

Toward Mesh-Invariant 3D Generative Deep Learning with Geometric Measures

Thomas Besnier¹, Sylvain Arguillère²
Emery Pierson^{1, 5}, Mohamed Daoudi^{3, 4}

¹Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRISTAL, Lille,
F-59000, France

²Univ. Lille, CNRS, UMR 8524 - Laboratoire Paul Painlevé,
F-59000 Lille, France

³Univ. Lille, CNRS, Centrale Lille, Institut Mines-Télécom,
UMR 9189 CRISTAL, Lille, F-59000, France

⁴IMT Nord Europe, Institut Mines-Télécom, Univ. Lille,
Centre for Digital Systems, Lille, F-59000, France

⁵Universität Wien, Austria

June 29, 2023

Abstract

3D generative modeling is accelerating as the technology allowing the capture of geometric data is developing. However, the acquired data is often inconsistent, resulting in unregistered meshes or point clouds. Many generative learning algorithms require correspondence between each point when comparing the predicted shape and the target shape. We propose an architecture able to cope with different parameterizations, even during the training phase. In particular, our loss function is built upon a kernel-based metric over a representation of meshes using geometric measures such as currents and varifolds. The latter allows to implement an efficient dissimilarity measure with many desirable properties such as robustness to resampling of the

mesh or point cloud. We demonstrate the efficiency and resilience of our model with a generative learning task of human faces.

1 Introduction

In this paper, we focus on the generation of believable deformations of 3D faces, which has practical applications in various graphics fields, including 3D face design, augmented and virtual reality, as well as computer games and animated films. Despite the rapid progress in 3D face generation thanks to deep learning, existing methods have not yet been able to learn from unregistered scans with varying parameterizations (see Figure 1). Indeed, one major restriction of the current methods such as graph convolutional networks [3, 44] for generating these facial deformations is their reliance on a unified graph structure (required by the network architecture), along with point correspondence (for the loss function) between the target and the predicted mesh. This is especially problematic when handling real-world data: surfaces can be acquired with different technologies (LIDAR, 3D scans, Neural radiance fields [34], ...) which results in inconsistent and heterogeneous databases [56, 48, 24].

Therefore, costly registration algorithms are often needed to create consistent datasets, and can even require manual intervention. As the registration of large point clouds or meshes can require hours or even days of processing time, especially with the growing number of available databases, this has motivated ongoing research on efficient methods and hardware acceleration techniques to make them more practical for real-world applications.

To address these issues, we propose an auto-encoder architecture that can, by design, be trained on inconsistent datasets with no unifying graph structure, resolution or point correspondence. Moreover, our approach works directly with explicit surface meshes, avoiding complex computation with volumetric data or functional description of the shapes. The main tools employed in our approach consist of a PointNet [6] encoder with the ability to process point clouds with variable number of points and map them to a low dimensional latent space, alongside a novel loss function that can withstand variations in parameterization. In particular, the proposed loss is robust to resampling of the mesh thanks to a representation of shapes in terms of geometric measures such as varifolds. Recently, by using a kernel metric on the space of these varifolds, Kaltenmark *et al.* [28] have proposed a general



Figure 1: Several reparameterized meshes from the COMA dataset [44]. Captured geometric data does not have identical graph structure and correspondence between points. Even the resolution driven by the number of points/vertices is subject to variations along a dataset of scans.

framework for 2D and 3D shape similarity measures, invariant to parameterization and equivariant to rigid transformations. This framework showed state-of-the-art performances in different shape matching frameworks, which motivates our approach. To the best our knowledge, this is the first use of kernel metric in the space of geometric measures as cost function in deep learning.

Our results show that the presented model is indeed able to learn on meshes with different parameterization. Moreover, our learned auto-encoder demonstrates expressive capabilities to rapidly perform interpolations, extrapolations and expression transfer through the latent space. Our main contributions are as follow:

- We propose a generative learning method using a parameterization invariant metric based on geometric measure theory. More precisely, we use kernel metrics on varifolds, with a novel multi-resolution kernel. We use it as a dissimilarity measure between the generated mesh and the target mesh during training. We highlight in particular many desirable properties of this metric compared to other metrics used in unsupervised geometric deep learning.

- We propose a robust training method for face registration. It is composed of an asymmetric auto-encoding architecture, that allows to learn efficiently on human faces, with a loss function based on a varifold multi-resolution metric. This approach allows us to learn on inconsistent databases with no correspondence between vertices. Moreover, we are able to learn efficiently an expressive latent space.
- To validate the robustness of our approach, we conduct several experiments, using the trained model, including face generation, interpolation, extrapolation and expression transfer.

2 Related work

2.1 Geometric deep learning on meshes

Within the field of geometric deep learning, a critical challenge is generalizing operators such as convolution and pooling to meshes. The first idea was a direct application of convolutions on the well-suited transformation of 3D meshes, such as multi-view images [41], or 3D (Convolutional neural networks) CNNs on volumetric data [21]. Recent advances have shown that the latter, with other representations such as signed distance or radiance fields can be used in 3D CNNs and allow the reconstruction of 3D shapes. It was successfully used in applications such as geometric aware image generation [57], shape reconstruction from images [5], or partial shape completion [10]. However, these approaches remain expensive and time-consuming to obtain detailed shapes, making them unpractical for generative tasks compared to mesh-based approaches [17].

By proposing a permutation invariant approach to learning on point clouds, PointNet [6] has opened a new and practical way to easily encode information of 3D points. Several improvements have been applied over the years with multi-scale aggregation techniques, such as PointNet++ [42], KPConv [52], or the PointTransformer [58], and recently, PointNext [43] has shown that such an approach can scale on large databases. However, while PointNet showed provable robustness to different discretization, PointNet++, and its derivatives are based on fixed-size neighborhood aggregation, thus being sensitive to the mesh discretization.

In the meantime, surface-based filters have been proposed to improve results and take in account the geometry of the surface. The first filters pro-

pose to exploit the graph structure of surface meshes [33], and apply Graph Neural Networks (GNN) on the underlying graph. The advantage is that it becomes easy to generalize popular CNNs architecture like autoencoders or U-Net. However, original GNN lacks expressivity because of their isotropy (they gather information in all directions equally), and several anisotropic filters have been proposed [16, 35, 50] and shown state-of-the-art results on shape classification and segmentation. Whether or not they can be applied to generative models remains however an open question. Recently, Lemeunier et al. [30] proposed to work in the spectral domain to learn on human bodies, but this approach needs the meshes to be converted to the spectral domain during learning and it is unclear how to adapt it to unparameterized data. Neural3DMM [3], which is based on the spiral ordering of mesh neighborhoods, has shown its efficiency on 3D faces [1, 38].

In this work, we follow recent advances [1] and we propose an asymmetric autoencoder, with a PointNet architecture for computing the latent vector from an unregistered mesh: the simplicity of PointNet combined to its proven robustness makes it the ideal candidate, as opposed to more expressive, but less parameterization robust approaches. In the contrary, the decoder is made of a template-dependent architecture, namely SpiralNet, for two reasons: the proven results on human faces, and the fact that spiral convolutions incorporate a better prior on the deformation model. We illustrate this with a very simple experiment in which a model learn a Multi-Layer Perceptron (MLP) and an SpiralNet decoder on the task of mapping a single vector to a target face. We observe in Figure 2 that the target shape is reached faster, and the intermediate shape are representing human faces, as opposed to the MLP decoder. This property will allow our model to reach more easily a suitable registration of shapes.

2.2 Robust generative learning in 3D

In the literature, "robustness" is often a welcomed side property of the model [25, 50, 6], but these models still rely on a consistent database for the training phase.

In 3D machine learning, while supervised learning relies on training data that has a precise and consistent order, unsupervised learning operates without any prior correspondence and instead utilizes self-organization to model the inherent geometry of the data. Here, we explore tools to allow **complete unsupervised** 3D generative deep learning. Little work has been done on

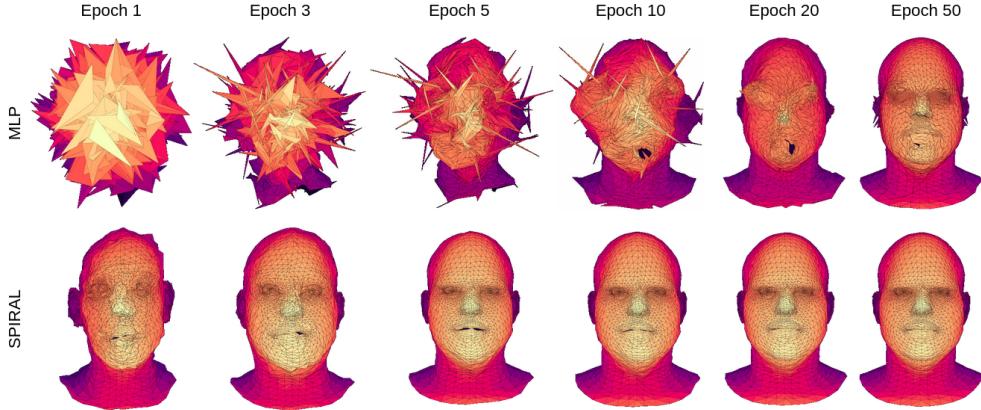


Figure 2: Visualization of the learning process of a single mesh using our model with different decoders. The first row shows the results using a MLP as the decoder and the second row, we used a spiral convolution. In addition to a slower learning process, the MLP struggles to map a linear space (the latent space) to a high-dimensional and non-linear one.

this subject, especially for generative tasks, as in [23] or [13]. Most of the time, unsupervised learning tasks are performed by considering a functional representation of shapes [19, 46, 4] but these methods have not been applied for face generation yet. We also mention [32, 1] who proposed an hybrid supervised/unsupervised learning protocols. Most of the other unsupervised registration methods are computed with non learning method such as LD-DMM (Large deformation diffeomorphic metric mapping) [2] or elastic shape matching [26] with a recent exception in [13] where the authors learn an unsupervised diffeomorphic registration with an auto-encoder architecture that uses optimal transport for the loss function.

