A Generative 3D Facial Model by Adversarial Training

Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, Edmond Boyer

A Generative 3D Facial Model by Adversarial Training

Victoria Fernandez Abrevaya[1], Adnane Boukhayma[2], Stefanie Wuhrer[1], Edmond Boyer[1]
[1] Inria - Univ. Grenoble Alpes - CNRS - LJK, France
{victoria.fernandez-abrevaya, stefanie.wuhrer, edmond.boyer}@inria.fr
[2] University of Oxford, UK
adnane.boukhayma@eng.ox.ac.uk

Abstract

We consider data-driven generative models for the 3D face, and focus in particular on factorized representations that can decouple sources of variation, typically identity and expression with faces. Such models provide semantically meaningful parameterizations, but existing methods are still limited in their ability to effectively learn the variability with respect to natural factors, especially when only sparse label information is available. In this work we explore a new direction for this problem by using Generative Adversarial Networks. We build in particular on auxiliary classifier GANs to design a model that maps 3D face shapes into a latent space where identity and expression attributes are explicitly disentangled. Our experiments demonstrate that GANs can contribute with better decoupling performances while achieving competitive model accuracy.

1. Introduction

Generative models of 3D shapes are widely used for their ability to provide compact representations that allow synthesizing realistic objects and their variations according to natural factors. This is particularly true with faces whose 3D shape span low dimensional spaces and for which generative models often serve as priors to solve under-constrained problems, such as reconstruction from partial data, recognition, or identity and expression transfer when these factors are decoupled by the model. Since the pioneering work of Blanz and Vetter [3], numerous approaches have been proposed to build data-driven generative models of the 3D face, *e.g*. [4, 7, 12]. In this line of work, we take inspiration from the success of Generative Adversarial Networks (GANs) in learning to synthesize real images and study how they can extend and contribute to the generation of 3D face shapes.

Data-driven generative 3D face models were first based on linear shape statistics, typically PCA [3], later extended to multi-linear statistics as in [35] to better decouple natural factors. The latter in particular requires fully labeled data where all the training samples are labeled according to some factors of variation, usually identity and expression. With the aim to relax the linear assumption and to better model the underlying structure of 3D face spaces, deep generative models with autoencoder (AE) architectures have recently been proposed. They demonstrate benefits in modeling geometric details [2], non-linear deformations present in facial expressions [28], and increasing robustness to different types of capture noise [12]. They nevertheless have difficulties to effectively decouple shape variations caused by natural factors without resorting to strong supervision, as in *e.g*. [12] where an initialization with fully labeled data is required whose size increases exponentially in the number of factors that are considered.

In this work, we investigate GANs for 3D face modeling and provide insights on their ability to learn shape variations with respect to different factors. In particular, our comparisons with recent approaches based on AE architectures [12, 28] demonstrate that GANs can better decouple identity and expression factors, in addition to enable training with labeled data whose size increases linearly in the number of factors.

To use GANs for learning generative face models we are faced with two challenges. First, GANs are known to be prone to mode collapse that leads to poor generalization. Second, the training is further complicated as we wish to decouple shape variations with respect to face factors. To address these challenges, we propose to split up the latent space into sub-spaces, where each of these is assigned to model the variations caused by one factor. We consider auxiliary classifiers for training that aim to correctly classify the label associated with that factor, *e.g*. identity or expression. This training per factor builds upon the idea recently proposed in [25] and reduces the risk of mode collapse. Training the generative model using only the label information is known to lead to models that do not generalize well. To address this issue, we propose a scheme that alternates between labeled samples and unlabeled ones where the factors in latent space are randomly drawn. To evaluate the classi-

fication of the unlabeled data we introduce a loss on the classifier features that leverages pairs of generated samples.

To summarize, our contributions are:

1. The first work to our knowledge that explores generative adversarial learning for building 3D facial models.
2. A new training scheme that enables to generate expressive faces while decoupling the factors of variation, improving the state of the art on 3D face models.

2. Related Work

Due to the importance of 3D face modeling for various virtual reality applications, numerous works have been proposed to learn expressive generative models. We focus here on data-driven models, often called *3D morphable models* in the literature. Blanz and Vetter [3] use principal component analysis (PCA) to learn the distribution of the facial shape and appearance across different identities scanned in a neutral expression. Subsequent works include different expressions, and model them as decoupled by either adding linear factors [1] or by extending PCA to a multilinear model [35]. Thanks to their simple structure these models are still heavily used, and have recently been extended by training from large datasets and by modeling additional variations including skeletal rotations [7, 4, 21]. To model geometric detail, another line of work has proposed localized data-driven models that take advantage of pyramidal, localized or multi-scale generative models [14, 24, 5, 6].

Autoencoders Recent works leverage deep learning methods to overcome the limitations of (multi-)linear models. Ranjan *et al*. [28] proposed an autoencoder architecture that learns a single global generative model of 3D faces, and as such the different factors cannot be decoupled directly. However, an extension called DeepFLAME is proposed that combines the AE trained using different expressions with a recent linear face model of identity [21]. Fernández Abrevaya *et al*. [12] developed a multilinear AE (MAE) in which the decoder is a multilinear tensor structure. To achieve convergence the tensor needs to be initialized well, which is in practice achieved using a complete labeled training tensor. This implies that the size of labeled training data needed for initialization increases exponentially in the number of factors considered. We compare our proposed approach to DeepFLAME and MAE, as they achieve state-of-the-art results on decoupling identity and expression variations and on model quality.

Bagautdinov *et al*. [2] propose a multiscale model of 3D faces at different levels of geometric detail. They experimentally found that GANs are hard to train for this scenario and resort consequently to variational AEs. Two recent works [34, 33] use AEs to learn a global or corrective morphable model of 3D faces and their appearance based on 2D training data. However, none of these methods allow to disentangle different factors of variation in the latent space. Unlike the aforementioned works, we investigate the use of GANs to learn a generative model of 3D faces.

Generative Adversarial Networks Generative Adversarial Networks [15] are based on a minimax game, in which a discriminator D and a generator G are optimized for competing goals. The discriminator is tasked with learning the difference between real and fake samples, whereas the generator strives to maximize the mistakes of the discriminator. At convergence, G approximates the real data distribution. Training involves the optimization of the following:

$$\begin{aligned}\min_G \max_D \mathcal{L}_{GAN} &= \mathbb{E}_{x\sim p_{data}}[\log D(x)] \\ &+ \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))], \quad (1)\end{aligned}$$

where p_{data} denotes the distribution of the training set, and p_z denotes the prior distribution for G, typically $\mathcal{N}(0, I)$. The generated samples, traditionally images, are shown to be sharper than with models trained using pixel-wise losses.

