Shape correspondences from learnt template-based parametrization

Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan Russell, Mathieu Aubry

Shape correspondences from learnt template-based parametrization

Thibault Groueix[1], Matthew Fisher[2], Vladimir G. Kim[2], Bryan C. Russell[2], Mathieu Aubry[1]
[1]LIGM (UMR 8049), École des Ponts, UPE, [2]Adobe Research
http://imagine.enpc.fr/~groueixt/correspondences/

Abstract. We present a new deep learning approach for matching deformable shapes by using a model which jointly encodes 3D shapes and correspondences. This is achieved by factoring the surface representation into (i) a template, that parameterizes the surface, and (ii) a learnt feature vector that parameterizes the function which transforms the template into the input surface. We show that our network can directly predict the feature vector and thus correspondences for a new input shape, but also that correspondence quality can be significantly improved by an additional regression step. This additional step improves the shape feature vector by minimizing the Chamfer distance between the input and parameterized shape. We show that this produces both a better shape representation and better correspondences. We demonstrate that our simple approach improves state of the art results on the difficult FAUST inter challenge, with an average correspondence error of 2.88cm. We also show results on the real scans from the SCAPE dataset and the synthetically perturbed shapes from the TOSCA dataset, including non-human shapes.

Keywords: 3D deep learning, computational geometry, shape matching

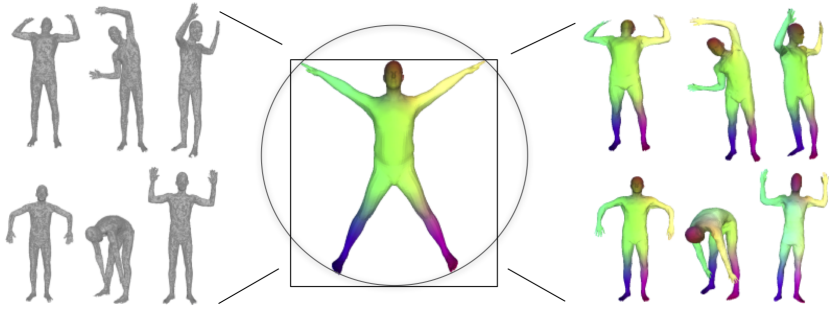


Fig. 1: Our approach predicts shape correspondences by learning a consistent mesh parameterization with a shared template.

1 Introduction

There is a growing demand for techniques that make use of the large amount of 3D content generated by modern sensor technology. An essential task is to establish reliable 3D shape correspondences between scans from raw sensor data, or between scans and a template 3D shape. This process is challenging due to low sensor resolution and high sensor noise, and is more challenging for articulated shapes, such as humans, that exhibit significant non-rigid deformations and shape variations.

Traditional approaches to estimating shape correspondences for articulated objects typically rely on intrinsic surface analysis either optimizing for an isometric map or leveraging intrinsic point descriptors. To improve correspondence quality, methods have been extended to take advantage of category-specific data priors. Effective human-specific templates and registration techniques have been developed over the last decade [1], but these methods require significant effort and domain-specific knowledge to design the parametric deformable template, create an objective function that ensures alignment of salient regions and is not prone to being stuck in local minima, and develop an optimization strategy that effectively combines global search for a good initial guess and a local refinement procedure.

In this work we propose a comprehensive, all-in-one solution to template-driven shape matching that relies on a deep encoder-decoder network. To train our network we use a single template shape (a point cloud or a mesh) with some example deformations [2,3]. Our encoder takes a deformed shape and maps it to a latent space, computing a global shape descriptor. Our decoder takes a point on the template and a global shape descriptor, and maps the template point map to the deformed shape. Our decoder network is able to use this learnt descriptor to deform the template to match the observed shape, while respecting provided ground truth correspondences. Together, our encoded shape feature and the decoder network define a deformation field from the template to the input data. This learnt deformation can then be used to establish correspondences between any two shape instances. At test time we encode two input shapes, and then decode every point of the template to both shapes, obtaining deformed templates that align with the inputs. While this step provides a good initial alignment, the template and the output might not be in precise correspondence. We use gradient descent through the decoder network to further improve the alignment, optimizing for the latent code that minimizes the Chamfer distances between the input shape and template reconstruction. The final map between two surfaces is trivially obtained by mapping via the deformed templates.

In contrast to previous work our method does not require a manually designed deformable template, instead the deformation parameters and degrees of freedom are implicitly learned by the encoder. We simply use surface-to-surface distance as our loss function during training and inference stages and do not need to define hand-crafted regularization terms. The fine-tuning optimization phase also naturally fits in this framework, where the initial guess is obtained via a feed-forward pass through a network, and it is further improved with a

gradient descent search in the latent space minimizing the distance between the input shape and the template reconstruction. The latter does not require any additional implementation effort, since neural network can be used to propagate gradients to the latent space. We demonstrate that with sufficient training data this simple approach achieves state-of-the-art results and outperforms techniques that require complex multi-term objective functions instead of the simple reconstructive loss used by our method.

2 Related work

Registration of non-rigid geometries with pose and shape variations is a long standing problem with extensive prior work. We first provide a brief overview of generic correspondence techniques. We then focus on category specific and template matching methods developed for human bodies, which are more closely related to our approach. Finally, we present an overview of deep learning approaches that have been developed for shape matching and more generally for working with 3D data.

