denote full\_mean and sampled\_mean as the full size mean shape and the sampled version of it respectively. For each discarded vertex during the sampling process (there are in total full\_mean - sampled\_mean of these vertices), we want to find the closest 4 vertices ( $p_i$ ) in sampled\_mean using Approximate Nearest Neighbors technique. These 4 vertices will uniquely identify the position of the discarded vertex in space, so we find the corresponding barycentric coordinates by solving the following system:

$$Ax = b$$

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ 1 \end{pmatrix}$$

the coefficients  $\alpha_i, \beta_i, \gamma_i$  are the vertex coordinates of the 4 closest points, while  $q_i$  are the coordinates of the discarded point. Note that the ones in the forth rows of A and b ensures that the barycentric coordinates sum up to 1.

Once we have found the unknowns  $x_i$  and we have applied PCA reconstruction with the modifying scaling on the sampled points, we need to recreate the new point using the same 4 points used to find  $x_i$ , in their new positions. The new vertex new\_q position will be:

$$\text{new\_q} = x_1 p_1 + x_2 p_2 + x_3 p_3 + x_4 p_4$$

Fig.4 shows some results of the upscaling procedure. Although, we realized that many points were really far off the mean shape, creating spiky surfaces. We believe that it's due to the fact that sometimes the 4 closest points are positioned such that they can't represent  $q_i$  very well; this happens especially in poorly dense body parts of the mean shape.

For this reason we added a threshold to the distance of new\_q from the closest of the 4 points. If the new point were too far from it, we would just assign a new value corresponding to the centroid of the 4 points. This would make the surfaces smoother but has the downside of losing many details of the human body shapes. Fig.5 shows the effect of changing the threshold value on some models.

Note that the less samples we have (lower sample rate) the worse the reconstruction is. This is probably because the 4 closest points are far or can't really get good barycentric coordinates of the new points. Fig. 6 shows these effects on upscaling.

## 5 Evaluation

The design of the evaluation methods is divided into two parts: first, we evaluate the models qualitatively by describing the realism of the reconstructed model; second, we compare the meshes by computing the mean distance of the vertices from the mean shape. To achieve this, we compute both the full size mean shape ( $n \times 3$ ) and the upscaled model from PCA-reconstruction after using our barycentric coordinates technique, in their new configurations. We subtract one from the other and compute the norm for each row, i.e. vertex. Then, the sum will give the overall distance between the two models. Table 1 shows some SSD values changing the scaling factors. We can notice that as we diverge from scaling factors equal to 1 (when there is no changing in feature), the SSD increases, as the error is accumulating.

Note that here the full size mesh is considered as ground truth because it is computed without downsampling so it required a much greater amount of time, and it is used for this evaluation only.

scaling	s0	s1	s2	s3	s4	s5
-0.50	2717.62	89.91	1847.92	1651.92	905.62	453.97
-0.25	2264.71	86.08	1541.14	1378.14	756.73	383.12
0	1812.01	82.43	1234.76	1104.62	608.60	313.15
+0.25	1359.79	78.93	928.81	831.58	461.52	244.69
+0.50	908.43	75.63	624.09	560.21	316.62	179.02
+0.75	460.37	72.49	324.35	293.84	178.71	118.48
+1.00	69.61	69.61	69.61	69.61	69.61	69.61
+1.25	468.99	72.79	322.53	290.33	178.19	116.59
+1.50	917.093	76.18	622.24	556.24	316.37	177.03
+1.75	1368.54	79.73	926.74	827.64	461.28	242.78
+2.00	1820.89	83.38	1232.51	1100.43	608.56	311.31

Table 1: SSD values altering the scaling factors s corresponding to the respective eigenvectors.

As result of the PCA it is noticeable that the first eigenvectors clearly control mainly some specific feature: the first one corresponds to height, the second one to forward-backward pose, the third one to weight. Fig.7 shows some examples. Adding the number of models to load probably alters this effect as the variances in feature space might be different.