There have been many improvements in order to increase training stability and sample diversity. DC-GAN [27] is a popular and successful network design proposed to stabilize adversarial learning, and we use this architecture in our work. WGAN [17] tackles mode collapse by minimizing an approximation of the Earth Mover's distance, and we adopt this strategy for adversarial learning.

When labels are available, using them has proven to be beneficial for GAN performance. CGANs [23] uses label information to improve sample quality and generate class-conditioned data. Odena *et al*. [25] proposed Auxiliary Classifier GANs (AC-GAN), in which D is augmented so that it also outputs the probability of an image belonging to one of a set of pre-defined class labels $c \sim p_c$. The loss function for G and D is extended with the terms

$$\mathcal{L}_C^{real} = \mathbb{E}_{x\sim p_{data}, c\sim p_c}[\log P(C = c|x)], \quad (2)$$

$$\mathcal{L}_C^{fake} = \mathbb{E}_{z\sim p_z, c\sim p_c}[\log P(C = c|G(z, c))]. \quad (3)$$

Our work extends this idea to learn a decoupling of the shape variations into factors.

Disentangled representations for GANs In their original form, GANs are unable to explicitly disentangle latent factors according to known features or attributes. Many works have been proposed that decouple by conditioning on an input image (*e.g*. [26, 32]), but only a few do it directly from the latent space. InfoGAN [8] addreses this in an unsupervised manner by maximizing the mutual information between a subset of the latent and the generated data. Mathieu *et al*. [22] combine an encoder-decoder generator and a reconstruction loss to disentangle identity, but only

experimented with very low resolution images. Donahue *et al*. [10] decouples by classifying pairs with a common identity. Neither of these are symmetrical with respect to the two factors to disentangle as they focus on preserving identity only. We propose an alternative that succeeds in decoupling latent codes into a constant number of separate factors.

GANs with 3D faces Two recent methods learn to enhance an input 3D face geometry with photometric information using a GAN. Given a texture map and a coarse mesh, Huyng *et al*. [18] augment the latter with fine scale details, and given an input image and a base mesh, Yamaguchi *et al*. [36] infer detailed geometry and high quality reflectance. Both works require the conditioning of an input, and unlike us they do not build a generative 3D face model.

3. Face Representation

Deep learning can efficiently be applied to regularly sampled signals, such as 2D pixel grids in the image domain. While face shapes live in a 3D space they can be mapped to a 2D domain with limited distortion. We follow such a strategy that allows us to take advantage of existing architectures and losses applied with GANs on 2D images.

2D Representation We consider as input a dataset of registered and rigidly aligned 3D facial meshes, where each mesh is defined by $(\mathcal{V}, \mathcal{F})$, the list of 3D vertices $\mathcal{V} \in \mathbb{R}^{n_v \times 3}$ and the list of triangular faces $\mathcal{F}$ that connect the vertices. Since the data is registered, the connectivity graph $\mathcal{F}$ is fixed across the dataset and any uv parametrization of the graph nodes defines a mapping from 3D to the 2D uv domain that is consistent across all registered shapes. We use such a parameterization where each vertex of the mesh can be mapped to an image pixel using the uv coordinates, and where the pixel values correspond to the (x, y, z) vertex coordinates. Continuous image representations for faces are obtained by interpolating the (x, y, z) values between vertex nodes in the uv image. Borrowing the term from [16], we call this representation a geometry image (see Fig. 1). To make the representation translation-invariant, instead of storing the (x, y, z) coordinates directly we store the displacements with respect to a mean face mesh. Furthermore, we quantize the pixel values to the $[0, 255]$ interval across the complete training set, thus obtaining standard 3-channel 2D image arrays.

3D Reconstruction To reconstruct a 3D face mesh from a given geometry image, we can directly pick the mesh vertex locations from their corresponding pixel values in the image. However, in practice this sparse strategy tends to produce artifacts as a result of the interpolation, particularly around the border, and of the (x, y, z) quantization. We use instead a dense approach where each pixel contributes and where the reconstruction is regularized with Laplacian shape deformation constraints by solving:

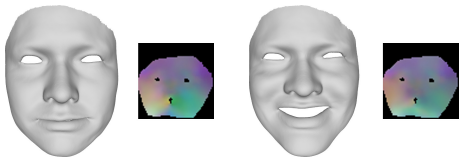


Figure 1: Two 3D faces and the corresponding geometry images

$$\arg\min_{\mathbf{v}} \sum_{(i,j)\in\mathcal{P}} ||\mathbf{v}_{i,j} - \mathbf{p}_{i,j}||_2^2 + \alpha||L\bar{\mathbf{v}} - L\mathbf{v}||_2^2. \quad (4)$$

$\mathcal{P}$ is the set of non-zero pixels, L is the cotangent discretization of the Laplace operator, $\bar{\mathbf{v}}$ are the vertices of the mean face, $\mathbf{p}_{i,j}$ is the pixel value at location (i, j), $\mathbf{v}_{i,j}$ is the point on the 3D surface associated with $\mathbf{p}_{i,j}$, computed using barycentric coordinates, and α is a weight controlling the influence of the regularization. This formulation results in a sparse linear system that can be solved efficiently, and has the benefit of producing meshes of higher quality without requiring high-resolution images.

4. Method

Given geometry images of faces, our goal is to build an expressive model that can decouple the representation based on known factors of variation. We take a detour from classical approaches in which a reconstruction error is optimized, and instead rely on the discriminative power of deep convolutional networks. We propose extensions over the original AC-GAN formulation with the goal of improving the model in terms of generalization and decoupling. The training and model fitting phases are explained below.

4.1. Training

Figure 2 depicts our architecture that is trained by considering both the adversarial loss for generating realistic geometry images and auxiliary classifiers for each factor of variation. To simplify notation, we will consider here a model that decouples between identity and expression, however the principle can be easily extended to more factors. In the following we detail the discriminator and generator losses used during training.