Generic shape matching. To estimate correspondence between articulated objects, it is common to assume that their intrinsic structure (e.g., geodesic distances) remains relatively consistent across all poses [4]. Finding point-to-point correspondences that minimize metric distortion is a non-convex optimization problem, referred to as generalized multi-dimensional scaling [5]. This optimization is typically sensitive to an initial guess [6], and thus existing techniques rely on local feature point descriptors such as HKS [7] and WKS [8], and use hierarchical optimization strategies [9,10]. Some relaxations of this problem have been proposed such as: formulating it as Markov random field and using linear programming relaxation [11], optimizing for soft correspondence [12,13,14], restricting correspondence space to conformal maps [15,16], heat kernel maps [17], and aligning functional bases [18].

While these techniques are powerful generic tools, some common categories, such as humans, can benefit from a plethora of existing data [2] to leverage stronger class-specific priors.

Template-based shape matching. A natural way to leverage class-specific knowledge is through the explicit use of a shape model. While such template-based techniques provide the best correspondence results they require a careful parameterization of the template, which took more than a decade of research to reach the current level of maturity [19,20,21,22,1]. For all of these techniques, fitting this representation to an input 3D shape requires also designing an objective function that is typically non-convex and involves multiple terms to guide the optimization to the right global minima. In contrast, our method only relies on a single template 3D mesh and surface reconstruction loss. It leverages a neural network to learn how to parameterize the human body while optimizing

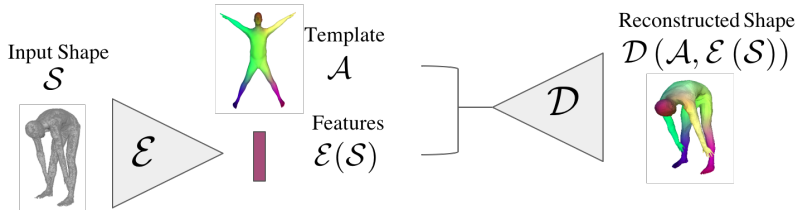
for the best reconstruction. The forward pass through this network provides a good initial guess for template fitting, and a simple back-propagation is used to refine the fitting parameters, replacing complex multi-term objective functions used in traditional optimization frameworks.

Deep learning for shape matching. Another way to leverage priors and training data is to learn better point-wise shape descriptors using human models with ground truth correspondence. Several neural network based methods have recently been developed to this end to analyze meshes [23,24,25,26] or depth maps [27]. One can further improve these results by leveraging global context, for example, by estimating an inter-surface functional map [28]. These methods still rely on hand-crafted point-wise descriptors [29] as input and use neural networks to improve results. The resulting functional maps only align basis functions and additional optimization is required to extract consistent point-to-point correspondences [18]. One would also need to optimize for template deformation to use these matching techniques for surface reconstruction. In contrast our method does not rely on hand-crafted features (it only takes point coordinates as input) and implicitly learns a human body representation. It also directly outputs a template deformation and the correspondences are directly estimated by projecting to the template.

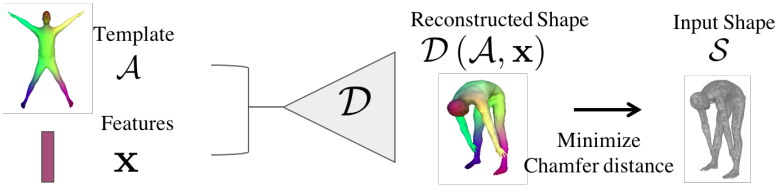
Deep Learning for 3D data. Following the success of deep learning approaches for image analysis, many techniques have been developed for processing 3D data, going beyond local descriptor learning to improve classification, segmentation, and reconstruction tasks. Existing networks operate on various shape representations, such as volumetric grids [30,31], point clouds [32,33,34], geometry images [35,36], seamlessly parameterized surfaces [37], by aligning a shape to a grid via distance-preserving maps [38], or by predicting chart representations [39]. We build on these works in several ways. First, we process the point clouds representing the input shapes using an architecture similar to [32]. Second, similar to [36], we learn a surface representation. However, we do not explicitly encode correspondences in the output of a convolution network, but implicitly learn them by optimizing for parameters of the generation network as we optimize for reconstruction losss.

3 Method

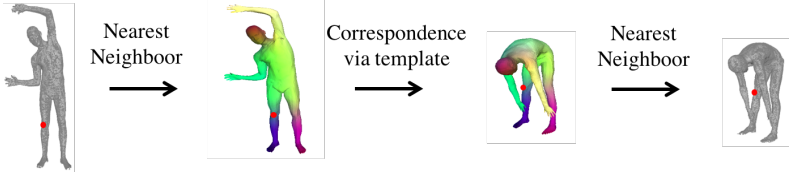
Our goal is, given a reference shape $\mathcal{S}_r$ and a target shape $\mathcal{S}_t$, to return a set of point correspondences $\mathcal{C}$ between the shapes. We do so using two key ideas. First, we learn to predict a transformation between the shapes instead of directly learning the correspondences. This transformation, from 3D to 3D can indeed be represented by a neural network much more easily than the association between variable and large number of points. The second idea is to learn transformations only from one template $\mathcal{A}$ to any shape. Indeed, the large variety of possible