2.3 Geometric loss functions

In the following, we consider 3D datasets with elements being meshes. We describe such a mesh $X = \{V(X), E(X), F(X)\}$ with $V(X) = (x_i)_i \in \mathbb{R}^{3 \times n_X}$ the set of vertices of X , $E(X)$ its edges and $F(X)$ its faces with $n_X = |V(X)|$ being the resolution of the mesh. Following this, \hat{X} will refer to a reconstruction of X .

As mentioned in the introduction, our goal is to learn explicit 3D data on inconsistent databases. In real-world situations, without prior registration,

we usually lack the correspondence between points, making it difficult to compute the mean squared error (MSE), or any similar euclidean distance. As a result, we investigate dissimilarity metrics that are robust or invariant to the parameterisation with a relevant geometric meaning. We review some conventional distances, some of them will be used to evaluate our method:

- The Mean-squared error metric requires correspondence between each points

$$\mathcal{L}^{MSE}(X, \hat{X}) = \frac{1}{n_X} \sum_{x \in X} \|x - \hat{x}\|_2^2 \quad (1)$$

It is the most commonly used dissimilarity measure for supervised learning tasks.

- The Hausdorff distance (H) is a strong metric with a powerful ability to generalize to heterogeneous spaces [45].

$$\mathcal{L}^H(X, \hat{X}) = \max \left\{ \sup_{x \in X} d(x, \hat{X}), \sup_{\hat{x} \in \hat{X}} d(X, \hat{x}) \right\} \quad (2)$$

where d is a pre-defined distance, usually euclidean.

Unfortunately, optimizing with respect to this metric will imply to correct only one point at each step of the gradient which make it an inefficient loss function.

- The Chamfer distance (CD) [55] is strongly linked with the iterative closest point algorithm (ICP) as it is basically the objective function to minimize for this algorithm:

$$\begin{aligned} \mathcal{L}^{CD}(X, \hat{X}) &= \frac{1}{n_X} \sum_{\hat{x} \in \hat{X}} \min_{x \in V(X)} \|\hat{x} - x\|_2^2 \\ &\quad + \frac{1}{n_{\hat{X}}} \sum_{x \in V(X)} \min_{\hat{x} \in V(\hat{X})} \|\hat{x} - x\|_2^2 \end{aligned} \quad (3)$$

In fact, as a loss function we only require the directed Chamfer distance (DCD) given by the first term in the previous expression

$$\mathcal{L}^{DCD}(X, \hat{X}) = \frac{1}{n_X} \sum_{\hat{x} \in \hat{X}} \min_{x \in V(X)} \|\hat{x} - x\|_2^2 \quad (4)$$

This loss is notably used for existing unsupervised learning tasks in [23, 32, 1, 9]. However, the Chamfer distance can suffer from poor performances for at least two reasons: the first one is that the use of min operator makes the loss unstable, because it is not fully differentiable with respect to a mesh positions. Second, it can be sensitive to outliers or collapse on points of a mesh. Generally, this loss is regularized using an additional term constraining the mesh deformation during the training phase. Various techniques exists, such as adding an edge loss \mathcal{L}^{edges} with respect to a template mesh X^t . It can also be combined with a Laplacian loss to ensure the smoothness of the reconstructed mesh.

- The Wasserstein distance (also called the *earth-mover distance*) from optimal transport theory is very popular to compute a distance between shapes. However, solving the optimal transport problem is NP-hard [39], and the distance can be approximated with the debiased Sinkhorn divergence (SD) developed in [14] and recently used in [20]. To compute this distance, we represent a mesh X as an aggregation of Dirac's measure $A(X) = \sum_i a_i^X \delta_{c_i^X}$ which gives a sum representing the center of faces $(c_i^X)_i$, weighted by their corresponding area $(a_i^X)_i$.

$$\begin{aligned} \mathcal{L}^{SD}(X, \hat{X}) &= OT_\epsilon(A(X), A(\hat{X})) - \frac{1}{2}OT_\epsilon(A(X), A(X)) \\ &\quad - \frac{1}{2}OT_\epsilon(A(\hat{X}), A(\hat{X})) \end{aligned} \quad (5)$$

with

$$OT_\epsilon(\alpha, \beta) = \min_{\pi \in \Pi} \sum_{i=1}^N \sum_{j=1}^M \pi_{i,j} \frac{1}{p} \|c_i^X - c_j^{\hat{X}}\|_p + \epsilon KL(\pi \|\alpha \otimes \beta)$$

where π is the regularized transport plan. The Kullback-Leibler divergence (KL) is a regularization term, called entropic penalty, and the blur ϵ is a hyperparameter that indicates how strong is the approximation of the Wasserstein distance with $OT_\epsilon \xrightarrow{\epsilon \rightarrow 0} OT$ the true Wasserstein distance. This hyperparameter needs to be adapted to each different task. Moreover, we observed in our experiments that the computation and differentiation of the distance can still be expensive, and makes it unpractical for learning on large datasets.

In contrast to all methods, the varifold approach is built using reproducing kernels, and thus have a spatial support: the outliers are not seen by the loss. Moreover, by taking into account every relationship in each pair of points, it is fully differentiable with respect to mesh position. Finally, the use of normals in the loss, allows to account for the shape of the surface, instead of seeing a mesh as just a cloud of points of \mathbb{R}^3 . We denote our proposed loss \mathcal{L}^{GM} and summarize its advantages in Table 1.

Property / Loss	\mathcal{L}^{MSE}	\mathcal{L}^H	\mathcal{L}^{DCD}	\mathcal{L}^{SD}	\mathcal{L}^{GM}
Unsupervised	✗	✓	✓	✓	✓
Smooth gradient	✓	✗	✗	✓	✓
Position	✓	✓	✓	✓	✓
Orientation	✗	✗	✗	✗	✓
Tunable	✗	✗	✗	✓	✓

Table 1: Summary of the properties of the aforementioned dissimilarity metrics

As we will see of the experiments, we do not need any regularization to obtain plausible mesh as outputs of our method.

3 Our approach

To complete the face generation task, we propose a simple auto-encoder detailed in Figure 3. The main originality of our model comes from the loss function based on a representation of meshes with discrete geometric measures.

3.1 Geometric measure theory applied to surfaces

Let S be a parameterized surface embedded in \mathbb{R}^3 . This surface and its triangulations can be understood as points in a high-dimensional space of shapes (infinite in the case of continuous surfaces). We look for a robust metric on this particular space, suitable for unsupervised learning tasks. If we take a diffeomorphism ϕ acting on the parameter space, such a metric should not differentiate between a given shape S and a reparameterization $\tilde{S} = S \circ \phi$ of this shape: S and \tilde{S} are at distance 0 for this metric.

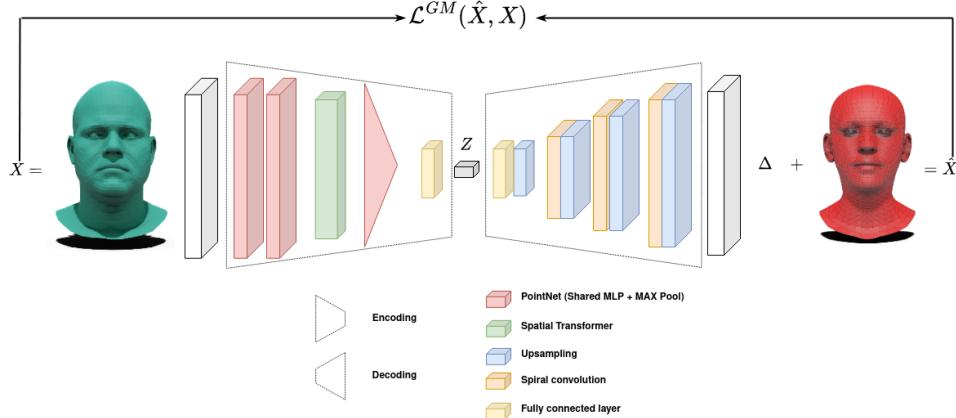


Figure 3: Architecture overview. The encoder takes a mesh X of any parameterization as input and outputs a deformation Δ added to a chosen template to obtain a registered mesh \hat{X} with a similar topology as the template.

Definition 3.1 (Varifold representation of surfaces) *The varifold μ_S associated to a continuous surface shape S is the measure on $\mathbb{R}^3 \times \mathbb{S}^2$ such that, for any continuous test function $u : \mathbb{R}^3 \times \mathbb{S}^2 \mapsto \mathbb{R}$:*

$$\mu_S(u) = \int_{\mathbb{R}^3 \times \mathbb{S}^2} u d\mu_S = \int_S u(x, \vec{n}_x) d\sigma(x) \quad (6)$$

where \vec{n}_x is the normal of S at x and $d\sigma$ the area measure of the surface S .

The key property that motivates the use of varifolds in our context is that for any two parameterized shapes S and \tilde{S} , $\mu_S = \mu_{\tilde{S}}$ if and only if \tilde{S} is a reparameterization of S [8].