4.1.1 Discriminator

As seen in Figure 2, the discriminator consists of three branches: the real/fake classifier, as in a standard GAN, along with one classifier per factor of variation, identity and expression in this work. This follows the method of [25],

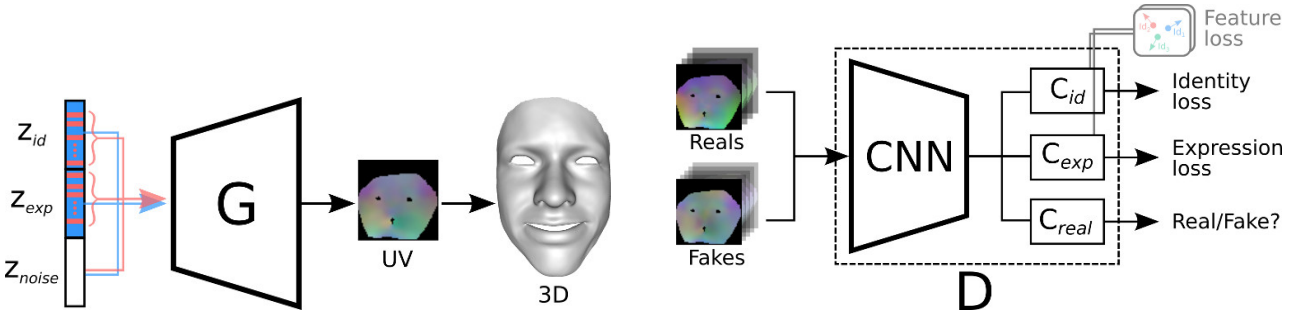


Figure 2: The generator G and discriminator D of our GAN model. Identity and expression codes z_{id}, z_{exp} are used to control the generator, and classification losses are added to decouple between the two. Alternate training, illustrated in blue and red arrows, allows to correctly generate outside labels. A feature loss is introduced to ensure consistency over features with fixed identities or expressions.

with two major differences. First, we provide classifiers for each of the factors, as opposed to classifying only one type of label. This encourages decoupling since the classification of one factor is independent of the choice of the labels for the other factors. Second, we provide distinct convolutional layers for each of these branches. This is motivated by the observation that the features needed to classify identities and expressions are not necessarily the same.

The discriminator is trained to *maximize*:

$$\mathcal{L}_D = \mathcal{L}_{GAN} + \lambda_C(\mathcal{L}_{ID} + \mathcal{L}_{EXP}). \quad (5)$$

Here, $\mathcal{L}_{GAN}$ denotes the standard adversarial loss (see Equation 1), and $\mathcal{L}_{ID}, \mathcal{L}_{EXP}$ the classification losses measured against the labels provided with the dataset and weighted by scalar λ_C. The latter are defined similarly to Equation 2 as:

$$\begin{aligned} \mathcal{L}_{ID} &= \mathbb{E}_{x\sim p_{data}, c\sim p_c^{id}}[\log P(C=c|x)], \\ \mathcal{L}_{EXP} &= \mathbb{E}_{x\sim p_{data}, c\sim p_c^{exp}}[\log P(C=c|x)], \end{aligned} \quad (6)$$

where p_c^{id} and p_c^{exp} denote the distribution of identity and expression labels, respectively. To account for missing labels with real data samples, we ignore the sample contribution in the classification loss associated to each missing label. In the remainder of the paper, we refer to the three branches of D as C_{real}, C_{id}, and C_{exp}, respectively.

4.1.2 Generator

The generator G takes as input a random vector $z = \{z_{id}, z_{exp}, z_{noise}\}$, which is the concatenation of the identity code $z_{id} \sim p_{id}$, the expression code $z_{exp} \sim p_{exp}$ and a random noise $z_{noise} \sim p_{noise}$. It produces geometry images that encodes face shapes as explained in Section 3. G is trained by minimizing:

$$\begin{aligned} \mathcal{L}_G = \lambda_1\mathcal{L}_{GAN} - \lambda_2\left(\mathcal{L}_{CLASS}^{id} + \mathcal{L}_{CLASS}^{exp}\right) \\ +\lambda_3\left(\mathcal{L}_{FEAT}^{id} + \mathcal{L}_{FEAT}^{exp}\right) + \lambda_4\mathcal{L}_{TV}, \end{aligned} \quad (7)$$

where: $\mathcal{L}_{GAN}$ is the standard GAN loss (Equation 1); $\mathcal{L}_{CLASS}^{id}$ and $\mathcal{L}_{CLASS}^{exp}$ are classification losses; $\mathcal{L}_{FEAT}^{id}$ and $\mathcal{L}_{FEAT}^{exp}$ are features losses that aim to decouple the factors; $\mathcal{L}_{TV}$ is a regularizer; and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are weights for the different loss terms.

Classification Loss In addition to the adversarial training, the generator is also trained to classify its samples with the correct labels by maximizing:

$$\begin{aligned} \mathcal{L}_{CLASS}^{id} &= \mathbb{E}_{z\sim p_z, c\sim p_c^{id}}[\log P(C=c|G(z))] \\ \mathcal{L}_{CLASS}^{exp} &= \mathbb{E}_{z\sim p_z, c\sim p_c^{exp}}[\log P(C=c|G(z))]. \end{aligned} \quad (8)$$

In order to generate data belonging to a specific class, we sample one identity/expression code z_{id}, z_{expr} for each label and fix it throughout the training; this becomes the input for G each time the classification loss must be evaluated. We denote the set of fixed codes for identity and expression as $\mathcal{T}^{id}$ and $\mathcal{T}^{exp}$ respectively.

Feature Loss The classification loss is limited to codes in $\mathcal{T}^{id}/\mathcal{T}^{exp}$, which have associated labels. In order to account for the other codes (that is, random z for which the label is unknown), we propose to generate samples in pairs which share the same identity or expression vector, and measure the decoupling error as:

$$\mathcal{L}_{FEAT}^{id} = \frac{2}{N}\sum_{z_{id}}\left(1 - cos(\mathbf{f}_{1,z_{id}}, \mathbf{f}_{2,z_{id}})\right), \quad (9)$$

$$\mathcal{L}_{FEAT}^{exp} = \frac{2}{N}\sum_{z_{exp}}\left(1 - cos(\mathbf{f}_{1,z_{exp}}, \mathbf{f}_{2,z_{exp}})\right), \quad (10)$$

where N is the batch size, and $\mathbf{f}_{i,z_{id}} = \mathbf{f}\left(G(z_{id}, z_{exp,i}, z_{noise,i})\right)$ are feature vectors obtained by inputting the sample $G(z_{id}, z_{exp,i}, z_{noise,i})$ through the classifier C_{id} and extracting the features from the second to

last layer. That is, given two inputs which were generated with the same identity vector, $\mathcal{L}^{id}_{FEAT}$ enforces that their feature vectors in the identity classifier also agree. The definition is analogous for $\mathbf{f}_{i,z_{exp}}$ with C_{expr} .

