# Semantify: Simplifying the Control of 3D Morphable Models using CLIP: Supplementary Material

## 1. Background

In this paper, we focus our experiments on four main 3D Morphable Models and their variants. FLAME [1] is a 3DMM for human heads which consists of identity and expression spaces, using $N = 5023$ vertices along with 4 joints. Similarly, SMPL and SMPLX [2] model human bodies using shape and expression spaces, with 23 joints and $N = 6890$ vertices (SMPL), or 54 joints and $N = 10,475$ and vertices (SMPLX). SMAL [3] was constructed using 3D scans of toy animals, and can represent a continuous space of animal shapes.

### 1.1. FLAME

FLAME uses standard vertex-based LBS with corrective shape, with N=5023 vertices and K=4 joints and is described by a function $M$ that returns $N$ vertices:

$$M(\vec{\beta},\vec{\theta},\vec{\psi}) : \mathbb{R}^{|\vec{\beta}|\times|\vec{\theta}|\times|\vec{\psi}|} \rightarrow \mathbb{R}^{3N}$$
$$\forall\vec{\beta} \in \mathbb{R}^{|\vec{\beta}|}, \vec{\theta} \in \mathbb{R}^{|\vec{\theta}|}, \vec{\psi} \in \mathbb{R}^{|\vec{\psi}|}$$

Where $\beta, \theta, \psi$ represent the coefficients of the shape, pose and expression respectively. The variations in the shape of different subjects are modeled by linear blendshapes as:

$$B_S(\vec{\beta}; S) = \sum_{n=1}^{|\vec{\beta}|} \beta_n S_n \tag{1}$$

where $\vec{\beta} = [\beta_1, ..., \beta_{|\vec{\beta}|}]^T$ denotes the shape coefficients and $S = [S_1, ..., S_{|\vec{\beta}|}] \in \mathbb{R}^{3N\times|\vec{\beta}|}$ denotes the orthonormal shape basis. Similarly, the expression blendshapes are modeled by linear blendshapes as

$$B_E(\vec{\psi}; \varepsilon) = \sum_{n=1}^{|\vec{\psi}|} \psi_n E_n \tag{2}$$

where $\vec{\psi} = [\psi_1, ..., \psi_{|\vec{\psi}|}]^T$ denotes the expressions coefficients, and $\varepsilon = [E_1, ..., E_{|\vec{\psi}|}] \in \mathbb{R}^{3N\times|\vec{\psi}|}$ denotes the orthonormal expression basis. In this paper we use the first 10 principal components of the shape $\vec{\beta}$ and the first 10 principal components of the expression $\vec{\psi}$

### 1.2. SMPL/SMPL-X

SMPL-X stands for SMPL eXpressive, with shape parameters trained jointly for the face, hands and body. Similarly to FLAME, SMPL-X uses standard vertex-based LBS with learned corrective blendshapes, with N=10,475 vertices and K=54 joints and is described by a function $M$ that returns $N$ vertices:

$$M(\vec{\beta},\vec{\theta},\vec{\psi}) : \mathbb{R}^{|\vec{\beta}|\times|\vec{\theta}|\times|\vec{\psi}|} \rightarrow \mathbb{R}^{3N}$$
$$\forall\vec{\beta} \in \mathbb{R}^{|\vec{\beta}|}, \vec{\theta} \in \mathbb{R}^{3(K+1)}, \vec{\psi} \in \mathbb{R}^{|\vec{\psi}|}$$

Where $\beta, \theta, \psi$ represent the coefficients of the shape, pose and expression respectively, and the shape blendshapes function is the same as (1). In this paper, we use the first 10 principal components of the shape $\vec{\beta}$.