Moreover, there is a natural discrete version of varifolds for meshes as follows. If f is a triangle (e.g. a face in a triangular mesh) with center $c(f)$, and normal \vec{n}_f , the corresponding discrete varifold μ_f is given by a Dirac mass $\delta_{c(f)}^{\vec{n}_f}$ at $(c(f), \vec{n}_f)$ weighted by $a(f)$ the area of f . In other words, for any continuous test function u on $\mathbb{R}^3 \times \mathbb{S}^2$,

$$\mu_f(u) := a(f)u(c(f), \vec{n}_f).$$

and we can write $\mu_f = a(f)\delta_{c(f)}^{\vec{n}_f}$. Therefore, we can extend this to a triangular mesh:

Definition 3.2 (Discrete varifold representation of surfaces) Let a mesh $X = \{V(X), E(X), F(X)\}$, where $V(X)$, $E(X)$, and $F(X)$ are respectively, the set of vertices, edges, and faces. The varifold representation associated to X is the measure on $\mathbb{R}^3 \times \mathbb{S}^2$, given by

$$\mu_X := \sum_{f \in F(X)} \mu_f = \sum_{f \in F(X)} a(f) \delta_{c(f)}^{\vec{n}_f},$$

with $\mu_f = a(f) \delta_{c(f)}^{\vec{n}_f}$ as described above.

This representation is well suited for triangular meshes as each triangle is represented by a measure on the position of its center $c(f)$ and its orientation \vec{n}_f given by a point on the 2-sphere, all weighted by the area of the triangle $a(f)$.

Moreover, we have the following result, which is an easy consequence of Proposition 1 from [28] combined with Corollary 1 from [37]:

Theorem 3.1 Take $u : \mathbb{R}^3 \times \mathbb{S}^2 \rightarrow \mathbb{R}$ a bounded k_u -Lipschitz function, with supremum $\|u\|_\infty$. Let S a be surface, and X a triangular mesh drawn from S whose vertices belongs to S , with greatest edge length η_X , and smallest angle θ_X among its faces. Denote κ_S the greatest principal curvature over S , and $a(S)$ its surface area. Then, there is a universal constant C such that

$$|\mu_S(u) - \mu_X(u)| \leq C a(S) \frac{(\kappa_S + 1)}{\sin \theta_X} (k_u + \|u\|_\infty) \eta_X$$

Consequently, for any two good triangulations with relatively small edges X and \hat{X} of S , $\mu_X \simeq \mu_{\hat{X}}$, making discrete varifolds natural tools for mesh-invariant purposes.

Comparing shapes with kernel metrics.

With this representation of shapes, we compute dissimilarities between shapes, both continuous and discrete, by using kernels on $\mathbb{R}^3 \times \mathbb{S}^2$. Following the works of [53, 8, 28, 40], we use a product $k = k_p k_n$, with k_p a kernel on \mathbb{R}^3 and k_n a kernel on \mathbb{S}^2 .

To ensure invariance under the action of rigid motions (rotations and translations), we choose a radial basis function ρ to drive the position kernel k_p and a zonal kernel γ for the orientation kernel k_n . Details on admissible functions for γ and ρ can be found in [22, 49].

$$k_p : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^3 & \rightarrow \mathbb{R} \\ (x, \hat{x}) & \mapsto \rho(|x - \hat{x}|) \end{cases} \quad (7)$$

$$k_n : \begin{cases} \mathbb{S}^2 \times \mathbb{S}^2 & \rightarrow \mathbb{R} \\ (\vec{n}_x, \vec{n}_{\hat{x}}) & \mapsto \gamma(\langle \vec{n}_x, \vec{n}_{\hat{x}} \rangle) \end{cases} \quad (8)$$

These kernels are extrinsic in the sense that they are defined on the ambient space $\mathbb{R}^3 \times \mathbb{S}^2$, and use the euclidean distances.

Then, we can derive a correlation between any two measures $\mu, \hat{\mu}$ on $\mathbb{R}^3 \times \mathbb{S}^2$ as

$$\langle \mu, \hat{\mu} \rangle_k = \int_{\mathbb{R}^3 \times \mathbb{S}^2} \int_{\mathbb{R}^3 \times \mathbb{S}^2} k_p(x, \hat{x}) k_n(\vec{n}_x, \vec{n}_{\hat{x}}) d\mu(x, \vec{n}) d\hat{\mu}(\hat{x}, \vec{n}_{\hat{x}})$$

This gives a parameterization-independent correlation between two surfaces S and \hat{S} through the kernel k as follows:

$$\langle \mu_S, \mu_{\hat{S}} \rangle_k = \iint_{S \times \hat{S}} k_p(x, \hat{x}) k_n(\vec{n}_x, \vec{n}_{\hat{x}}) d\sigma(\hat{x}) d\sigma(x) \quad (9)$$

For the discrete setting, we write the correlation between two faces f and \hat{f}

$$\langle f, \hat{f} \rangle_k = a(f)a(\hat{f})k_n(\vec{n}_f, \vec{n}_{\hat{f}})k_p(c(f), c(\hat{f})). \quad (10)$$

This can be summed along the meshes to give a discretized version of Equation (9). For X and \hat{X} two meshes, the correlation is

$$\langle \mu_X, \mu_{\hat{X}} \rangle_k = \sum_{f \in F(X)} \sum_{\hat{f} \in F(\hat{X})} a(f)a(\hat{f})k_p(c(f), c(\hat{f}))k_n(\vec{n}_f, \vec{n}_{\hat{f}}) \quad (11)$$

Now for some kernels, these formulas actually give a positive definite dot-product on the space of measures, so that $\mu = \hat{\mu}$ if and only if $\|\mu - \hat{\mu}\|^2 = \langle \mu - \hat{\mu}, \mu - \hat{\mu} \rangle = 0$. From there, we define the "geometric measure" (GM) loss associated to such a kernel by

$$\begin{aligned} \mathcal{L}_k^{GM}(X, \hat{X}) &= \langle \mu_X - \mu_{\hat{X}}, \mu_X - \mu_{\hat{X}} \rangle_k \\ &= \langle \mu_X, \mu_X \rangle_k + \langle \mu_{\hat{X}}, \mu_{\hat{X}} \rangle_k - 2\langle \mu_X, \mu_{\hat{X}} \rangle_k. \end{aligned} \quad (12)$$

For well-chosen kernels, this function is fully differentiable and can be written in closed form, making this loss function suitable for GPU accelerated computations. Moreover, thanks to Theorem 3.1, this GM loss is robust to a mesh change in both X and \hat{X} .

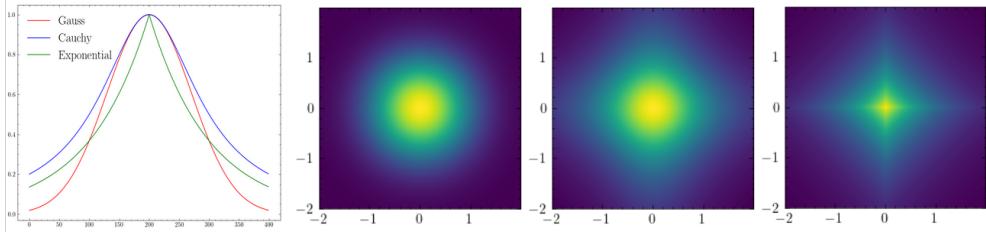


Figure 4: Visualization of 3 kernels suitable for k_p , namely a Gaussian kernel of the form $x \mapsto \exp\left(\frac{x^2}{\sigma^2}\right)$, a Cauchy kernel of the form $x \mapsto \frac{1}{1+(\frac{x}{\sigma})^2}$ and an exponential kernel $x \mapsto \exp\left(\frac{|x|}{\sigma}\right)$. On the right is displayed the 1D plot of the three kernels: in red the Gaussian kernel, in blue the Cauchy kernel and in green the exponential one. Next, 2D plots of the Gaussian, Cauchy and exponential kernel respectively from left to right.

Choice of kernel and loss function. Several kernels are suitable for k_p such as Gaussian, linear and Cauchy kernels. We display some of them in Figure 4.

By default, we use a Gaussian kernel for the position and, depending on the type of geometric measure, we either use a linear (current), squared (varifold) or exponential (oriented varifold) zonal kernel on \mathbb{S}^2 for k_n . In particular, k_p is defined by a scale parameter σ . Typically, this parameter should be chosen in order to encompass the structure of local neighborhoods across the meshes.

To improve the versatility and efficiency of our loss, we propose to use a sum of kernels with different scale parameter for each term, such that, our final loss is defined as

$$\mathcal{L}^{GM} = \sum_i \lambda_i \mathcal{L}_{k_i}^{GM} \quad (13)$$

with k_i associated with the scale σ_i and a scalar weighting coefficient λ_i . The number of kernels and the coefficients $(\lambda_i)_i$ are hyperparameters of the model. For our task, we observed experimentally that setting $\lambda_i = \left(\frac{\sigma_i}{\max_i \sigma_i}\right)^2$ could give good enough results. This way, we penalize small scales but still allow the metric to distinguish fine structures on the mesh made up of small triangles.

3.2 Application: face generation

Human face modeling involves deformable geometries with small but meaningful changes. In general, we model a face shape S as the combination of an identity (shape of the face) and an expression (reversible deformation from a neutral expression): this is the so-called **morphable model** [18], [15]. It has seen many improvements over the past few years thanks, in part, to computer vision research and deep learning. Indeed, using nonlinear, deep representations presents the potential to outperform traditional linear or multi-linear models in terms of generalization, compactness, and specificity. In addition, the implementation of deep networks for parameter estimation allows for quick and dependable performance even with uncontrolled data.

$$S = \text{Id} + \text{Expr} = \text{Template} + \Delta\text{Id} + \Delta\text{Expr} \quad (14)$$

and we set $\Delta = \Delta\text{Id} + \Delta\text{Expr}$ the total deformation of the template to match the target face.