(a) **Initial reconstruction**



(b) **Refined reconstruction**



(c) **Correspondences**

Fig. 2: **Method overview.** (a) A feed-forward pass in our autoencoder encodes input point cloud $\mathcal{S}_\mathbf{r}$ to latent code $\mathcal{E}_\phi(\mathcal{S}_r)$ and reconstruct $\mathcal{S}_\mathbf{r}$ using $\mathcal{E}_\phi(\mathcal{S}_r)$ to deform the template $\mathcal{A}$. (b) We refine the reconstruction $\bar{\mathcal{S}}_\mathbf{r}$ by performing a regression step over the latent variable $\mathcal{E}_\phi(\mathcal{S}_r)$, minimizing the Chamfer distance between $\bar{\mathcal{S}}_\mathbf{r}$ and $\mathcal{S}_\mathbf{r}$. (c) Finally to match a point $\mathbf{q}_r$ on $\mathcal{S}_\mathbf{r}$ to a point $\mathbf{q}_t$ on $\mathcal{S}_\mathbf{t}$, we look for the nearest neighbor $\mathbf{p}_r$ of $\mathbf{q}_r$ in $\bar{\mathcal{S}}_\mathbf{r}$, which is by design in correspondence with $\mathbf{p}_t$; and look for the nearest neighbor $\mathbf{q}_t$ of $\mathbf{p}_t$ on $\mathcal{S}_\mathbf{t}$.

poses of humans makes considering all pairs of possible poses intractable during training. We instead decouple the correspondence problem into finding two sets of correspondences to a common template shape. We can then form our final correspondences between the input shapes via indexing through the template shape. An added benefit is during training we simply need to vary the pose for a single shape and use the known correspondences to the template shape as the supervisory signal.

Our approach has three main steps which are visualized figure 2. First, a feed-forward pass through out encoder network generates an initial global shape descriptor (Section 3.1). Second, we use gradient descent through our decoder network to refine this shape descriptor to improve the reconstruction quality (Section 3.2). We can then use the template to correspond points between any two input shapes (Section 3.3).

3.1 Learning 3D shape reconstruction by template deformation

To put an input shape $\mathcal{S}$ in correspondence with a template $\mathcal{A}$, our first goal is to design a neural network that will take $\mathcal{S}$ as input and predict transformation parameters. We do so by training an encoder-decoder architecture. The encoder $\mathcal{E}_\phi$ defined by its parameters ϕ takes as input 3D points, and is a simplified version of the network presented in [32]. It applies to each input 3D point coordinate a multi-layer perceptron with hidden feature size of 64, 128 and 1024, then maxpooling over the resulting features over all points followed by a linear layer, leading to feature of size 1024 $\mathcal{E}_\phi(\mathcal{S})$. This feature, together with the 3D coordinates of a point on the template $\mathbf{p} \in \mathcal{A}$ is taken as input to the decoder $\mathcal{D}_\theta$ with parameters θ, which is trained to predict the position $\mathbf{q}$ of the corresponding point in the input shape. This decoder is a multi-layer perceptron with hidden layers of size 1024, 512, 254 and 128, followed by a hyperbolic tangent. This architecture maps any points from the template domain to the reconstructed surface. By sampling the template more or less densely, we can generate an arbitrary number of output points by sequentially applying the decoder over sampled template points.

This encoder-decoder architecture is trained end-to-end. We assume that we are given as input a training set of N shapes $\{\mathcal{S}_i\}_{i=1}^N$ with each shape having a set of P vertices $\{\mathbf{q}_j\}_{j=1}^P$. For each point $\mathbf{q}_j$ on a training shape, we assume that we know the correspondence $\mathbf{p}_j \leftrightarrow \mathbf{q}_j$ to a point $\mathbf{p}_j \in \mathcal{A}$ on the template $\mathcal{A}$. Given these training correspondences, we learn the encoder $\mathcal{E}_\phi$ and decoder $\mathcal{D}_\theta$ by optimizing the following reconstruction loss,

$$\mathcal{L}'(\theta, \phi) = \sum_{i=1}^{N} \sum_{j=1}^{P} |\mathcal{D}_\theta(\mathbf{p}_j; \mathcal{E}_\phi(\mathcal{S}_i)) - \mathbf{q}_{i,j}|^2. \quad (1)$$

We optimize this loss using the Adam solver, with a learning rate of 10^{-3} for 25 epochs then 10^{-4} for 2 epochs, batches of 32 shapes using 6890 points per shape.

One interesting aspect of this step is that it learns jointly a parametrization of the input shapes via the decoder and to perdict the parameters $\mathcal{E}_\phi(\mathcal{S})$ for this parametrization via the encoder. However, the predicted parameters $\mathcal{E}_\phi(\mathcal{S})$ for an input shape $\mathcal{S}$ are not necessarily optimal, because of the limited power of the encoder. Optimizing these parameters turns out to be important for the final results, and is the focus of the second step of our pipeline.