### 1.3. SMAL

Analogous to SMPL, the SMAL function is also defined by $M(\beta, \theta, \gamma)$ such that $\beta, \theta, \gamma$ represent the shape, pose and translation respectively. $\beta$ is a vector of the coefficients of the learned PCA shape space, $\theta \in R^{3N} = \{r_i\}_{i=1}^{N}$ is the relative rotation of the N=33 joints in the kinematic tree, and $\gamma$ is the global translation applied to the root joint. The SMAL function returns 3D mesh's vertices, where the template model is shaped by $\beta$, articulated by $\theta$ through LBS, and shifted by $\gamma$. In this paper, we use the first 10 principal components of the shape $\vec{\beta}$.

## 2. Word Descriptors

We provide our initial sets of descriptors for each one of the models. Colored words represent the final set of descriptors that were chosen by our method.

### 2.1. Body

The color coding is blue for SMPLX, red for SMPL, and cyan when the word was chosen for Both models.

**Male model:** short, tall, long legs, big, fat, broad shoulders, built, curvy, fit, heavyset, lean, long torso, long, muscular, pear shaped, petite, proportioned, rectangular, round apple, short legs, short torso, skinny, small, stocky, strudy, narrow waist, thin.

**Female model:** fat, thin, hourglass, short, long legs, narrow waist, skinny, tall, broad shoulders, pear shaped, average, big, curvy, lean, proportioned, sexy, fit, heavyset, petite, small.

**Neutral model:** short, tall, long legs, big, fat, curvy, feminine, fit, heavyset, lean, long torso, long, masculine, muscular, pear shaped, petite, proportioned, rectangular, round apple, short legs, short torso, skinny, small, stocky, strudy, attractive, sexy, narrow waist, hourglass.

### 2.2. Face Model

The color coding is cyan for the chosen descriptors.

**Shape:** fat, thin, long neck, big forehead, nose sticking-out, ears sticking-out, small chin, long head, chubby cheeks, big head.

**Expression:** happy, sad, angry, surprised, disgusted, fearful, neutral, smiling, serious, pensive, confused, bored, sleepy, tired, excited, relaxed, calm, nervous, worried, scared, open mouth, raise eyebrows, open eyes, smile.

### 2.3. Animals Model

The color coding is cyan for the chosen descriptors.

**Shape:** hippo, donkey, horse, cow, lion,cat, dog.

## 3. CLIP-Based Optimization

To estimate how a single descriptor will affect the 3DMM with respect to CLIP's semantic understanding, we ran CLIP-based optimization experiments. Figure 1 demonstrates the optimization's results with respect to "smile" descriptor on FLAME model. This method was not a good estimation for the effect of the descriptors for two reasons:

1. In our method CLIP is used for rating (that is, we use CLIP's scores for a given image that has already been deformed and a given set of descriptors), rather than using CLIP as to edit the mesh using its semantic understanding of a given set of descriptors.

2. CLIP-based optimization optimizes a single descriptor each time (feeding multiple descriptors together may enforce putting them in a sentence to optimize CLIP's performance), therefore the relations that appear between descriptors in our model could not be foreseen by using this method.

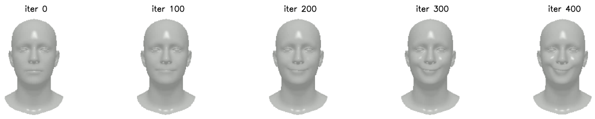

Figure 1. CLIP-based optimization visualization by iteration

## 4. Clustering

Prior to clustering the images in CLIP embedding space's dimension, we used Uniform Manifold Approximation and Projection (U-MAP) to reduce the dimension of the data in order to visualize it and verify that close images resemble each other and distant images differ (an example for such visualization could be found in Figure 2).