The COMA dataset [44] encapsulates this representation as it is made of 12 identities executing 12 different expressions. Each expression is a sequence of meshes during which the subject starts from a neutral face, executes the expression and goes back to a neutral face. Each sequence is made from 25 to more than 200 meshes and the sequence length is not consistent across the identities. A distinction is made between the unregistered meshes obtained from scans and their registered counterpart providing us with two distinct databases.

4 Experiments

The model corresponding to the architecture presented in Figure 3 is trained end-to-end, both the encoder and the decoder weights are optimized at the same time. The Python code is built over the one from [3] using the Pytorch framework. All measurements were conducted using the same machine (a laptop) with an NVIDIA Corporation / Mesa Intel® UHD Graphics (TGL GT1) as GPU and Intel® Core i5-9300HF 2.40GHz CPU with 8,00 Go RAM.

4.1 Implementation details

Our model takes as input a mesh of any parameterization and gives out a mesh in the COMA topology which has 5023 vertices and 9976 faces with

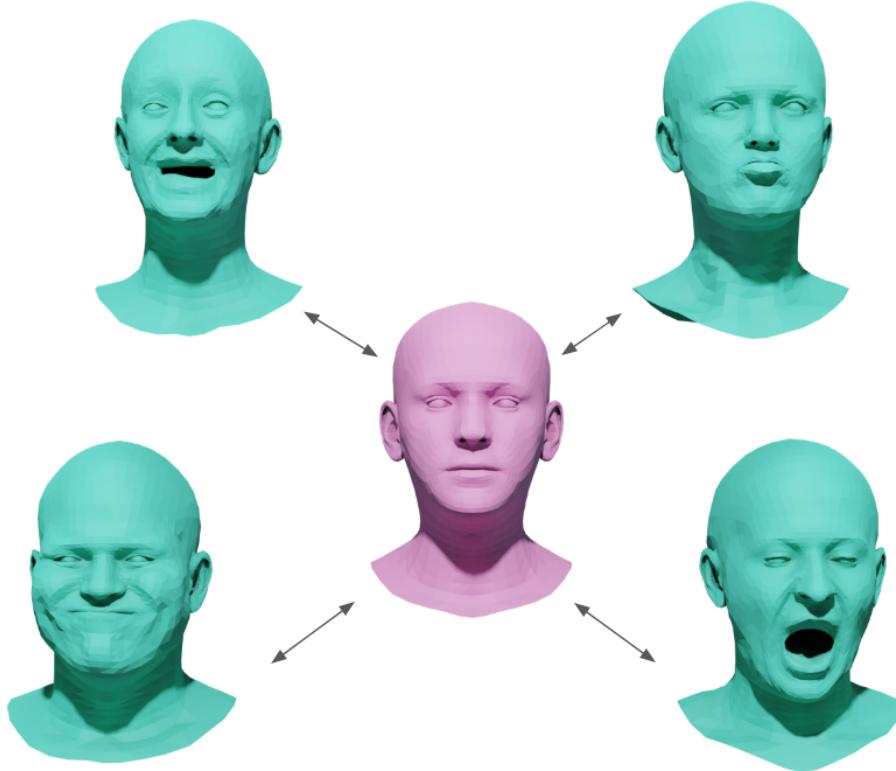


Figure 5: Illustration of a morphable model: a template mesh (in red) is deformed into a given mesh (in blue).

a fixed graph structure (the topology of the output solely depends on the topology of the chosen template which can be adapted).

The encoder is a combination of a simple PointNet architecture (PN), a spatial transformer (TF) as described in [27] to improve invariance to euclidean transformations of the input and a fully connected layer (FC). We use the parameters of [12], that were optimized on the COMA dataset. The filter sizes for the decoder are [128, 64, 64, 64, 3]. It starts with a fully connected layer (FC) and then alternately performs up-sampling (US) and spiral (de)-convolution (SC). The parameters are taken from the original Neural3DMM [3] paper.

- *Encoder:* PN(64,1024) → TF(64) → FC(128,64,64)

- *Decoder:* $\text{FC}(128) \rightarrow \text{US}(4) \rightarrow \text{SC}(128) \rightarrow \text{US}(4) \rightarrow \text{SC}(64) \rightarrow \text{US}(4) \rightarrow \text{SC}(64) \rightarrow \text{US}(4) \rightarrow \text{SC}(64)$

The model learns to code the deformation from a fixed template mesh X^t that does not belong to the training dataset. It is trained for 100 epochs with a latent space of fixed size 128. We used Adam optimizer with a learning rate of 10^{-3} and a batch size equal to 16.

Unfortunately, this loss alone is optimal when the size of the triangles is relatively constant along the mesh. In order to cope with this limitation, we use what we call a **multi-kernel metric** with a collection of different scale (σ_i)_i. Experimentally, we observe that the loss function induces poor performances when we do not use enough kernels but also when we use too much of them (see Figure 6).



Figure 6: Qualitative comparison for the reconstruction of a target mesh (on the left) with a growing number of kernel (summed) as we go on the right.

Even with a carefully optimized loss function, we still observe some noise in the reconstruction. To correct this, we propose to use a post-processing step with a Taubin smoothing [51]. We highlight the improvement in quality of the reconstructed mesh in Figure 7. More sophisticated techniques can be applied such as using the pretrained model from Kim *et al.*[29] to remove the noise.

4.2 Learning performances

We train our model on 11 out of the 12 identities of the COMA dataset and we evaluate the performance on the remaining identity. Therefore, we

assess the generalizability of the model and we compare the performances to other unsupervised methods. FLAME [31] is a face morphable model. In the paper, the authors use this model to register faces easily. They minimize a regularized Chamfer distance and use landmark information to register a face model. To be fair to our method, we do not use landmark information and minimize the mesh-to-mesh distance. 3DCODED [23] is a deep learning model using a Pointnet encoder and a MLP decoder that deforms a template mesh. Their unsupervised version uses Chamfer distance as a loss function, and is regularized using an edge loss and a Laplacian loss. This model has demonstrated its efficiency for human body registration, and we modify it in order to apply it on human faces. We also tried a more recent model: Deep Diffeomorphic Registration [13] (DDFR). It is a deep learning model that learns a diffeomorphism of the ambient space in order to morph a source mesh (the template) to a target mesh. Unfortunately, the training process has shown overflowing computational cost while showing limiting ability to produce expressive faces. This can be explained as diffeomorphisms of the ambient space can hardly separate the lips and produce "sliding motions" which are not differentiable.

We compute the reconstruction error according to 3 metrics: the Hausdorff distance d_H , the Chamfer distance d_{CD} and a Varifold distance d_V with k_p being a Gaussian kernel with $\sigma = 0.1$. This specific scale has been chosen as it is roughly ten times the average size of a triangle. The results are presented in Table 2.

Model	d_H	$d_{CD} (\times 10^{-4})$	$d_V (\times 10^{-4})$
3DCODED [23]	0.018	0.312	0.267
FLAME [31]	0.012	0.109	0.013
Ours	0.010	0.091	0.011
Ours (+ filter)	0.009	0.088	0.011

Table 2: Reconstruction error when learning faces from 11 out of the 12 identities of COMA and tested on the remaining identity which constitutes around 1200 meshes for the test set. The error is averaged along this test set.

We highlight the following observations:

1. The FLAME model struggles to reconstruct "extreme" facial expressions such as a wide opened mouth in the 4th row of figure Figure 7.

Loss function	Hausdorff				Chamfer ($\times 10^{-4}$)				Varifold ($\times 10^{-4}$)			
	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4
\mathcal{L}^{DCD}	0.031	0.031	0.030	0.031	0.45	0.44	0.42	0.44	0.38	0.38	0.38	0.38
$\mathcal{L}^{DCD} + \mathcal{L}^{edges}$	0.030	0.029	0.030	0.029	0.43	0.42	0.43	0.45	0.35	0.34	0.36	0.34
$\mathcal{L}^{SD}(\epsilon = 0.0001)$	0.018	0.017	0.017	0.017	0.101	0.099	0.094	0.102	0.016	0.017	0.015	0.016
$\mathcal{L}^{SD}(\epsilon = 0.001)$	0.024	0.027	0.023	0.023	0.135	0.133	0.120	0.134	0.023	0.023	0.021	0.022
$\mathcal{L}^{SD}(\epsilon = 0.01)$	0.036	0.041	0.030	0.031	0.218	0.198	0.152	0.155	0.030	0.035	0.032	0.034
\mathcal{L}^{GM} (ours)	0.010	0.011	0.09	0.010	0.089	0.084	0.086	0.085	0.011	0.011	0.010	0.011

Table 3: Ablation study: We trained our model using different unsupervised losses (only the loss is changed) and we report the mean reconstruction error when learning expressions in the COMA dataset. The reported evaluation is performed on meshes that does not belongs to the training data. We account for the quality of the learning process in relation to the loss function employed for the task.

We also report a longer time for the registration.

2. 3DCODED, while being a lot faster than linear methods such as FLAME, shows poor performance for learning human faces. In fact, the model only output indistinguishable deformations from the template mesh.
3. Our model surpass FLAME in term of expressivity and 3DCODED in terms of efficiency.