The feature loss also allows us to remove the classification loss on fake samples while training D (Equation 3), which classically tends to saturate quickly and to provide hence poor gradients when training G, therefore decreasing the quality of decoupling.

Regularization To encourage smoothness, the last component of our loss function is a regularization cost applied on the output geometry images and composed of an anisotropic Total Variation (TV) term that favors homogeneous regions in the images:

$$\mathcal{L}_{TV} = \sum_{i,j} |\mathbf{p}_{i+1,j} - \mathbf{p}_{i,j}| + |\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j}|. \quad (11)$$

4.1.3 Alternate Training

Using auxiliary classifiers has been shown to help stabilize training and avoid mode collapse, as also confirmed in our experiments. However, it can lead to the generation of unrealistic geometry images when sampling away from the pre-defined labels. To address this, we propose an alternate scheme for training as follows.

We distinguish here between two types of samples: *labeled codes*, which are drawn from the sets $\mathcal{T}^{id}$ or $\mathcal{T}^{exp}$ and have a label associated with them, and *unlabeled codes*, which are simply generated from the latent distributions p_{id} or p_{exp}. We then alternate for each batch iteration between the sampling of labeled identity codes with unlabeled expression codes, and the sampling of unlabeled identity codes with labeled expression codes. This allows to better cover the identity and expression sub-spaces during training. The classification is evaluated for the labeled factor only, while the feature loss is used for unlabeled codes.

4.2. Model fitting

Given a trained GAN and a novel registered face, we now explain how to obtain the latent code z that best explains the corresponding input geometry image. Unlike PCA models for which a closed-form solution exists, or recent deep-learning based models that use an encoder to obtain the coefficients, *e.g.* [12, 2, 28], we need to find the best code through an optimization approach. In practice, we optimize the reconstruction error directly against the input mesh by sampling the geometry image according to the uv mapping.

Denote by $\{\tilde{\mathbf{v}}_i(z), i \in [1, n_v]\}$ the set of vertex coordinates obtained by sampling a geometry image $G(z)$ using the known uv map and bilinear interpolation. Given the input mesh vertices $\{\mathbf{v}_i, i \in [1, n_v]\}$, we estimate z as:

$$\arg\min_z \sum_{i=1}^{n_v} ||\tilde{\mathbf{v}}_i(z) - \mathbf{v}_i||_2 + \lambda ||z||_2^2, \quad (12)$$

with $\lambda = 0.0001$. We use a gradient descent approach [20] starting from a randomly sampled z. In case of divergence, multiple random seeds for z are used; at most two seeds were required in our experiments.

5. Results

We now provide results using our GAN framework and demonstrate that our model can achieve better decoupling than state-of-the-art works. We start by giving implementation details and the datasets used. In Section 5.3 we describe the evaluation metrics we propose, which will be used to show that despite the known difficulties in training GANs our model achieves competitive results. In Section 5.4 we perform ablation studies to show that all components are necessary to effectively train a rich model. In Section 5.5 we compare to state-of-the-art 3D face models, and show that we outperform in terms of decoupling while still achieving competitive results on the remaining metrics. More visual results can be found in the supplemental material.

5.1. Implementation Details

We set the weights to $\alpha = 1.2$ (Equation 4), $\lambda_C = 0.1$ (Equation 5), $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 0.5$ and $\lambda_4 = 0.0001$ (Equation 7). The classification losses are further weighted to account for unbalanced labels [19]. We use the architecture from DC-GAN [27], with the first two convolutional blocks shared between C_{real}, C_{id} and C_{expr}, while the remaining are duplicated for each module. The models were trained for 200 epochs using ADAM optimizer [20] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, a learning rate of 0.0002 and a batch size of 64. We train the discriminator for 3 iterations each time we train the generator. The models take around 2 hours to train on a NVidia GeForce GTX 1080 GPU.

The template mesh used for registration of the training data contains 22129 vertices, and the 3D mesh along with its associated uv map was obtained from the BiWi dataset [11]. We generate geometry images of size 64×64. The dimensions for $(z_{id}, z_{exp}, z_{noise})$ are set to $(65, 15, 5)$ to facilitate comparison with [12], and the feature vectors used in Equations 9 and 10 are of size 2048.

5.2. Datasets

All models were trained using a combination of four publicly available 3D face datasets. In particular, we use two datasets containing static 3D scans of multiple subjects: BU-3DFE [38] and Bosphorus [30], and combine these with two datasets of 3D motion sequences of multiple subjects: BP4D-Spontaneous [39] and BU-4DFE [37]. The static

datasets provide variability of identities, while the motion ones provide variability of expressions. The static data is registered with a template fitting approach [29] and the motion data is registered using a spatiotemporal approach [13].

The final dataset contains 30559 registered 3D faces and was obtained by subsampling the motion sequences. We provide identity labels for all meshes, while the expression labels are limited to the seven basic emotional expressions, labeled in both static datasets. For BU-4DFE, expression labels are assigned to three frames per sequence: the neutral expression to the first and last frame, and the labeled expression of the sequence to the peak frame. For BP4D, one neutral frame is labeled per subject (this is a requirement for comparison to [28]). Overall, only 7% of the data is assigned expression labels. As test set, we leave out 2 identities from BP4D and 7 identities from BU-4DFE which we further subsample, for a total of 175 registered 3D faces.

5.3. Evaluation Metrics

The evaluation of generative models in the image domain is an active research area. To quantitatively analyze the quality of the model we resort in this work to traditional metrics used for generative models of 3D shapes. Furthermore, we propose a protocol to measure both decoupling and diversity which is inspired from those in the GAN literature. In detail, we use the following metrics.