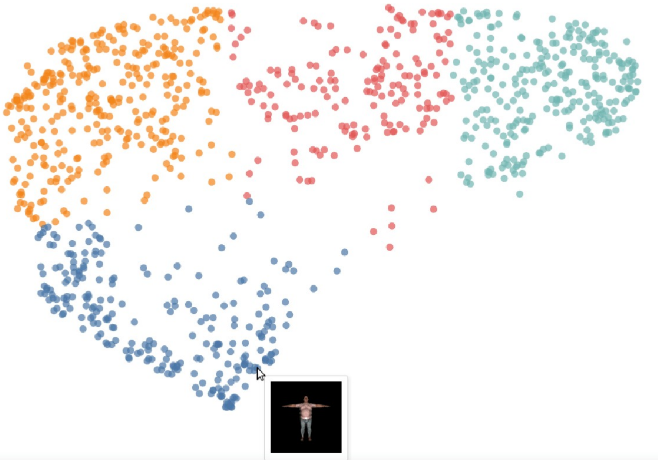

Figure 2. Clustering the encoded images after reducing the dimension by U-Map

## 5. Manually Tuned Model

Our goal in this paper was to present a novel general method for mapping the semantic representation to its parametric counterpart *without human-in-the-loop*. As noted in our conclusions, our method relies mainly on the data that is generated to train the Mapper, therefore, outliers in the data (which in 3D meshes corresponds with "broken" meshes) would probably lead to a degradation in the Mapper's performance. A case-specific solution might produce a better performance since images could be created manually or in a more supervised and case-specific manner. Some examples of such an implementation:

- When using SMAL model, by sampling instances randomly, it is hard to generate instances from the Hippopotamidae family (without "breaking" the mesh), whereas generating it non-randomly is quite an easy task.

|  | number of descriptors |  |  |  |  | number of samples |  |  |
|---|---|---|---|---|---|---|---|---|
|  | 2 | 5 | 6* | 10 | 15 | 1K | 3K* | 10K |
| Error (cm) | 0.0372 | 0.0219 | 0.0233 | 0.0087 | 0.0047 | 0.0247 | 0.0233 | 0.0154 |
| Steps | 568 | 2402 | 2684 | 3769 | 4679 | 2932 | 2684 | 4034 |

Table 1. Model expressiveness ablation study. These are ablations of different mappers that were trained on various numbers of descriptors and different numbers of samples. Columns with * represent our final configurations. Steps indicate the average number of steps that took the optimization process to converge. The error is evaluated on a scale of cm. Although 15 descriptors results in the lowest error, the semantic meaning of the descriptors degrades due to correlations between the descriptors.