4.3 Ablation study

Next, we show that the effectiveness of our model stems in particular from the loss function. We compare some unsupervised losses mentioned in section 3 with our geometric measure based loss \mathcal{L}^{GM} without changing any other parameters. We focus on the encoding and decoding of 4 sequences of expression (#1: *bareteeth*, #2: *cheeks_in*, #3: *high_smile* and #4: *mouth_extreme* in Table 3). Here, \mathcal{L}^{GM} is computed with a Gaussian kernel for k_p and a squared zonal kernel for k_n which correspond to the framework of unoriented varifolds.

We conducted the test using \mathcal{L}^{SD} with different value of approximation ϵ as it is the most challenging candidate to surpass \mathcal{L}^{GM} . In spite of this, setting ϵ to be less than 10^{-4} showed declining performances in addition to much higher computational cost.

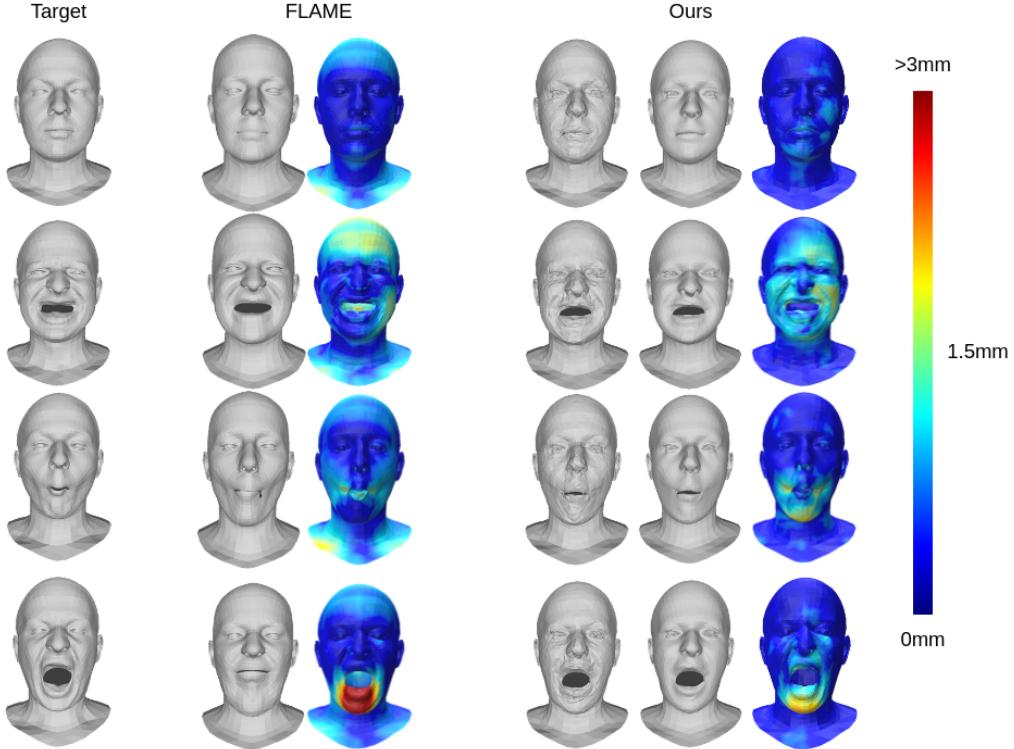


Figure 7: Qualitative results on the reconstruction of facial expressions from the COMA dataset with our AE model. On the left is the target mesh, the first pair of reconstructions shows the reconstructed mesh with a linear method (FLAME) and on the right is displayed the reconstruction with our model. The first proposed reconstruction is obtained without any post-processing and the other one with a Taubin smoothing (parameters: $\lambda = 0.5$, $\mu = -0.53$) with its corresponding MSE heatmap of error.

4.4 Robustness

We evaluate our model on different reparameterised meshes of faces to demonstrate that our model learns the geometry of the shape instead of the graph structure. This experience is conducted on one identity executing all 12 expressions. The proposed reparameterizations of the meshes are displayed in Figure 8.

In a similar fashion as in [50], we test the robustness of our model against three different reparameterizations. *UpDown* is obtained by subdividing the

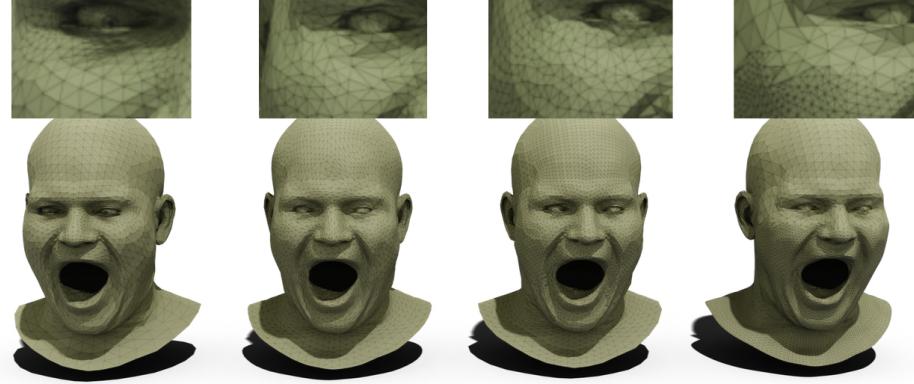


Figure 8: Examples of reparameterized meshes of a single expressive face, from left to right: original parameterization, *UpDown*, *Iso* and *Variable*

mesh and then performing a quadric edge simplification. *Iso* is the result of one iteration of explicit isotropic remeshing. Finally, the *Variable* parameterization is obtained by dividing the mesh in two-half: the top and the bottom part. On the top part, we perform a simplification to diminish the number of triangles and on the bottom part, we subdivide the mesh in order to get a much higher density of triangles. These new meshes are obtained via Meshlab [11] remeshing routines, using the *pymeshlab*¹ Python library.

We stress the robustness of our model by comparing the outputs of the model when given with the original parameterisation as input and the reparameterization as input. The results are summarized in Table 4, where we display the relative difference between the two outputs. While we have in-

	Hausdorff	Chamfer
Original - Original	0	0
Original - UpDown	0.0035	0.0096
Original - Iso	0.0012	0.0016
Original - Variable	0.0018	0.0030

Table 4: Relative difference of reconstruction when the model is tested on reparameterized meshes. The evaluation is performed on meshes that do not belong to the training set in order to evaluate the robustness of the learning process.

¹<https://pymeshlab.readthedocs.io/>

deed a slight difference between the outputs, the worst relative difference is around 2% in Chamfer distance. Qualitative results that we display in Figure 9, highlight this robustness visually.

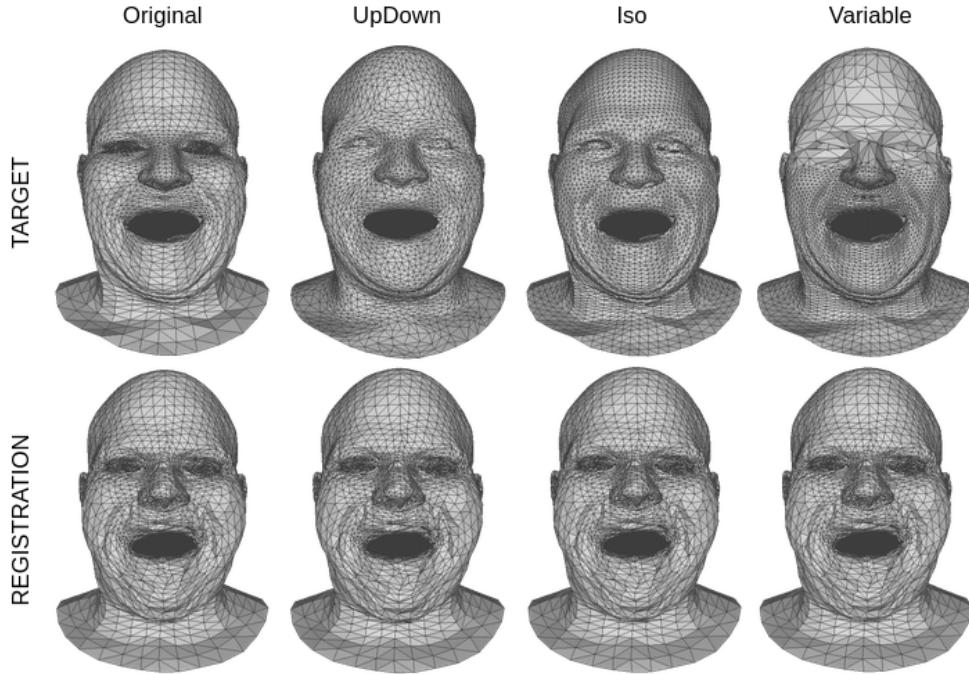


Figure 9: An example of a mesh, its three reparameterizations and the corresponding registration with our model.

4.5 Training on inconsistent batches of meshes

As stated above, our model is able to train on mesh data with variable resolution, hence it can be trained on sampled scans from COMA which makes an inconsistent database. We show that the model is still capable of learning identity and expressions. The only required pre-processing step is a rigid alignment (with scaling) to the chosen template.

As the scans are highly detailed (several gigabytes for each subject), we train our model on one identity with its 12 expressions. We compare the results of our model trained on the registered meshes against another model, with same parameters, trained on the raw scans. We synthesize the results in Table 5 and display a few examples of reconstruction in Figure 10.