Model quality Generative models of 3D shapes are commonly evaluated by measuring *generalization* and *specificity* [9]. Generalization measures how well the model can explain unseen faces, by computing the mean vertex distance obtained after projection into the model. We measure generalization by optimizing as explained in Section 4.2 on the testing set. Complementary to this, specificity measures whether randomly generated samples belong to the modeled shape class, faces in our case. For this, n samples are randomly drawn from the model and for each the mean vertex distance to each member of the training set is measured, keeping the minimum value. The metric then reports the mean of the n values. We set $n = 1000$. We use generalization and specificity to indirectly evaluate whether the GAN converged to a model that produces realistic shapes.

Decoupling To evaluate decoupling in both identity and expression spaces we follow the protocol proposed in [10], which we adapt to our problem. In particular, we first train two networks (one for identity and one for expression) that transform an image representation of the mesh to an n-dimensional vector using triplet loss [31] (we use $n = 128$ in this work). The trained networks allow to measure whether two meshes share the same identity or expression by checking whether the distance between their embeddings is below a threshold τ.

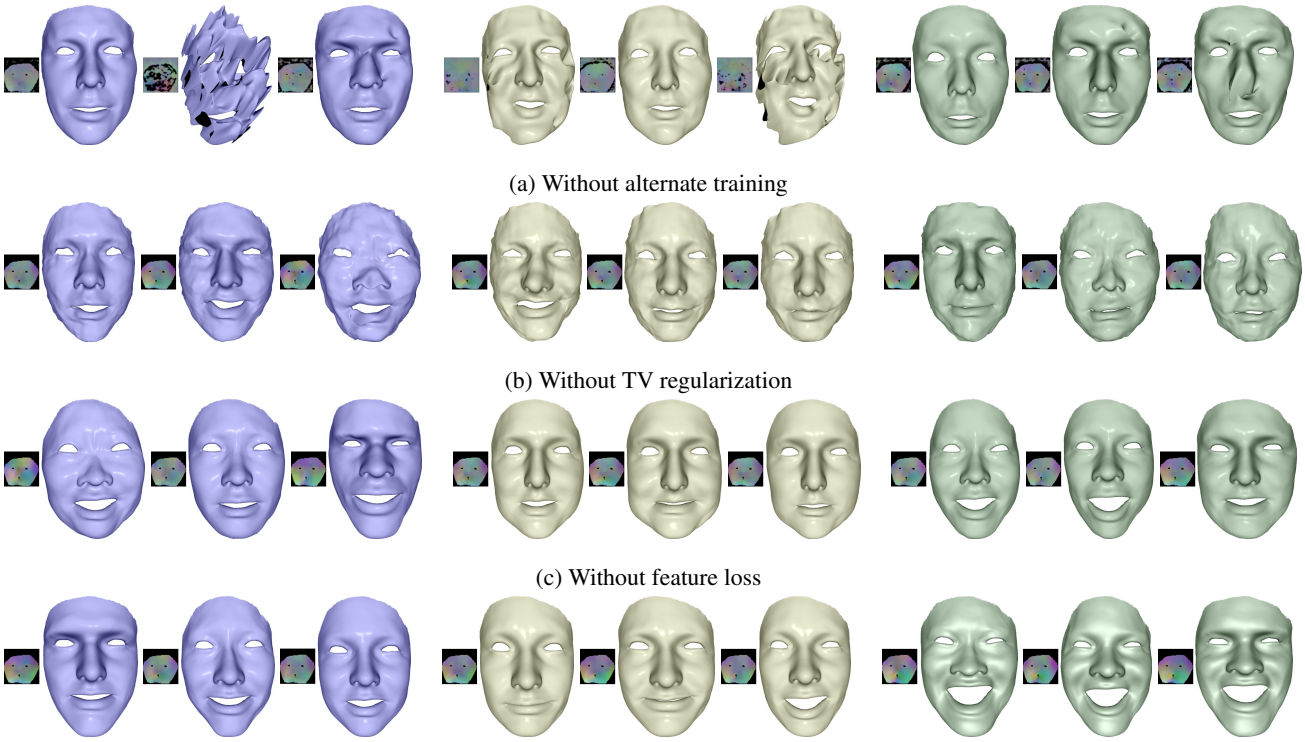
To measure identity decoupling we generate n random faces $\mathbf{x}_i = G(z^i_{id}, z^i_{exp}, z^i_{noise})$, and for each random face we fix the identity code and sample m faces $\mathcal{Y}(\mathbf{x}_i) = \{G(z^i_{id}, z^j_{exp}, z^j_{noise}), j \in [1, m]\}$ with fixed identity codes. We then use the embedding networks to evaluate whether the original face $\mathbf{x}_i$ and each of the samples in $\mathcal{Y}(\mathbf{x}_i)$ are classified as a same identity, and report the percentage of times the pairs were classified as "same". We proceed analogously for expression decoupling. We set $n = 100$, $m = 100$, $\tau = 0.14$ for identity and $\tau = 0.226$ for expression; more implementation details can be found in the supplemental material.

Diversity It is known that GANs can exhibit mode collapse, and thus we consider it important to evaluate the diversity of the generated meshes, which to the best of our knowledge has not been considered before in the context of 3D face models. We take inspiration from the metric proposed in [25], which measures the multi-scale structural similarity (MS-SSIM) score between randomly chosen pairs of samples, and contrasts the MS-SSIM of the samples to the MS-SSIM of the training set. Our preliminary experiments with this score showed that the values saturated above 0.95 and thus provided little information on our specific application. Instead, we sample p pairs of randomly generated meshes and measure diversity as the *mean vertex distance* among the pairs. We expect here to see higher values for more diverse models. We evaluate on three sets of sampled pairs: (1) among pairs chosen randomly (*global diversity*), (2) among pairs that share the same identity code (*identity diversity*) and (3) among pairs that share the same expression code (*expression diversity*). For all cases we evaluate on 10000 pairs. For comparison, the training set is also evaluated on these three metrics by leveraging the labels.

5.4. Ablation Studies

Training GANs is known to be difficult, and we start by demonstrating that each of the proposed components is necessary to obtain state of the art quality of the generated samples and of decoupling. To this end, we compare our approach against five alternatives: (1) standard AC-GAN; (2) our GAN without alternate training (Section 4.1.3); (3) our GAN without TV regularization (Equation 11); (4) our GAN without feature loss (Equations 9 and 10); and (5) our GAN with identity classification only, as approached by most related works in disentangled representations.