3.2 Optimizing shape reconstruction

We now assume that we are given a shape $\mathcal{S}$ as well as learned weights for the encoder $\mathcal{E}_\phi$ and decoder $\mathcal{D}_\theta$ networks. To find correspondences between the template shape and the input shape, we will use a nearest neighbor search to find correspondences between that input shape and its reconstruction. For this step to work, we need the reconstruction to be very accurate. The reconstruction given by the parameters $\mathcal{E}_\phi(\mathcal{S})$ is only approximate and can be improved. Since

Algorithm 1: Algorithm for finding 3D shape correspondences

Input : Reference shape $\mathcal{S}_r$ and target shape $\mathcal{S}_t$
Output: Set of 3D point correspondences $\mathcal{C}$

#Regression steps over latent code to find best reconstruction of $\mathcal{S}_r$ and $\mathcal{S}_t$
$\mathbf{x}_r \leftarrow \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \mathcal{S}_r)$
$\mathbf{x}_t \leftarrow \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \mathcal{S}_t)$
$\mathcal{C} \leftarrow \varnothing$
Matching of $\mathbf{q}_r \in \mathcal{S}_r$ to $\mathbf{q}_t \in \mathcal{S}_t$
foreach $\mathbf{q}_r \in \mathcal{S}_r$ **do**
  $\mathbf{p} \leftarrow \arg\min_{\mathbf{p}' \in \mathcal{A}} |\mathcal{D}_\theta(\mathbf{p}'; \mathbf{x}_r) - \mathbf{q}_r|^2$
  $\mathbf{q}_t \leftarrow \arg\min_{\mathbf{q}' \in \mathcal{S}_t} |\mathcal{D}_\theta(\mathbf{p}; \mathbf{x}_t) - \mathbf{q}'|^2$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{q}_r, \mathbf{q}_t)\}$
end
return $\mathcal{C}$

we do not know correspondences between the input and the generated shape, we cannot minimize the loss given in equation 1, which requires correspondences. Instead, we minimize with respect to the feature $\mathbf{x}$ the Chamfer distance between the reconstructed shape and the input:

$$\mathcal{L}(\mathbf{x}; \mathcal{S}) = \sum_{\mathbf{p} \in \mathcal{A}} \min_{\mathbf{q} \in \mathcal{S}} |\mathcal{D}_\theta(\mathbf{p}; \mathbf{x}) - \mathbf{q}|^2 + \sum_{\mathbf{q} \in \mathcal{S}} \min_{\mathbf{p} \in \mathcal{A}} |\mathcal{D}_\theta(\mathbf{p}; \mathbf{x}) - \mathbf{q}|^2 . \quad (2)$$

Starting from the parameters predicted by our first step $\mathbf{x} = \mathcal{E}_\phi(\mathcal{S})$, we optimize this loss using the Adam solver for 3,000 iterations with learning rate $5 * 10^{-4}$. Note that the good initialization given by our first step is key since Equation 2 corresponds to a highly non-convex problem.

3.3 Finding 3D shape correspondences

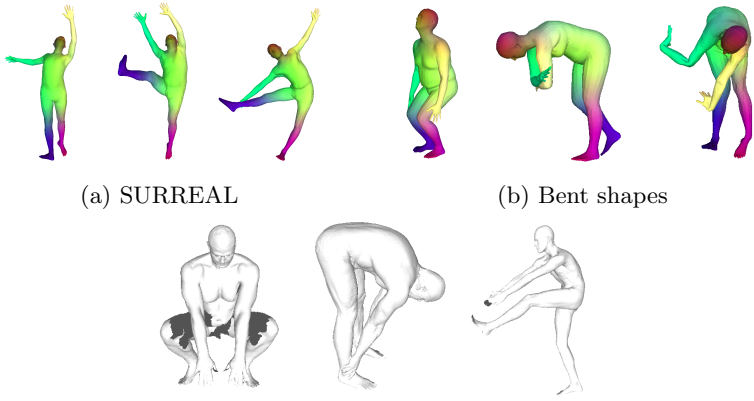
To recover correspondences between two 3D shapes $\mathcal{S}_r$ and $\mathcal{S}_t$, we first compute the parameters to deform the template to these shapes, $\mathbf{x}_r$ and $\mathbf{x}_t$, using our first two steps. Next, given a 3D point $\mathbf{q}_r$ on the reference shape $\mathcal{S}_r$, we first find the point $\mathbf{p}$ on the template $\mathcal{A}$ such that its transformation with parameters $\mathbf{x}_r$, $\mathcal{D}_\theta(\mathbf{p}; \mathbf{x}_r)$ is closest to $\mathbf{q}_r$. Finally we find the 3D point $\mathbf{q}_t$ on the target shape $\mathcal{S}_t$ that is the closest tp the transformation of $\mathbf{p}$ with parameters $\mathbf{x}_t$, $\mathcal{D}_\theta(\mathbf{p}; \mathbf{x}_t)$.Our algorithm is summarized in Algorithm 1 and illustrated in Figure 2.

4 Results

In this section we show qualitative and quantitative results for our approach and compare against baselines.

4.1 Data

We describe the datasets used in our study.



(a) SURREAL (b) Bent shapes

(c) FAUST

Fig. 3: **Examples of the different datasets used in the paper.**

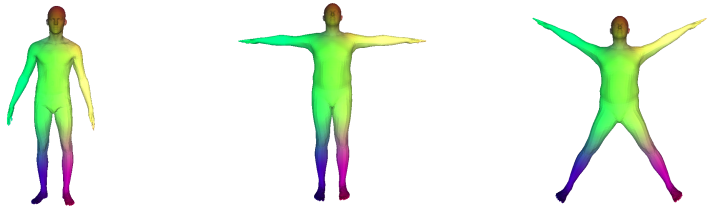
Synthetic training data. To train our algorithm, we require a large set of shapes with ground-truth correspondences. Since these ground-truth correspondences are time consuming to obtain, we rely on synthetic data for training our model. More precisely, we use SMPL [2], a state-of-the-art generative model for synthetic humans. To obtain realistic human body shape and poses from the SMPL model, we use the pose parameters from the SURREAL dataset [3] where they inferred $4 \cdot 10^6$ pose parameters from 2000 video sequences and 1700 body shape parameters. We randomly sampled among these pose and shape parameters to generate 10^5 synthetic males and 10^5 synthetic females.