Expression	Registered		Raw scans	
	Hausdorff	Chamfer	Hausdorff	Chamfer
Neutral	0.009	0.075	0.014	0.24
bareteeth	0.010	0.075	0.014	0.26
cheeks_in	0.010	0.074	0.014	0.25
eyebrow	0.009	0.076	0.014	0.28
high_smile	0.010	0.076	0.014	0.26
lips_back	0.009	0.073	0.014	0.25
lips_up	0.009	0.075	0.015	0.24
mouth_down	0.009	0.077	0.014	0.25
mouth_extr	0.009	0.075	0.012	0.24
mouth_mid	0.009	0.074	0.013	0.27
mouth_open	0.009	0.074	0.013	0.28
mouth_side	0.009	0.076	0.013	0.27
mouth_up	0.009	0.075	0.014	0.29

Table 5: Reconstruction error for each expression

4.6 Evaluating the latent space

The advantage with our solution is that we can directly operate in the latent space to deform any face. We use this property for three applications, interpolation between faces, extrapolation of a face motion and expression transfer between faces.

Interpolation. We compute a linear interpolation $(z_t)_{t \in [0,1]}$ between a source latent vector z_0 and a target one z_1 with

$$z_t = (1 - t)z_0 + tz_1.$$

We display the resulting interpolation on faces in Figure 11, between two identities, two poses and the interpolation of two faces with both characteristics being different. The figure show that the results are visually satisfying.

Extrapolation. Given a initial motion of a face (two close meshes starting a motion), we would like to extrapolate the full motion. This can be formulated easily in the latent space: from the two meshes' latent codes z_1, z_2 , we shoot a time dependent path z_t from the initial speed $(z_2 - z_1)$:

$$z_t = z_1 + t(z_2 - z_1).$$

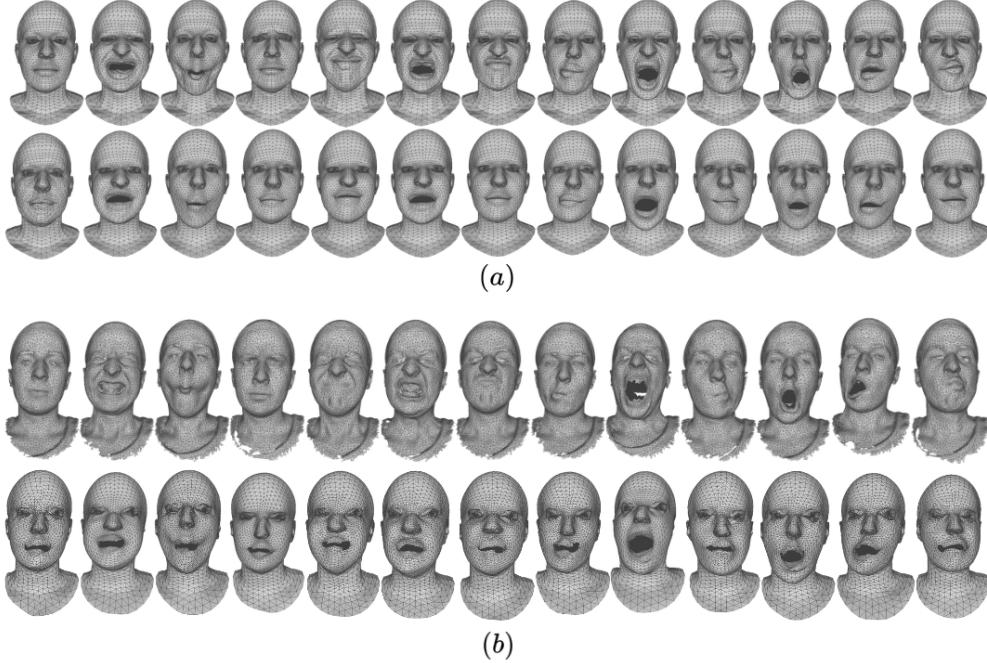


Figure 10: (a): Registered case, fixed resolution, point-wise correspondence and no noise. (b): Raw scans, variable resolution, no correspondence and noise.

We display the resulting motions in Figure 12. We observe that the desired motion is reproduced. At large time steps, some non natural deformations start to appear, but we are still able to recognize the expression of the face.

Expression transfer in the latent space. Thank to the auto-encoding architecture, we also demonstrate its ability to perform complex mesh manipulation such as expression transfer with simple arithmetic operations in the latent space (additions and subtractions). We also demonstrate the robustness of such operations as in Figure 13.

In a similar fashion, manipulating the latent space to cancel an expression and recover the neutral face is possible as shown in Figure 14.

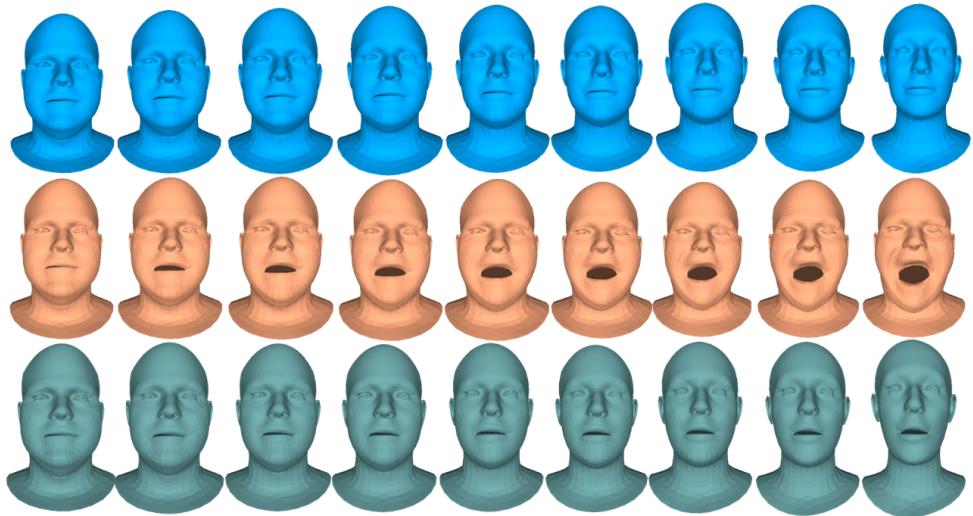


Figure 11: Linear interpolation in the latent space. The first row shows an interpolation between two neutral identities, the second row shows the result of an interpolation between a neutral face and one of its expressions. The last row shows an interpolation between a neutral identity and an expressive other identity.

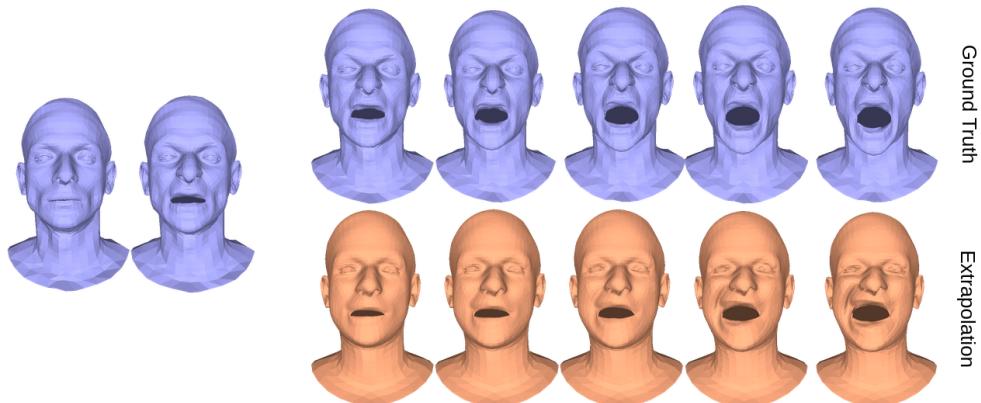


Figure 12: Linear extrapolation (in orange) from two meshes (on the left) of a sequence describing an expression.

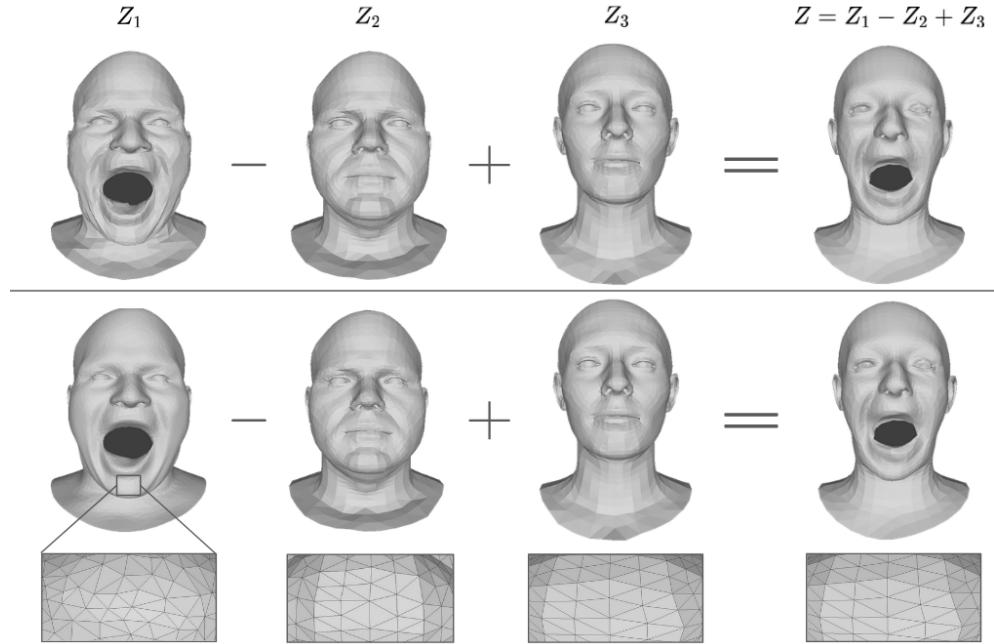


Figure 13: An example of a robust expression transfer: from an expressive face encoded as Z_1 , we subtract its neutral identity encoded as Z_2 and replace it with another one encoded as Z_3 .