Table 1 gives the evaluation metrics for each of these options, and Figure 3 provides visual examples for some of these variants. From the results we observe that: (1) AC-GAN alone is not sufficient to obtain good quality samples without conditioning on a label. This is reflected by a very high specificity value, caused by random samples that are not always realistic geometry images. The high levels of



(a) Without alternate training

(b) Without TV regularization

(c) Without feature loss

(d) Proposed

Figure 3: Comparison of alternative approaches. From left to right: randomly generated samples (purple), random samples that share the identity code (grey), random samples that share the expression code (green). For each sample, the geometry image and its 3D reconstruction are shown.

noise lead to an artificially high diversity value, which is much larger than the training set. (2) The alternate training is crucial to generate samples that are realistic facial shapes. This is reflected by a large value in specificity as well as a poor decoupling in identity. Note also the high levels of noise that do not correspond to valid geometry images (see Figure 3a). (3) The absence of TV regularization results in generated faces with geometric noise. This effect can be observed in higher values for generalization and specificity compared to the proposed approach, and are shown qualitatively in Figure 3b. (4) Without feature loss, the model still achieves good results in terms of model quality and decoupling, but the expression classification accuracy is significantly lower. This is shown qualitatively on the right of Figure 3c, where models with the same expression code lead to faces with different expressions. Note that the expression space is more challenging than identity space, as the provided labels are sparse. (5) Using classification only in identity space results in good identity decoupling, but the decoupling of expression is poor. The low number in identity diversity also suggests that the network incurred in mode collapse for this space, generating minor variations when the identity code is fixed.

Our proposed approach performs best in generalization, specificity and expression decoupling, while still achieving high identity decoupling. This is visually demonstrated in Figure 3d, where fixing either identity or expression codes leads to plausible results.

5.5. Comparisons

Finally, we compare our approach against state of the art generative 3D face models. One of the main features of our model is its ability to decouple, and we thus focus our comparison on works that either enforce this explicitly [12], or combine a model trained on expressions with a linear space of identities [28]. We train all models using the same dimensions (65 for identity and 20 for expression).

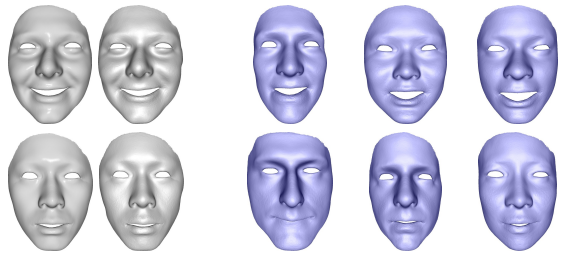
The model proposed in [12], called MAE in the following, was trained with the same dataset as our model and the same label information for 200 epochs, with default parameters from the paper. We initialize encoder and decoder from the publicly available models, and obtain generalization numbers by the same optimization approach as in [12].

The model proposed in [28], called COMA in the following, does not explicitly encourage decoupling and thus we use the DeepFLAME alternative [21], which we train

	Dec-Id	Dec-Expr	Sp.	Gen.	Div-Global	Div-Id	Div-Expr
Training data	–	–	–	–	4.89 ± 1.47	3.30 ± 1.97	5.04 ± 1.54
AC-GAN	25.8	88.1	5.23	0.77	14.35 ± 11.27	12.28 ± 12.02	6.98 ± 3.85
Proposed w/o alternate training	37.9	88.9	4.69	0.76	9.52 ± 6.94	7.53 ± 7.94	6.30 ± 3.26
Proposed w/o TV loss	97.8	89.4	2.08	0.98	4.52 ± 1.17	2.10 ± 0.92	3.86 ± 1.10
Proposed w/o feature loss	99.7	89.1	1.93	0.78	4.34 ± 1.22	1.86 ± 0.77	3.83 ± 1.13
Proposed w/o expression classif.	100	53.3	1.99	0.82	4.57 ± 1.26	0.41 ± 0.08	4.58 ± 1.27
MAE [12]	**99.5**	53.3	2	**0.31**	3.89 ± 1.05	0.92 ± 0.35	3.76 ± 1.02
COMA [28]	97.5	64.1	2.42	0.42	3.34 ± 0.89	1.65 ± 0.40	2.91 ± 0.89
Us	99.1	**93.4**	**1.93**	0.74	$\mathbf{4.54 \pm 1.28}$	$\mathbf{2.08 \pm 0.97}$	$\mathbf{3.90 \pm 1.15}$

Table 1: Quantitative evaluation with respect to generalization (*Gen.*, in mm.), specificity (*Sp.*, in mm.), decoupling of identity and expression (*Dec-*, percentage) and diversity (*Div-*, in mm). Higher is better, except for generalization and specificity.

(a) MAE



(b) COMA

(c) Us

Figure 4: Comparison against MAE and COMA. From left to right : (1) input mesh; (2) reconstruction of each model; (3) synthetic samples obtained by fixing the expression code from the reconstruction (in purple).

with the same dataset as our model. This results in a PCA model built from 299 identities and an autoencoder trained on 30330 displacements from the corresponding neutral face. For the identity space we manually selected one neutral frame for each sequence in BP4D-Spontaneous, as this dataset does not provide labels. The model was trained using the publicly available code for 150 epochs. We obtain generalization results by subtracting the ground-truth neutral face.

We show quantitative results with respect to the proposed metrics in the bottom of Table 1. Note that our approach significantly outperforms the other two in terms of *expression decoupling*, which is more challenging than identity due to sparse label information provided with the training data. This is shown qualitatively in Figure 4, where the expression is preserved well by our model when changing the other codes. With respect to *identity decoupling* the three methods perform similarly well, and while MAE achieves the best value, we note that this was facilitated by a low diversity when generating variations of a same subject, as suggested by the significantly lower value on *Div-Id*. Our model also outperforms in terms of *specificity*, with MAE performing closely well, suggesting that random samples from our GAN can correctly represent the human facial shape, as defined by our training set. The *generalization* result, while still being a close fit to the data with a mean error below $1mm$, does not achieve the best value compared to the related work. This could be explained by the coarser representation used, and the fact that a reconstruction error is never optimized during training. Still, we can observe in Figure 4c that the generated faces closely resemble the input.