One limitation of the SURREAL dataset is it does not include any humans bent over. Our algorithm generalized poorly to these poses. To overcome this limitation, we generated an extension of the dataset. We first manually estimated 7 key-joint parameters (among 23 joints in the SMPL skeletons) to generate bent humans. We then sampled the 7 parameters around these values, and used random parameters from the SURREAL dataset for the other pose and body shape parameters. Note that not all meshes generated with this strategy are realistic as shown in figure 3. They however allow us to better cover the space of possible poses, and we added $3 \cdot 10^4$ shapes generated with this method to our dataset. Our final dataset thus has $2.3 \cdot 10^5$ human meshes with a large variety of realistic poses and body shapes, each having 6890 vertices in correspondence.

Real testing data. We evaluate our algorithm on the FAUST dataset [2]. The FAUST dataset is separated into 100 training shapes, for which correspondences are available, and 200 testing shapes. In this paper, we never used the training set, except for a single baseline experiment, and we focus on the (more challenging) test set. The test set consists of scans with approximately 170,000 vertices.

The scans include some noise and may have holes in them, typically missing part of the feet. Given a pair of meshes, the task is to associate to each vertex of the first shape a vertex of the second shape. Two challenges are available, focusing on intra- and inter-subject correspondences. We focused on the more challenging inter-subject correspondence task. The error is the average Euclidean distance between the estimated projection and the ground-truth projection. We evaluated our method through the publicly available online server and are the best public results at the time of submission[1].

Shape normalization. To be processed and reconstructed by our network, the training and testing shapes must be normalized in a similar way. Since the vertical direction is respected in the FAUST dataset, we used synthetic shapes with approximately the same vertical axis. We also kept a fixed orientation around this vertical axis, and at test time tested for each of 50 different orientations and selected the one which leads to the smaller reconstruction error in term of Chamfer distance. We thus get invariance to the orientation at testing time, but we believe a strategy which learns to reconstruct shapes in all orientations and would thus be invariant to orientation from the training data would also succeed. Finally, we centered all meshes according to the center of their bounding box and, for the training data only, added a random translation in each direction sampled uniformly between -3cm and 3cm to increase robustness.



(a) "FAUST" template (b) "Zero" template (c) "Separated" template

Fig. 4: **Shapes for template study.** We evaluate three different template shapes used in our model.

4.2 Experiments

The method presented above leads to the best results to date on the FAUST-inter dataset: 2.878 cm : **an improvement of 8% over state of the art**, 3.12cm for [1] and 4.82cm for [28]. In this part, we analyze the key components of our pipeline and their contribution in the final quality of our results.

[1] `http://faust.is.tue.mpg.de/challenge/Inter-subject_challenge`

template 0	Faust error (cm)
"FAUST" template	**3.255**
"Zero" template	3.385
"Separated" template	3.314

Table 1: **Comparison of different template shapes.** We compare different choices for the template shape shown in Figure 4. Notice that the neutral "FAUST" template performs best out of the three tested shapes.

Choice of template. The template is a critical element for our method. We experimented with three different templates: (i) a "FAUST" template associated with SMPL parameters fitted to a body in a neutral pose in the FAUST training set, (ii) a "zero" template corresponding to the "zero" shape of SMPL, and (iii) a "separated" template in which this "zero" shape is modified to have the legs better separated and the arms higher. Figure 4 shows the different templates, while table 1 shows quantitative results using the different templates. Interestingly, the best results were obtained with the more "natural" template, selected in the "FAUST" training dataset, rather than with the templates from simple SMPL parameters, where points from different body parts seem easier to separate.

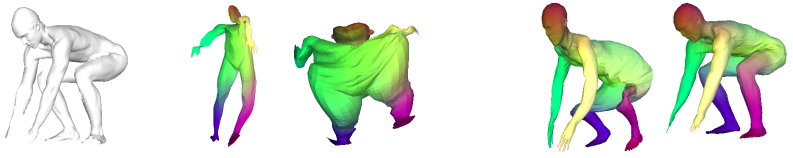
training data	Faust error (cm)
FAUST training set	18.219
non-augmented synthetic dataset	5.625
augmented synthetic data	**3.255**

Table 2: **Results of our method, trained on different datasets.** The difference in performance between the basic synthetic dataset and its augmented version is mostly due to failure on specific poses, such as the one in Figure 3.

Training data. By default, we report the results of our model trained on the augmented synthetic dataset, including poses from SURREAL and additional bent human poses. The FAUST training set is indeed too small to train our network to generalize, which includes only 10 different poses and body shapes. Training on synthetic data with a large variety of poses helps overcome this generalization problem. However, if the synthetic dataset does not include certain human poses (such as bent-over humans), the method will fail on these poses. The quantitative results corresponding to these three experiments: training on FAUST, training on SURREAL shapes, and training on SURREAL shapes augmented with bent shapes, are reported in table 2. A qualitative example is given figure 5.