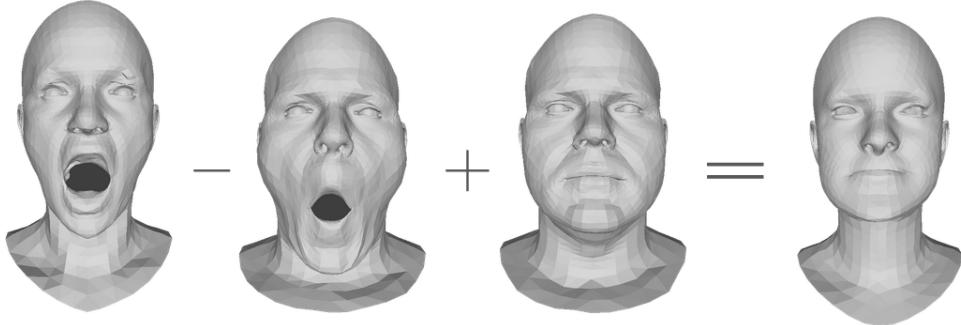


Figure 14: An example of neutralization of an expression to recover a neutral face. From an expressive face on the left, we subtract a similar expression from another identity, which we cancel with its corresponding neutral face to obtain an estimation of the neutral face of the first identity (on the right).

5 Discussion

In this last section, we discuss some areas for improvement and implications of the presented work.

5.1 Complexity

Regarding the complexity of our model, it is made of simple elements and most of the computation complexity comes from the calculation of the loss at each epoch, during training.

The computation of the kernel metric is already optimised with the KeOps library [7] which uses symbolic matrices to avoid memory overflow. Consequently, training has a squared polynomial complexity.

Once the model is trained, the registration is performed with a simple function evaluation which makes it a lot faster than non-learning method such as FLAME, elastic matching or LDDMMs.

We report the training time for a single epoch, having the setting detailed in 4.1, in Table 6.

Model	3DCODED	Ours (\mathcal{L}^{DCD})	Ours (\mathcal{L}^{GM})
Time	2min38	3min30	17min

Table 6: Mean time per epoch reported during our experiment

5.2 Limitations

As we can observe, some regions with high curvature are hardly reconstructed, especially around the eyes, the lips and the nostrils. We believe it is due to the fact that the varifold struggles to take into account both large and finer structures on the mesh. Therefore, it is possible that the model can be enhanced using normal cycles [47] which take curvature information into account. But this comes at a high computational cost.

We also point out that our model is limited by the encoder which certainly has the advantage of being robust. But, this robustness comes at a cost regarding the performances as we show in Figure 15. Indeed, the encoder of our model struggles to keep small deformations (such as a slight eyebrow movement) during the encoding part.

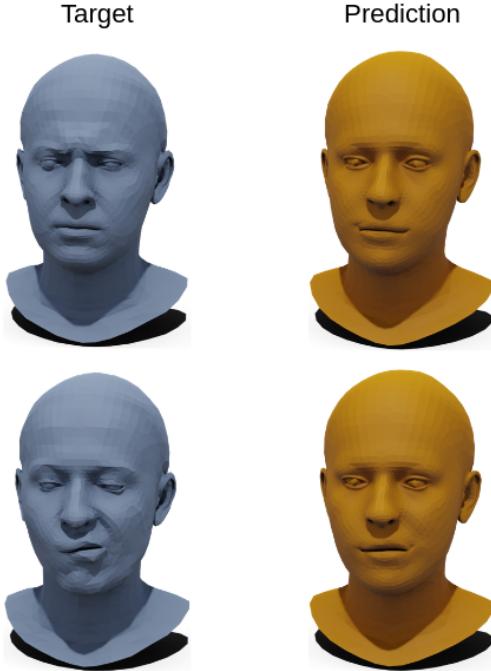


Figure 15: Examples of failed reconstruction of the expression. On the left is displayed the target and on the right its reconstruction with our model. The identity is preserved but the model fails to encode the expression.

Overall, we believe that using our loss function based on geometric measures has the potential to yield superior results than the Chamfer distance which is widely used for similar tasks. Indeed, during our experiments, the Chamfer loss has shown poor capability to set an effective objective function, especially when used to generate faces.

5.3 Perspectives related to recent mesh-invariant models

In the recent literature, models based on differential operators acting on the mesh, such as DiffusionNet [50] and DeltaConv [54] shows promising results with reassuring theoretical facts. However, these models have still not been applied to generative tasks and may require additional work but will be investigated in the future. In particular, their reliance on the intrinsic properties

of a surface, such as the Laplacian, makes them sensitive to noise and topology changes, as opposed to our PointNet-based auto-encoder. Our results for unsupervised generative tasks are thus competitive with the state-of-the-art as demonstrated by the experiences. In particular, while there is still a gap between fully supervised and unsupervised methods, our work opens up the possibility of extending the amount of available data for supervised generative tasks. Towards this objective, the incorporation of mesh invariant deep learning models within our framework is a promising avenue of work to improve the expressiveness of such data.

6 Conclusion

In this paper, we propose a novel deep learning based approach for face registration. We use a varifold representation of shapes and extend kernel metrics on varifold with a multi resolution kernel. Our asymmetric auto encoder allows to learn a map from meshes with variable discretization to a low dimensional latent space. We demonstrate that our method allows for an efficient registration of meshes, and the learned latent space allows for powerful and easy deformation on this dataset.

In the future, we plan to extend this approach to new data, such as human bodies, or animals. But the most crucial work to do remains the development of a better encoder with a similar versatility than PointNet.

Acknowledgments

This work is supported by the ANR project Human4D ANR-19-CE23-0020, and was further supported by Labex CEMPI (ANR-11-LABX-0007-01) and the Austrian Science Fund (grant no P 35813-N). The authors would also like to thank Alexandre Mouton (CNRS, UMR 8524 - LPP, Lille) and Deise Santana Maia (CNRS, UMR 9189 CRISTAL, Lille) for their advice and many fruitful conversations.

A Proof of Theorem 3.1 on the independence to mesh structure of the varifold representation

Let S be a smooth compact surface. Let $x \mapsto \vec{n}_S(x)$ be its outer normal vector field. There is a radius $r_S > 0$ such that any x point with distance less than r from S has a unique projection (i.e. closest point) on S , denoted $\xi(x)$. Note that $\xi(x) - x = x \pm d(x, S)\vec{n}_S(x)$. We have $r_S \leq 1/\kappa_S$, with equality for most surfaces.

Let f be a full triangle (corresponding to a face in a mesh) with its vertices belonging to a surface S , with center c , normal \vec{n}_f and greatest edge length η . As long as η is small enough, the projection $\xi : f \rightarrow S$ is one-to-one. We denote $\Delta = \xi(f)$. For x in f , we have $d(\xi(x), x) \leq \eta\kappa_\Delta$, κ_Δ denotes the

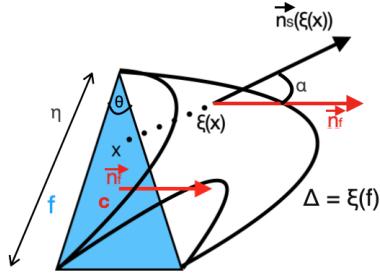


Figure 16: Face f and its projection on a Surface S .

greatest eigenvalue of the second fundamental form of S among all points of Δ . See Figure A.16.

For any $y = \xi(x)$ in Δ , and η small enough, we have ([37, 2.2.2]).

$$d(y, c) \leq d(\xi(x), x) + d(x, c) \leq \eta\kappa_\Delta + \eta \leq (\kappa_\Delta + 1)\eta.$$

On the other hand, let $\alpha_f = \max_{y \in \Delta}(\vec{n}_f, \vec{n}_S(y))$. Then we have, for any $y = \xi(x)$ in Δ [36][Corollary 1],

$$d(\vec{n}_S(y), \vec{n}_f) \leq \sqrt{2} \sin(\alpha) \leq 6\sqrt{2} \frac{\kappa_\Delta}{\sin \theta_f} \eta,$$

with θ the smallest of the three angles of f . Moreover, for η small enough [36][Corollary 2],

$$|a(f) - a(\Delta)| \leq 3a(\Delta)\kappa_\Delta\eta,$$

with $a(\cdot)$ the area of a surface in \mathbb{R}^3 .

Now take some bounded Lipschitz function $u : \mathbb{R}^3 \times \mathbb{S}^2 \rightarrow \mathbb{R}$ with Lipschitz constant $Lip(u) = k_u$. Let μ_Δ be the varifold associated with the smooth surface Δ and $\mu_f = a(f)\delta_c^{\vec{n}_f}$ as in our paper.

The main argument follows these two remarks, which are just the results above:

1. for any point $y = \xi(x)$ in Δ , the value of u at $(y, \vec{n}_S(y))$ is close to its value at (c, \vec{n}_f) . Indeed:

$$|u(y, \vec{n}_S(y)) - u(c, \vec{n}_f)| \leq k_u(d(y, c) + d(\vec{n}_S(y), \vec{n}_f)).$$

We have estimates for both of these distances that go linearly to 0 as η goes to 0, so that for any $C = 6\sqrt{2}$,

$$|u(y, \vec{n}_S(y)) - u(c, \vec{n}_f)| \leq C \frac{(\kappa_\Delta + 1)}{\sin \theta_f} k_u \eta$$

2. the difference between the area of f and that of Δ go linearly to 0 as η goes to 0, so for any $C \geq 3$

$$|(a(\Delta) - a(f))u(c, \vec{n}_f)| \leq Ca(\Delta)\kappa_\Delta \|u\|_\infty \eta,$$

with $\|u\|_\infty = \sup_{(x, n) \in \mathbb{R}^3 \times \mathbb{S}^2} |u(x, n)|$.