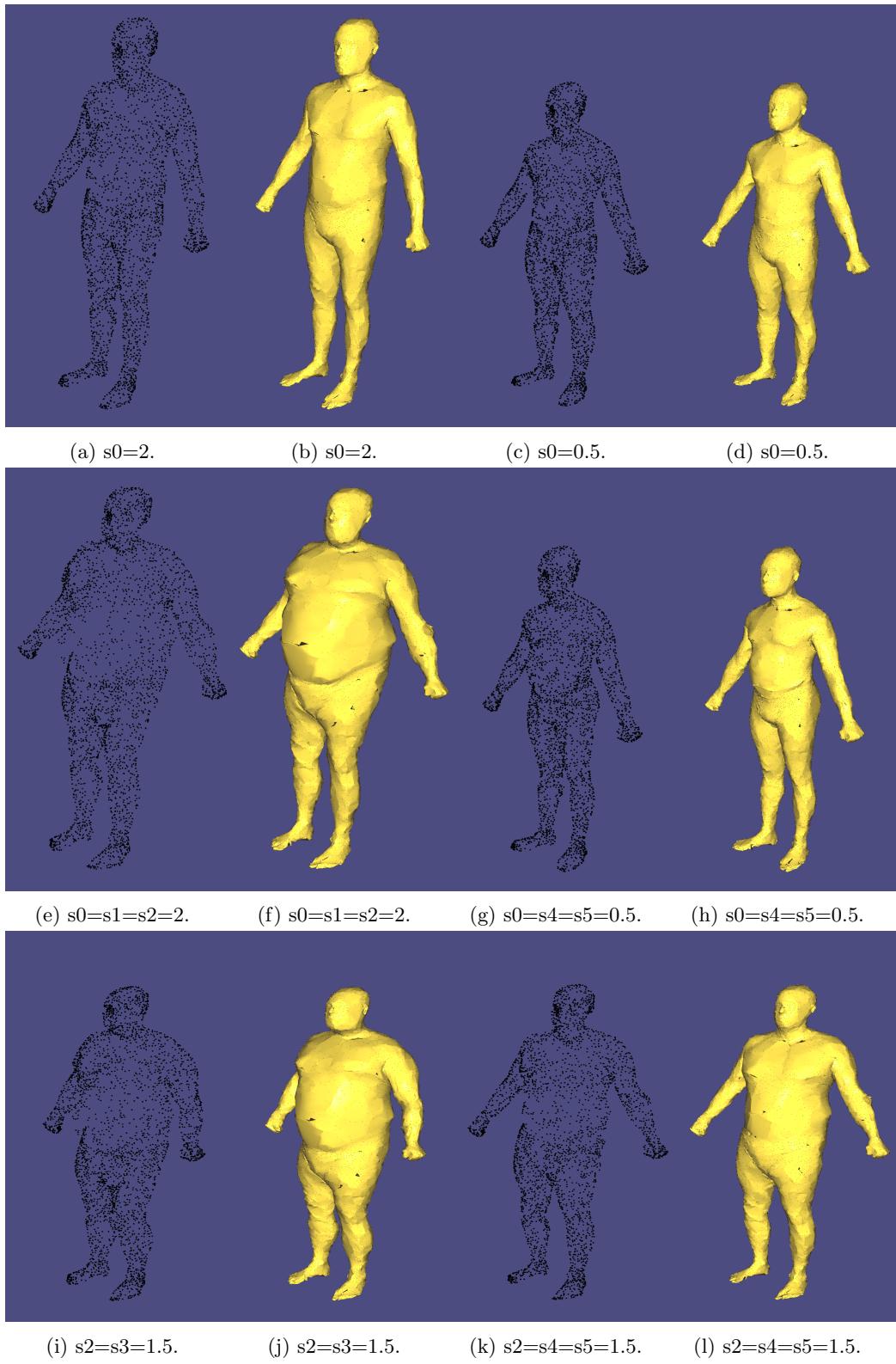


Figure 4: Sampled points and reconstructed model from upscaling. The scaling factors are indicated by  $s$ , where the number followed by  $s$  indicates which eigenvector is it corresponding to. They are equal to 1 except the one indicated under the images.