Necessary amount of training data and supervision. We trained our method with 1 000 and 10 000 training shapes. As expected, this decreases the performance,

respectively to 5.76cm and 4.70cm average error for the FAUST-inter, but still yields competitive results. Not augmenting the dataset with bent shapes makes the method fail on 4 pairs of shapes (out of 40).



(a) Input (b) FAUST training data (c) Augm. synth. training data

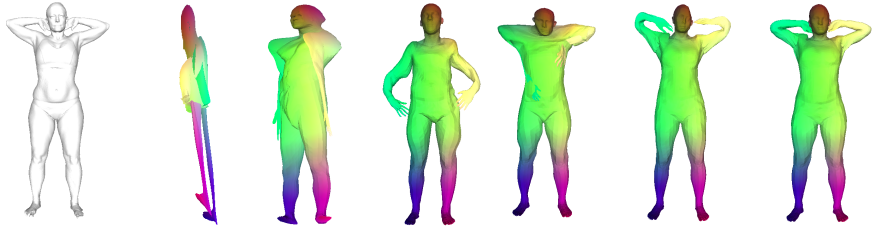
Fig. 5: **Dependence on the training data.**
For a given target shape (a) reconstructed shapes when the network is trained on FAUST training set (b) and on our augmented synthetic training set (c), before (left) and after (right) the optimization step.

Reconstruction optimization. The second step of our pipeline, which finds the optimal features for reconstruction is crucial to obtain high quality results. This is because the nearest neighbors used in the matching step are sensitive to small errors in alignment. The regression step compensates for the lack of robustness of nearest neighbors by matching, as closely as possible, the reconstruction with the original mesh. This also results in qualitatively better reconstructions. This optimization however converges to a good optimum only if it is initialized with a reasonable reconstruction, as visualized in Figure 6. Since we optimize using Chamfer distance, and not correspondences, we also rely on the fact that the network was trained to generate humans in correspondence. As a consequence, we expect that as we explore the latent space during the parameter opimization, the correspondences between generated shapes and the template will still be meaningful.

We ran an ablation study, removing this optimization completely, or optimizing only an asymmetric version of the Chamfer distance. The results are reported in Table 3 and Figure 7. The quantitative results highlight the importance of the optimization step, and show that the best approach is to optimize the reconstruction using the full Chamfer distance. Figure 7 illustrates that optimizing an asymmetric Chamfer distance can in some cases, especially when the 3D scans have holes, produce qualitatively better results.

We show that using a high density template (∼200k vertices) for the nearest neighbor step, and sampling points regularly on the surface during training (instead of taking a random subset of the irregularly sampled synthetic shapes) leads to our best result, 2.878 cm, **an improvement of 8% over state of the art**.

We also evaluated our method on the **FAUST intra challenge**, and although our method cannot take advantage of the fact that two meshes represent



(a) Input (b) Random init. (c) Wrong init. (d) Valid init.

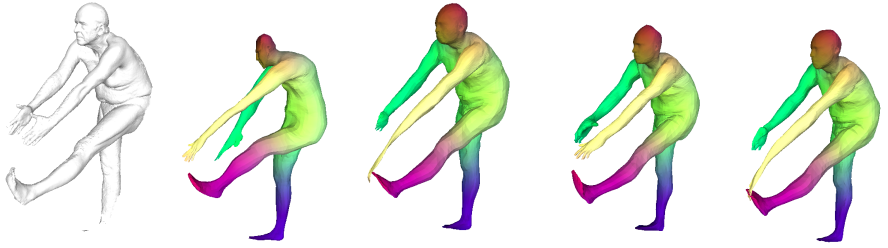
Fig. 6: **Reconstruction optimization.** The quality of the initialization (i.e. the output of the first step of our algorithm) is crucial for the result of the deformation optimization step. This figure visualize for a given target shape (a) and for different initializations (left of (a), (b) and (c)) the result of the optimization. If the deformation is initialized with random (b) or bad (c) parameters, the optimization converges to bad local minima. If it is initialized with a reasonable transformation (d) it converges to a shape very close to the target ((d), right).

the same person (because correspondences are established through the generic human template), our method is the second best performing (average error of 1.99 cm)

Method	Faust error (cm)
Without regression	6.29
With regression, Chamfer asym (R attracts T)	4.023
With regression, Chamfer asym (T attracts R)	3.336
With regression, Chamfer sym	3.255
With regression, Chamfer sym + Regular Sampling	3.048
With regression, Chamfer sym + Regular Sampling + High-Res template	**2.878**

Table 3: **Importance of the reconstruction optimization step.** Performing a search in the latent space by backpropagation to get the best reconstruction is key to the success of the nearest neighbors step.