Letting ourselves be guided by these remarks, we have

$$\mu_f(u) = a(f)u(c, \vec{n}_f) = \int_\Delta u(c, \vec{n}_f) d\sigma(x) + (a(f) - a(\Delta))u(c, \vec{n}_f).$$

We therefore get, for C big enough (e.g., $C=10$), that $\|\mu_\Delta(u) - \mu_f(u)\|$ is bounded by:

$$\begin{aligned} & \int_\Delta |u(y, \vec{n}_S(y)) - u(c, \vec{n}_f)| d\sigma(y) + |a(f) - a(\Delta)|u(c, \vec{n}_f) \\ & \leq C \frac{(\kappa_\Delta + 1)\eta k_u a(\Delta)}{\sin \theta_f} + Ca(\Delta)\kappa_\Delta \eta |u(c, \vec{n}_f)| \\ & \leq Ca(\Delta) \frac{(\kappa_\Delta + 1)}{\sin \theta_f} (k_u + \|u\|_\infty) \eta. \end{aligned}$$

Now if X is a mesh inscribed in S whose vertices are in S , such that ξ is bijective from the full triangles of X onto S , we can sum this estimate over all faces and get

$$|\mu_S(u) - \mu_X(u)| \leq Ca(S) \frac{(\kappa_S + 1)}{\sin \theta_X} (k_u + \|u\|_\infty) \eta,$$

with κ_S the greatest eigenvalue of the second fundamental form of S (i.e. its greatest principal curvature), η its greatest edge length and θ_X the smallest angle among all faces of X .

Finally, for two meshes X, \hat{X} inscribed in S , a triangular inequality immediately gives

$$|\mu_X(u) - \mu_{\hat{X}}(u)| \leq Ca(S) \frac{(\kappa_S + 1)}{\sin \theta_X} (k_u + \|u\|_\infty) \eta,$$

with $C = 20$. Similar estimates allow the computation of kernel norms, with the addition of explicit bounds on $(k_u + \|u\|_\infty)$. Indeed, kernel norms are computed by integrating the kernel along the varifolds, and bounds on the Lipschitz constant of the kernel and its $\|\cdot\|_\infty$ are easily computed.

Note that, to keep the proof readable, the estimates were purposely rough, just to give an idea on the order of the convergence. In practice, edges are shorter and areas are smaller near high curvature areas, allowing much better approximation than suggested by the formula.

References

- [1] Mehdi Bahri, Eimear O' Sullivan, Shunwang Gong, Feng Liu, Xiaoming Liu, Michael M. Bronstein, and Stefanos Zafeiriou. Shape my face: Registering 3d face scans by surface-to-surface translation. *International Journal of Computer Vision (IJCV)*, Sep 2021.
- [2] M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision (IJCV)*, page 139–157, Feb 2005.
- [3] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation.

In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

- [4] Dongliang Cao and Florian Bernard. *Unsupervised Deep Multi-shape Matching*, page 55–71. Lecture Notes in Computer Science. Springer, 2022.
- [5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022.
- [6] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 77–85. IEEE, Jul 2017.
- [7] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunès, François-David Collin, and Ghislain Durif. Kernel operations on the gpu, with autodiff, without memory overflows. *Journal of Machine Learning Research*, page 1–6, 2021.
- [8] Nicolas Charon and Alain Trouv . The varifold representation of non-oriented shapes for diffeomorphic registration. *CoRR*, abs/1304.6108, 2013.
- [9] Siheng Chen, Chaojing Duan, Yaoqing Yang, Duanshun Li, Chen Feng, and Dong Tian. Deep unsupervised learning of 3d point clouds via graph topology inference and filtering. *IEEE Transactions on Image Processing*, 29:3183–3198, 2019.
- [10] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6970–6981, 2020.
- [11] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

- [12] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodola. Limp: Learning latent shape representations with metric preservation priors. In *European Conference on Computer Vision*, pages 19–35. Springer, 2020.
- [13] Balder Croquet, Daan Christiaens, Seth M. Weinberg, Michael Bronstein, Dirk Vandermeulen, and Peter Claes. Unsupervised diffeomorphic surface registration and non-linear modelling. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, page 118–128. Springer, 2021.
- [14] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [15] Mohamed Daoudi, Remco Veltkamp, and Anuj Srivastava. *3D Face Modeling, Analysis and Recognition*. John Wiley & Sons, Ltd, 2013.
- [16] Pim De Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations (ICLR)*, 2021.
- [17] Aysegul Dundar, Jun Gao, Andrew Tao, and Bryan Catanzaro. Fine detailed texture learning for 3d meshes with generative models. *CoRR*, abs/2203.09362, 2022.
- [18] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, Jun 2020.
- [19] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2020.
- [20] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd In-*

ternational Conference on Artificial Intelligence and Statistics, pages 2681–2690, 2019.

- [21] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pages 402–411. IEEE, 2017.
- [22] Tilmann Gneiting. Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4):1327 – 1349, 2013.
- [23] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3d-coded : 3d correspondences by deep deformation. In *European Conference on Computer Vision (ECCV)*, 2018.
- [24] Shalini Gupta, Mia K. Markey, and Alan C. Bovik. Anthropometric 3d face recognition. *International Journal of Computer Vision (IJCV)*, page 331–349, 2010.
- [25] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics*, page 1–12, Aug 2019.
- [26] Emmanuel Hartman, Yashil Sukurdeep, Eric Klassen, Nicolas Charon, and Martin Bauer. Elastic shape analysis of surfaces with second-order sobolev metrics: A comprehensive numerical framework. *International Journal of Computer Vision (IJCV)*, Jan 2023.
- [27] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
- [28] Irene Kaltenmark, Benjamin Charlier, and Nicolas Charon. A general framework for curve and surface comparison and registration with oriented varifolds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3346–3355, 2017.
- [29] Seong Uk Kim, Jihyun Roh, Hyeonseung Im, and Jongmin Kim. Anisotropic spiralnet for 3d shape completion and denoising. *Sensors*, Jan 2022.

- [30] Clément Lemeunier, Florence Denis, Guillaume Lavoué, and Florent Dupont. Representation learning of 3d meshes using an autoencoder in the spectral domain. *Computers & Graphics*, 107:131–143, 2022.
- [31] 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)*, 2017.
- [32] F. Liu, L. Tran, and X. Liu. 3d face modeling from diverse raw scan data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9407–9417, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society.
- [33] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE International Conference on Computer Vision workshops*, pages 37–45, 2015.
- [34] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.
- [35] Thomas W Mitchel, Vladimir G Kim, and Michael Kazhdan. Field convolutions for surface cnns. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 10001–10011, 2021.
- [36] J. M. Morvan and Boris Thibert. Approximation of the normal vector field of a smooth surface. *Discrete and Computational Geometry*, 32:383–400, 09 2004.
- [37] J.M. Morvan and B. Thibert. On the approximation of a smooth surface with a triangulated mesh. *Computational Geometry*, 23(3):337–352, 2002.
- [38] Naima Otberdout, Claudio Ferrari, Mohamed Daoudi, Stefano Berretti, and Alberto Del Bimbo. Sparse to dense dynamic 3d facial expression generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20385–20394, 2022.

- [39] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–206, 2018.
- [40] Emery Pierson, Mohamed Daoudi, and Sylvain Arguillère. 3d shape sequence of human comparison and classification using current and varifolds. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part III*, volume 13663 of *Lecture Notes in Computer Science*, pages 523–539. Springer, 2022.
- [41] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656, 2016.
- [42] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.
- [43] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.
- [44] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 725–741, 2018.
- [45] Günter Rote. Computing the minimum hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38(3):123–127, 1991.
- [46] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *The IEEE International Conference on Computer Vision (ICCV)*, page 1617–1627. IEEE, Oct 2019.
- [47] Pierre Roussillon and Joan Alexis Glaunès. Kernel metrics on normal cycles and application to curve matching. *SIAM Journal on Imaging Sciences*, page 1991–2038, Jan 2016.

- [48] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. *Bosphorus Database for 3D Face Analysis*, page 47–56. Springer, 2008.
- [49] I. J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, 39:811–841, 1938.
- [50] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022.
- [51] G. Taubin. Curve and surface smoothing without shrinkage. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 852–857, 1995.
- [52] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 6411–6420, 2019.
- [53] Marc Vaillant and Joan Glaunès. Surface matching via currents. In Gary E. Christensen and Milan Sonka, editors, *Information Processing in Medical Imaging*, page 381–392. Springer, 2005.
- [54] Ruben Wiersma, Ahmad Nasikun, Elmar Eisemann, and Klaus Hildebrandt. Deltaconv: Anisotropic operators for geometric deep learning on point clouds. *Transactions on Graphics*, 41(4), July 2022.
- [55] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. In *In Advances in Neural Information Processing Systems (NeurIPS), 2021*, 2021.
- [56] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M.J. Rosato. A 3d facial expression database for facial behavior research. In *7th International Conference on Automatic Face and Gesture Recognition (FGR)*, page 211–216, Apr 2006.
- [57] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable

rendering for inverse graphics and interpretable 3d neural rendering. In *International Conference on Learning Representations*, 2021.

- [58] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *The IEEE International Conference on Computer Vision*, pages 16259–16268, 2021.