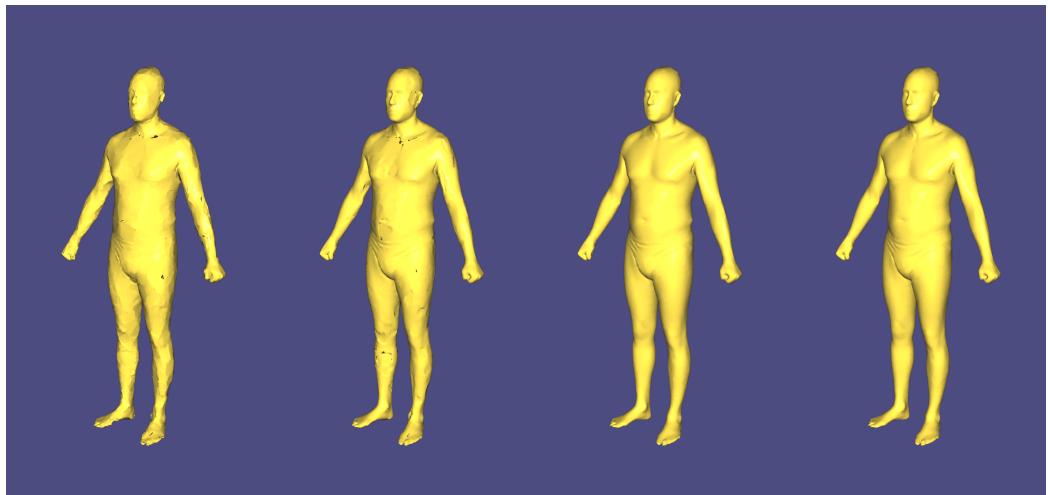
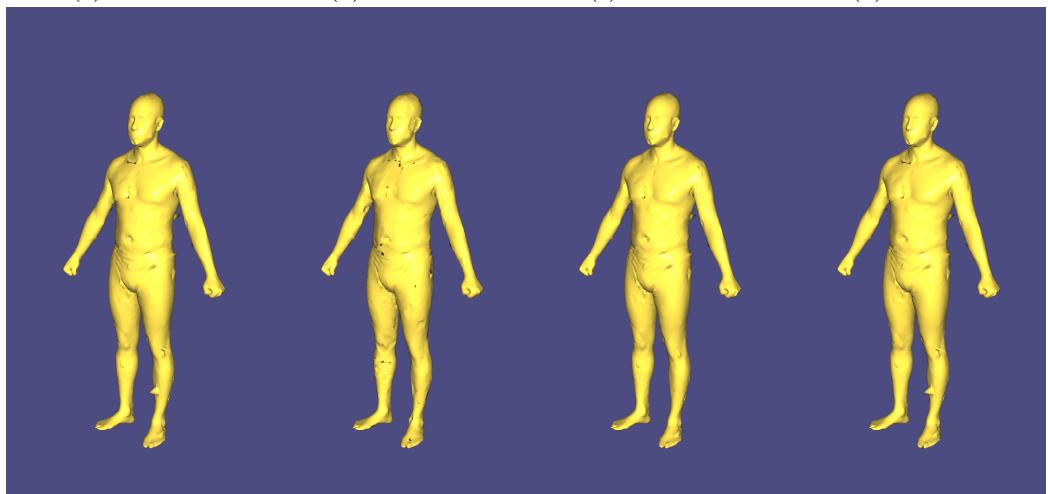
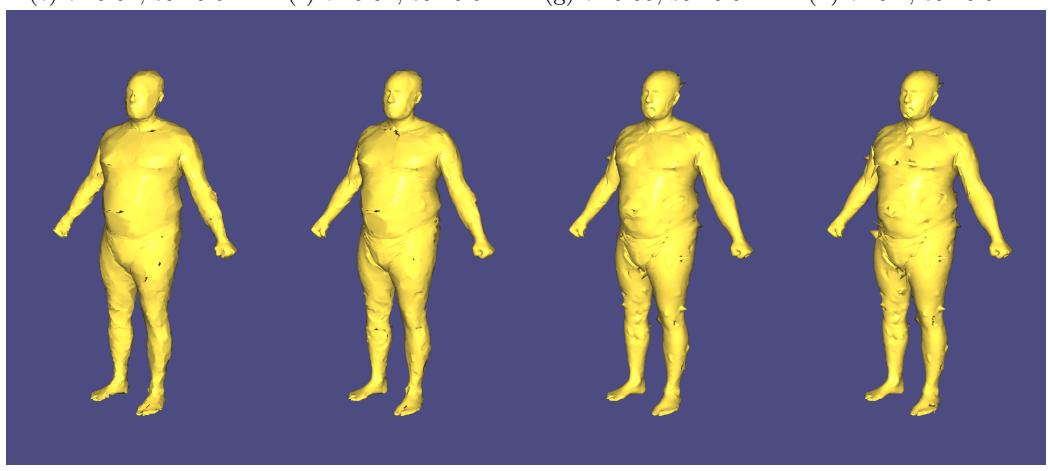
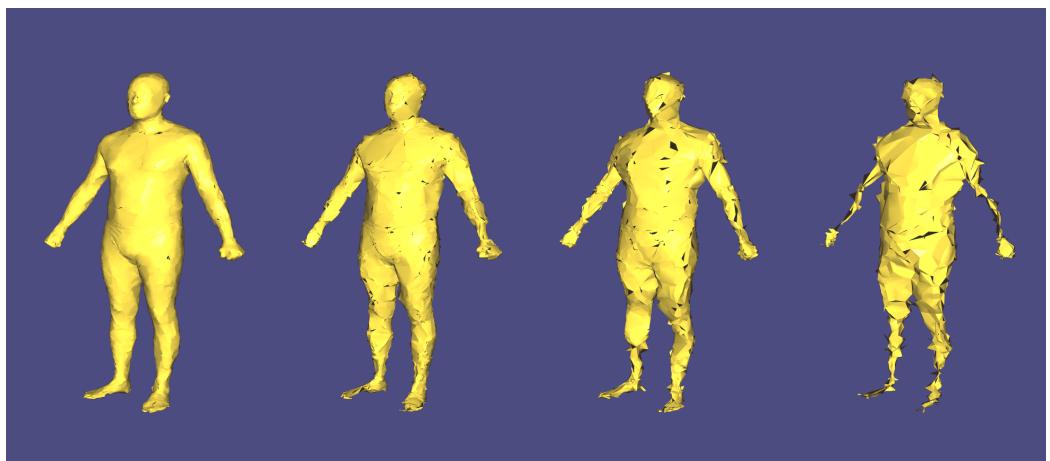
(a)  $t=0.01$ .(b)  $t=0.02$ .(c)  $t=0.05$ .(d)  $t=0.1$ .(e)  $t=0.01, s_0=0.5$ .(f)  $t=0.02, s_0=0.5$ .(g)  $t=0.05, s_0=0.5$ .(h)  $t=0.1, s_0=0.5$ .(i)  $t=0.01, s_2=1.5$ .(j)  $t=0.02, s_2=1.5$ .(k)  $t=0.05, s_2=1.5$ .(l)  $t=0.1, s_2=1.5$ .