Correspondences between non-human shapes High-quality parametric models of animals are now available; SMAL [40] provides the SMPL equivalent for several animals. Recent papers estimate model parameters from images, but no large-scale parameter set is yet available. For training we thus generated models of horses from SMAL with random parameters (drawn from a Gaussian distribution of *ad-hoc* variance 0.2) and evaluated successfully on the qualitatively different horse models from TOSCA dataset (shown in Fig 8c) demonstrating the



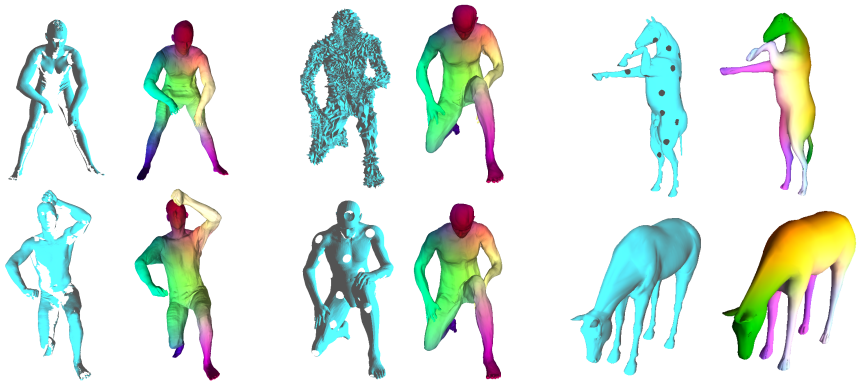
(a) target T (b) Result R (c) R attracts T (d) T attracts R (e) Both ways

Fig. 7: **Asymmetric Chamfer loss in reconstruction optimization.** Given an input scan, with holes (a), our network outputs a reconstruction result (b), that can be improved by an optimization step. When the scan has holes, it is better to only consider a loss where the scan attracts the reconstruction (d), rather than using a loss where reconstruction attracts the scan (c), or the Chamfer distance where they attract each other (e).

generality of our method. The same procedure can be applied to a large group of animals, whose shapes are successfully encoded by SMAL [40]. 5 categories are available in SMAL, and SMALR [41] introduces a method to generalize to new categories from an image dataset alone. In total 17 additional categories are available. Note that if the templates for two animal are in correspondences (as is the case for SMAL), our method can be used to get inter-category correspondences for animals. We qualitatively demonstrate this on hippopotame/horses in the appendix [42].

Partial data / robustness to perturbations - SCAPE/TOSCA. The SCAPE dataset provides meshes aligned to real scans and includes poses different from our training dataset. When applying a network trained directly on our SMPL data, we obtain satisfying performance, namely 3.14cm average Euclidean error. Quantitative comparison of correspondence quality in term of geodesic error are given in Fig 9. We perform better than most methods but Deep Functional Maps. SCAPE also allows evaluation on real partial scans. Quantitatively, the error on these partial meshes was 4.04cm, similar to the performance on the full meshes. Qualitative results are given Fig 8a. The TOSCA dataset provides several versions of the same synthetic human with different perturbations. We found that our method, still trained only on SMPL and SMAL data, is robust to all perturbations (isometry, noise, shotnoise, holes, micro-holes, topology changes, and sampling), except scale, which can be trivially fixed by normalizing all meshes to have consistent surface area. Examples of representative qualitative results are shown Fig 8b and quantitative results are reported in appendix [42].

Unsupervised correspondences. A natural question is whether our method could be trained without correspondence supervision, i.e. simply using a recon-



(a) SCAPE (b) TOSCA (c) TOSCA animals

Fig. 8: **Other datasets.** Left images show the input, right images the reconstruction with color giving a sense of correspondences. Our method works with real incomplete scans, strong synthetic perturbations, and on non-human shapes.

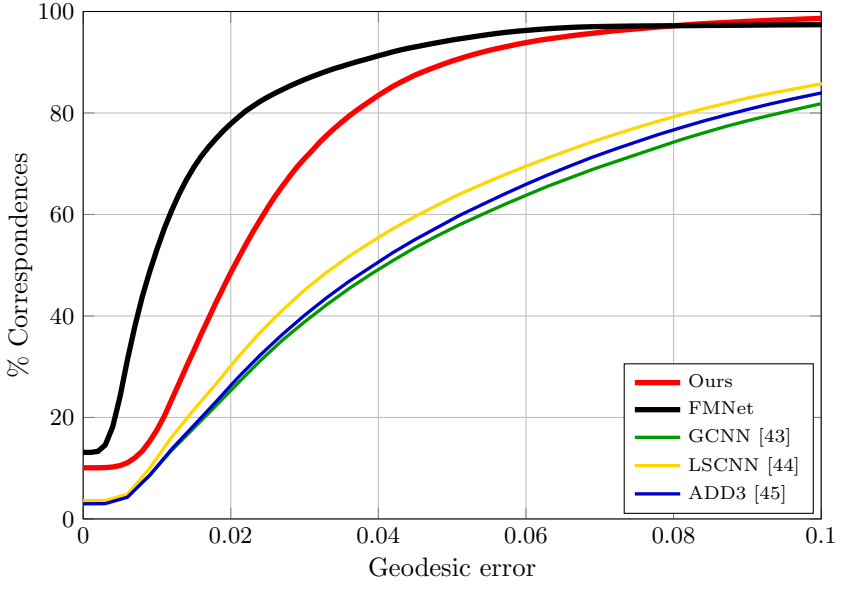


Fig. 9: Comparison with learning-based shape matching approaches on the SCAPE dataset. Our method is trained on synthetic data, FMNet was trained on FAUST data, and all other methods on SCAPE.

struction loss similar to the one described in Equation 2. One could indeed expect that an optimal way to deform the template into training shapes would respect correspondences. We thus trained a network only for reconstruction quality, without using correspondences, but we found that network did not respect correspondences between the template and the input shape, as visualized figure 10. However, these results improve with adequate regularization such as a cost encouraging regularity of the mapping between the template and the reconstruction (e.g. a Laplacian regularization, similar to Kanazawa et. al. [46], and/or a cost enforcing the ratio of face edges in the template or reconstruction to be close to one). We trained such a network with the same data as in the paper but **without any correspondence supervision** and obtained a 4.88cm of error on the FAUST-inter (similar to Deep Functional Map which had an error of 4.83 cm), i.e. close to the second best performance. This demonstrate that our method can be efficient even without correspondence supervision. Further details on regularization losses are in the appendix [42].