6. Conclusion

In this work we explored the use of GANs as an alternative for 3D facial modeling, and proposed a training scheme capable of decoupling the factors of variation present in the facial shape. The experimental results show that the model outperforms state of the art with respect to decoupling, while achieving competitive results in the rest of the

metrics. We believe this work opens up exciting new directions for building generative models of the 3D face.

References

[1] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3d face recognition with a morphable model. In *Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008. 2

[2] T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. Modeling Facial Geometry using Compositional VAEs. In *Conference on Computer Vision and Pattern Recognition*, volume 1, page 1, 2018. 1, 2, 5

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, 1999. 1, 2

[4] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway. A 3d morphable model learnt from 10,000 faces. In *Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2

[5] A. Brunton, T. Bolkart, and S. Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *European Conference on Computer Vision*, 2014. 2

[6] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *Transactions on Graphics*, 2015. 2

[7] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: a 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 3:413–425, 2014. 1, 2

[8] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016. 2

[9] R. Davies, C. Twining, and C. Taylor. *Statistical models of shape: Optimisation and evaluation*. Springer Science & Business Media, 2008. 6

[10] C. Donahue, Z. C. Lipton, A. Balsubramani, J. McAuley, S. Rezvani, N. Mokari, M. R. Javan, M. H. Salas-Olmedo, J. C. Garcia-Palomares, J. Gutierrez, et al. Semantically decomposing the latent spaces of generative adversarial networks. *International Conference on Learning Representations*, 2018. 3, 6

[11] G. Fanelli, J. Gall, H. Romsdorfer, T. Weise, and L. Van Gool. A 3-d audio-visual corpus of affective communication. *IEEE Transactions on Multimedia*, 12(6):591–598, 2010. 5

[12] V. Fernández Abrevaya, S. Wuhrer, and E. Boyer. Multilinear autoencoder for 3d face model learning. In *Winter Conference on Applications of Computer Vision*, pages 1–9, 2018. 1, 2, 5, 7, 8

[13] V. Fernández Abrevaya, S. Wuhrer, and E. Boyer. Spatiotemporal modeling for efficient registration of dynamic 3d faces. In *International Conference on 3D Vision*, pages 371–380, 2018. 6

[14] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser. A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics*, 25(3):1025–1034, 2006. 2

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[16] X. Gu, S. Gortler, and H. Hoppe. Geometry Images. *ACM Transactions on Graphics*, 21(3), 2002. 3

[17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 2

[18] L. Huynh, W. Chen, S. Saito, J. Xing, K. Nagano, A. Jones, P. Debevec, and H. Li. Mesoscopic facial geometry inference using deep neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 8407–8416, 2018. 3

[19] G. King and L. Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137–163, 2001. 5

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015. 5

[21] T. Li, T. Bolkart, M. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics*, 36(6):194:1–17, 2017. 2, 7

[22] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016. 2

[23] M. Mirza and S. Osindero. Conditional generative adversarial nets. Technical report, arXiv:1411.1784, 2014. 2

[24] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics*, 32:179:1–10, 2013. 2

[25] A. Odena, C. Olah, and J. Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *International Conference on Machine Learning*, 2017. 1, 2, 3, 6

[26] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018. 2

[27] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 2, 5

[28] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3d faces using convolutional mesh autoencoders. In *European Conference on Computer Vision*, 2018. 1, 2, 5, 6, 7, 8

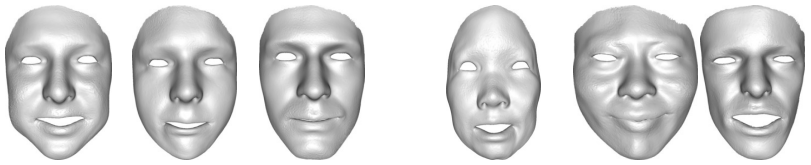
[29] A. Salazar, S. Wuhrer, C. Shu, and F. Prieto. Fully automatic expression-invariant face correspondence. *Machine Vision and Applications*, 25(4):859–879, 2014. 6

[30] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56. Springer, 2008. 5

A Generative 3D Facial Model by Adversarial Training
Supplementary Material

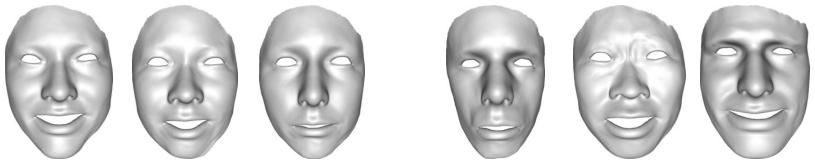
Qualitative Comparisons

We provide qualitative examples for the results in Section 5.5, Table 1. Figure 5 shows three random samples with best and worst *specificity* values, and Figure 6 provides reconstructions from the testing set, as used for the evaluation of *generalization*. In Figures 7 and 8 we show examples used for *decoupling* and *diversity* evaluation of identity and expression, respectively.



(a) COMA

(b) MAE



(c) Us

Figure 5: Random samples which obtained the three best (left) and worst (right) values in the specificity metric.

Latent Space Manipulation

We show here two sets of examples obtained by manipulating the latent space.

First, given a source mesh obtained with $G(z_{id}^{src}, z_{expr}^{src}, z_{noise}^{src})$ and a target mesh obtained with $G(z_{id}^{target}, z_{expr}^{target}, z_{noise}^{target})$ we generate new expressions for the target mesh by either

1. Replacing the expression with that of the source: $G(z_{id}^{target}, \mathbf{z}_{\mathbf{expr}}^{\mathbf{src}}, z_{noise}^{target})$
2. Adding the expression vectors: $G(z_{id}^{target}, \mathbf{z}_{\mathbf{expr}}^{\mathbf{src}} + \mathbf{z}_{\mathbf{expr}}^{\mathbf{target}}, z_{noise}^{target})$

Results can be seen in Figure 9.

We also show examples of interpolation in either identity or expression space, which can be seen in Figure 10.