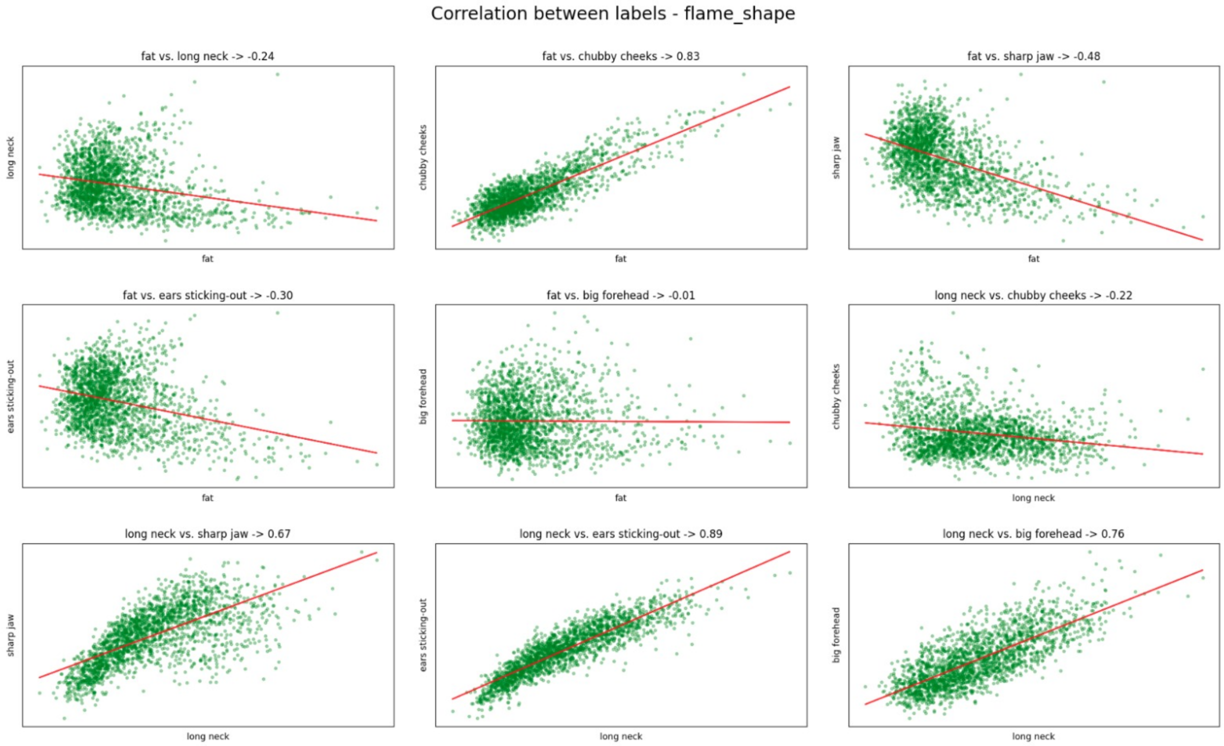

Figure 3. Example for the correlation between shape descriptors (in this case from FLAME 3DMM).

- For the zero-shot shape prediction from an image, it is possible to fine-tune the Mapper by feeding extreme instances or generating specific types of body shapes that would improve the predictions.

In addition, using Semantify as zero-shot predictor of shape from images allows the user to easily fine-tune the results and reach a better fit as can be seen in the examples in Figure 4 where fine-tuning of around one minute was applied to the zero shot results.

## 6. User Study - Feedbacks

We asked the users to provide us with feedback on their experience of using both applications as a tool to fit a 3D shape (A was ours, B was the alternative). Here are more examples of such reviews:

- When using application B, words are sometimes not clear or not relevant, while on the other hand, clear and straightforward words such as “fat” and “thin” are missing.

- Application B’s sliders are not “frozen”, so occasionally when changing one slider it affects the other, which causes the user to start all over again.

- In terms of user experience using the different sliders, application A was far more comfortable to use than B. It was clearer what was meant to be the effect of each slider, and each change influenced that specific body feature alone. Conversely, application B every minor change in a certain slider generated major changes in the rest of the sliders in a way that made it more difficult to control the result. The abundance of sliders on application B only made it harder to control, not the other way around. Ultimately, application B created a figure which was less muscular, and paid little attention to detail in regards to the body curves and body fat. Overall, the results were undeniably better in application A.

- Trying out the application B ... even the slightest touch of a single slider that was intended to get me closer to my desired goal, prompted a significant change in 10 different sliders which completely and utterly ruined what I was aiming for. To the contrary, the experience with application A was far friendlier and I sensed as though I maintained much more control over the different body features. With regards to the final results, you just can't draw a comparison. Juxtaposing the two finalised models makes it humorously obvious that application A wins by a landslide.

| | Number of samples used for training the model | | | |
|---|---|---|---|---|
| | 1k | 3k* | 7k | 10k |
| Error (cm) | 0.0026 | 0.0027 | 0.0022 | 0.0025 |
| Steps | 2850 | 1203 | 4618 | 2328 |

| | Number of descriptors used in the model | | | | |
|---|---|---|---|---|---|
| | 2 | 5 | 6* | 10 | 15 |
| Error (cm) | 0.003 | 0.0028 | 0.0027 | 0.0023 | 0.002 |
| Steps | 1813 | 1466 | 1203 | 4795 | 4999 |

Table 2. Ablation results on 10 meshes that were registered to range scans from CAESER dataset, we used chamfer distance as an objective to fit our sementified sliders with our pretrained mapper to the GT mesh. The real-world captured shapes from CAESER dataset on our SMPL-X **male** mapper.

## References

[1] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017.