Figure 5: Upscaled models with varying threshold values  $t$  and in different scaling factors  $s$ , where the number followed by  $s$  indicates which eigenvector is it corresponding to (when not specified they are equal to 1).



(a) reduction size= 4. (b) reduction size= 12. (c) reduction size= 20. (d) reduction size= 50.

Figure 6: Example showing how results of upscaling change depending on the sampling rate. Reduction size =  $x$  means we sample 1 vertex every  $x$  vertices.

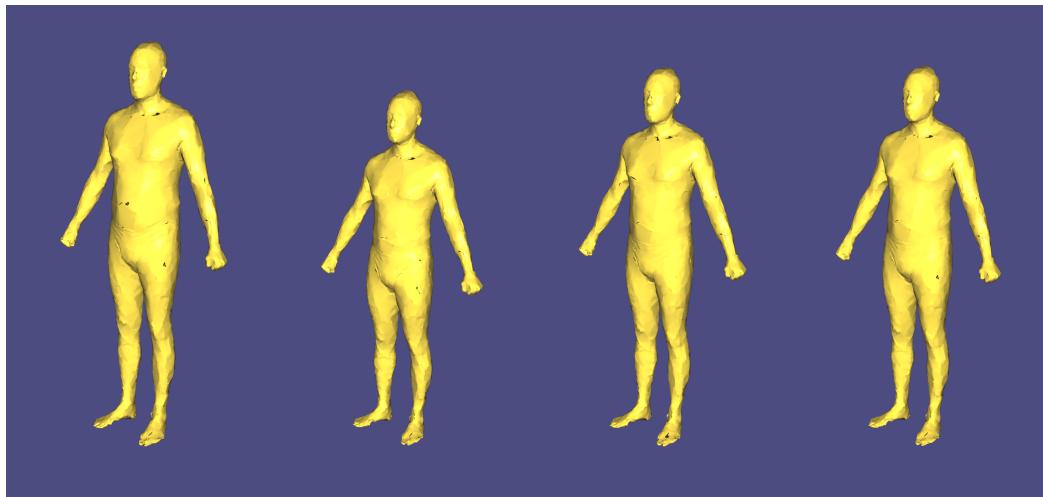
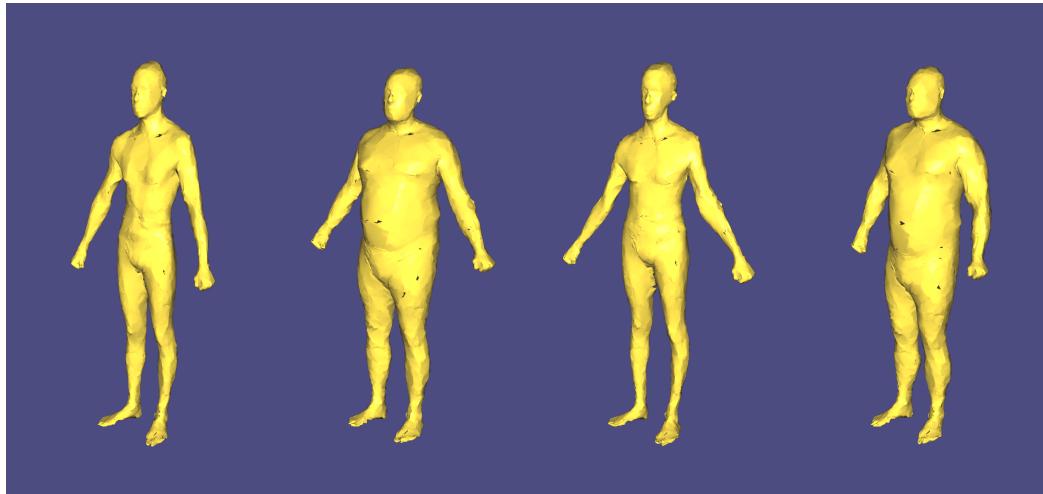
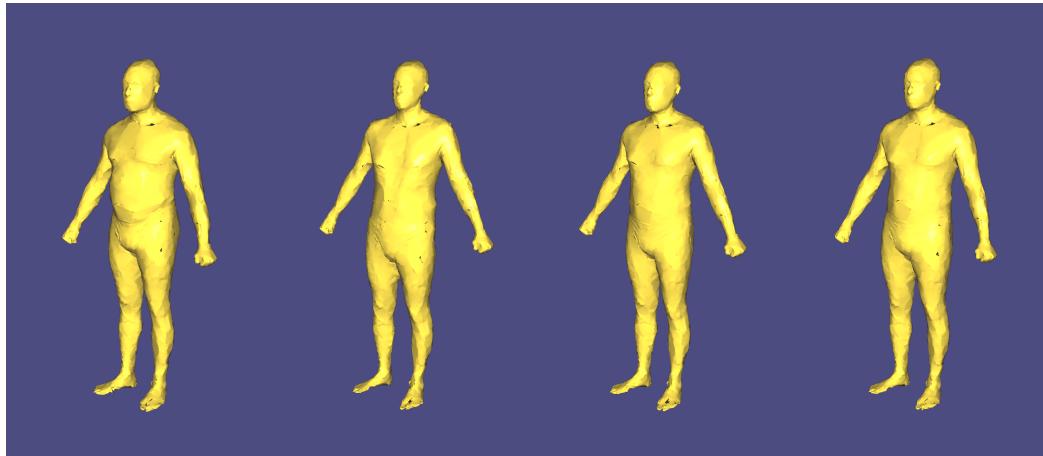
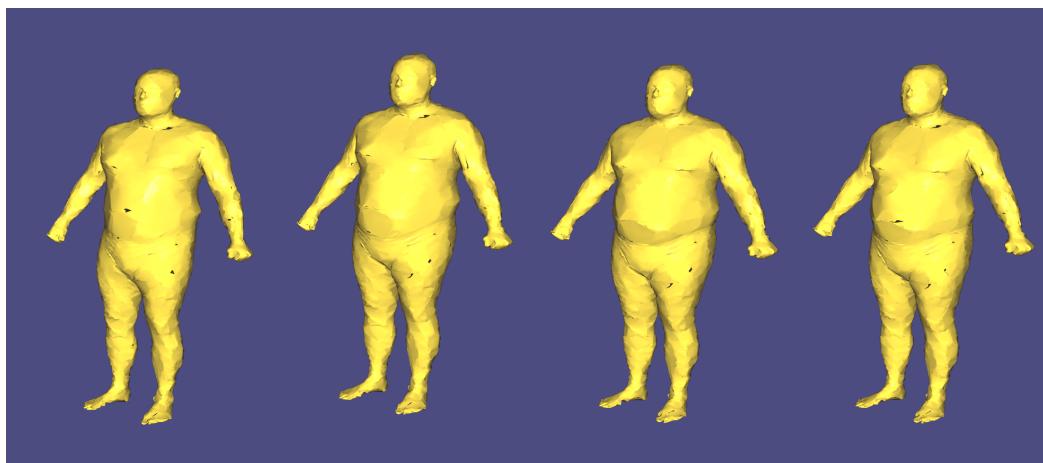
(a)  $s_0=0.5$ .(b)  $s_0=1.5$ .(c)  $s_1=0.5$ .(d)  $s_1=1.5$ .(e)  $s_2=0.5$ .(f)  $s_2=1.5$ .(g)  $s_3=0.5$ .(h)  $s_3=1.5$ .(i)  $s_4=0.5$ .(j)  $s_4=1.5$ .(k)  $s_5=0.5$ .(l)  $s_5=1.5$ .

Figure 7: Showing the features affected by scaling the first eigenvectors by the values  $s$  (the number followed by  $s$  indicates which eigenvector is it corresponding to).



(a) First 50 models. (b) First 100 models. (c) First 200 models. (d) First 500 models.

Figure 8: Example of how mesh reconstruction vary when different number of models are loaded. First 6 scaling factors are set to 1.25.

## References

- [1] BodyModels, CAESAR. <https://graphics.soe.ucsc.edu/data/BodyModels/index.html>. Accessed: 2019-05-25.
- [2] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM transactions on graphics (TOG)*, volume 22, pages 587–594. ACM, 2003.