Decoupling Evaluation - Implementation Details

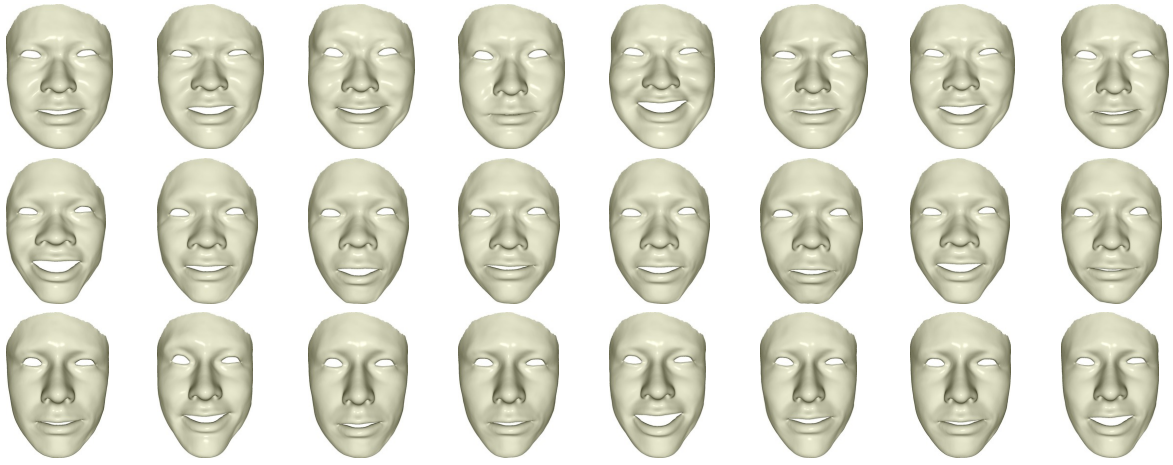
We train the embedding networks using a Resnet-18 architecture with input images of size 224×224. The images contain the orthographic projection of the facial mesh, and the values in the RGB channels encode the normal direction of each vertex, as we found this to give better results than the UV images. The networks were trained using the datasets described

Figure 6: Example of model fitting used in generalization evaluation, for the three compared methods.

in Section 5.2 with the provided labels and the same train/test split. The threshold is selected such that it maximizes the accuracy on the validation set, while keeping the False Acceptance Rate (FAR) below 10%. We build the validation set by randomly choosing an equal number of positive and negative pairs from the testing split. We choose 0.14 as threshold for identity, which achieves 98.66% accuracy and a FAR of 1.21%. For expression we use 0.226 as threshold, which achieves 84.2% of accuracy and a FAR of 8.03%.

(a) COMA

(b) MAE



(c) Us

Figure 7: Example of results used for identity decoupling and diversity evaluation, for the three compared methods. Each row shows samples with a same identity code, while the expression code is drawn randomly.

(a) COMA

(b) MAE

(c) Us

Figure 8: Example of results used for expression decoupling and diversity evaluation, for the three compared methods. Each row shows samples with a same expression code, while the identity code is drawn randomly.

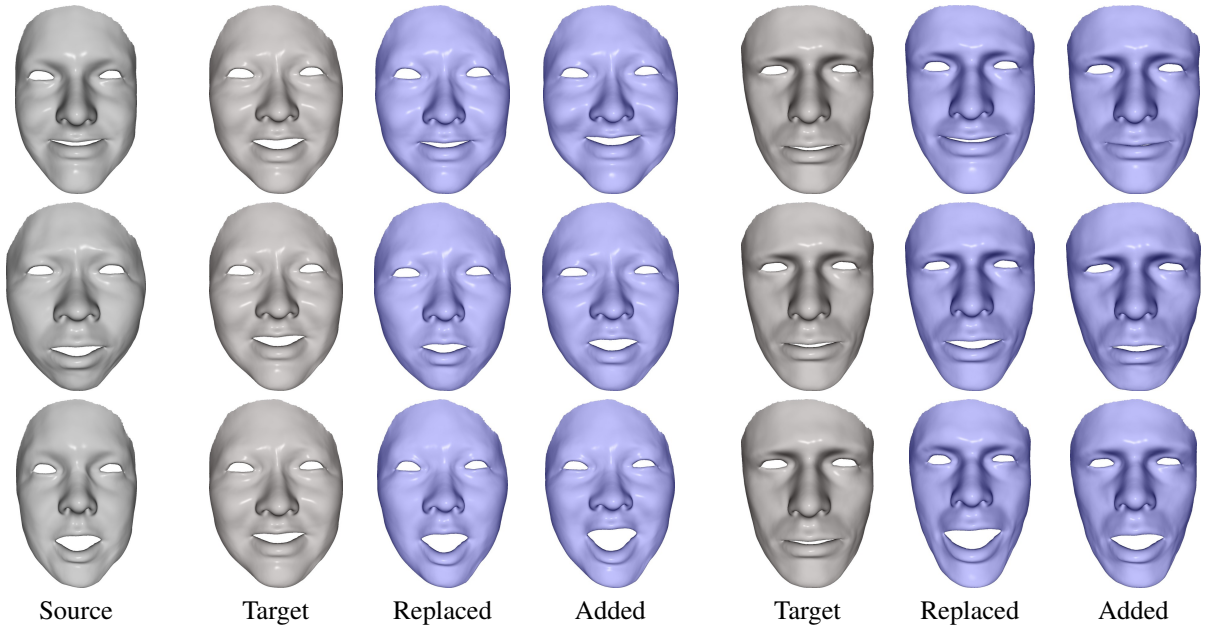
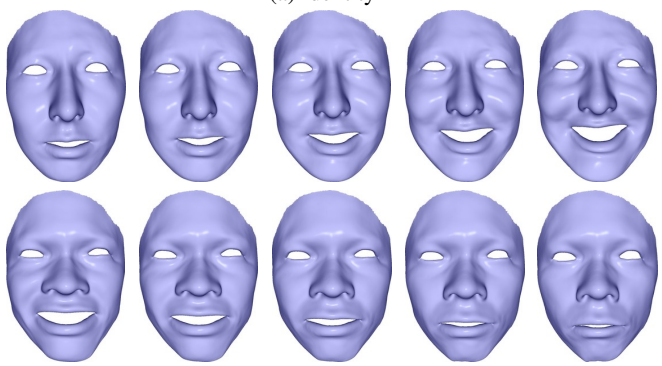


Figure 9: Example of expression space manipulation. In gray a source mesh and a target mesh. In purple the result of (1) replacing the expression code of the target with that of the source *(replaced)*, and (2) adding the source and target expression codes *(added)*.

(a) Identity



(b) Expression

Figure 10: Interpolation in identity and expression space