[2] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[3] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

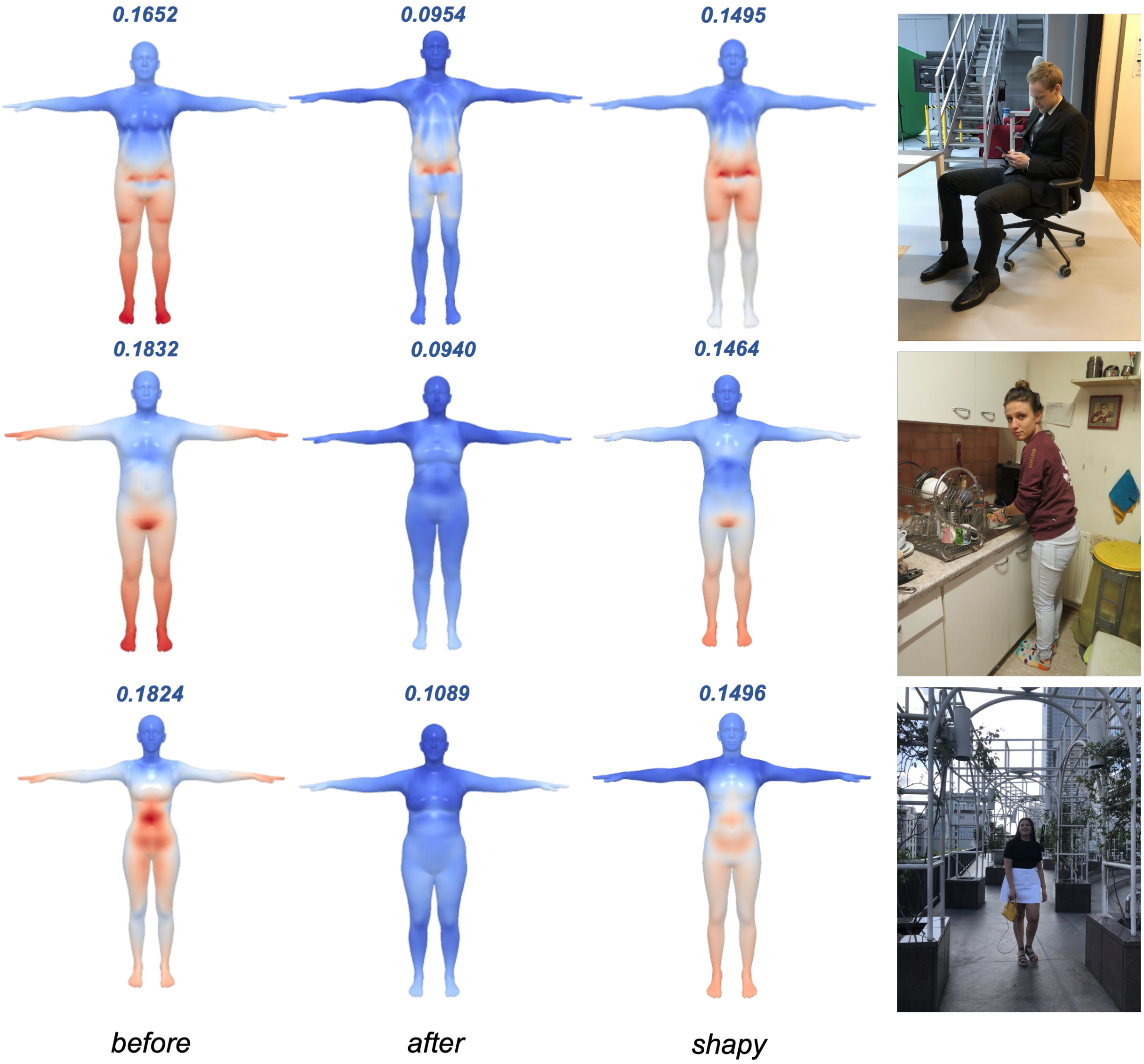

Figure 4. Zero-Shot Image to Shape Reconstruction task with fine-tuning the predicted shape with respect to the image using our interactive application. The initial prediction is on the left, then the fine-tuned shape and on the right is SHAPY's prediction. Fine tuning takes around 1 minute.