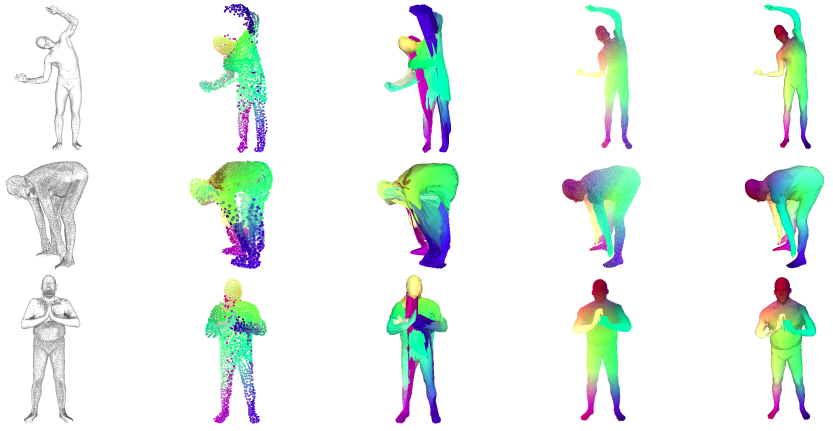
Loss	Faust error (cm)
Chamfer distance (unsupervised)	8.727
Chamfer distance (unsupervised) + Regularization	4.835
Correspondences, eq. 1 (supervised)	**2.878**

Table 4: Results with and without supervised correspondences

Rotation invariance We handled rotation invariance by rotating the shape and selecting the orientation for which the reconstruction is optimal. As an alternative, we tried to learn a network directly invariant to rotations around the vertical axis. It turned out the performances were slightly worse on FAUST-inter (3.10cm), but still better than the state of the art. We believe this is due to the limited capacity of the network and should be tried with a larger network. However, interestingly, this rotation invariant network seems to have increased robustness and provided slightly better results on SCAPE.

5 Conclusion

We have demonstrated an encoder-decoder deep network architecture that can generate human shape correspondences competitive with state-of-the-art approaches and that uses only simple reconstruction and correspondence losses. Our key insight is to factor the problem into an encoder network that produces a global shape descriptor, and a decoder network that uses this encoded descriptor to map points on a template domain back to the original geometry. A straightforward regression step uses gradient descent through the decoder network to significantly improve the final correspondence quality.



(a) Input P.C. (FAUST) (b) P.C. after optim. (c) Mesh after optim. (d) P.C. after optim + Regul (e) Mesh after optim + Regul

Fig. 10: **Unsupervised correspondences** There is a clear distortion between the reconstructed shapes and the template, the left foot of the template being matched to left hand of the reconstruction, but this distortion is consistent for different poses. We visualize for different inputs (a), the point clouds predicted by our algorithm after our optimization step (b,d) and the corresponding meshes (c,e). Note that without regularization, because of the strong distortion, the meshes appear to barely match to the input, while the point clouds are reasonable. On the other hand surface regularization creates reasonable meshes.

Our method currently assumes a fixed template, and another promising area for future research is to look into techniques that can combine multiple templates to deal with a wider range of underlying topologies. We believe that our encoder-decoder template-based approach to modeling correspondences will prove an effective basis for these future explorations in this area.

References

1. Zuffi, S., Black., M.J.: The stitched puppet: A graphical model of 3d human shape and pose. (2015)
2. Bogo, F., Romero, J., Loper, M., Black, M.J.: FAUST: Dataset and evaluation for 3D mesh registration. (June 2014)
3. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. CVPR (2017)
4. Mémoli, F., Sapiro, S.: A theoretical and computational framework for isometry invariant recognition of point cloud data. (2005)
5. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. Proc. National Academy of Sciences (PNAS) (2006)

6. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Efficient computation of isometry-invariant distances between surfaces. SIAM J. Scientific Computing (2006)
7. Sun, J., Ovsjanikov, M., Guibas, L.: One point isometric matching with the heat kernel. In: Computer Graphics Forum (Proc. of SGP). (2010)
8. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. (2011)
9. Sahillioglu, Y., Yemez, Y.: Coarse-to-fine combinatorial matching for dense isometric shape correspondence. Computer Graphics Forum (2011)
10. D.Raviv, A.Dubrovina, R.Kimmel: Hierarchical framework for shape correspondence. Numerical Mathematics: Theory, Methods and Applications (2013)
11. Chen, Q., Koltun, V.: Robust nonrigid registration by convex optimization. International Conference on Computer Vision (ICCV) (2015)
12. Solomon, J., Nguyen, A., Butscher, A., Ben-Chen, M., Guibas, L.: Soft maps between surfaces. SGP (2012)
13. Kim, V.G., Li, W., Mitra, N.J., DiVerdi, S., Funkhouser, T.: Exploring Collections of 3D Models using Fuzzy Correspondences. Transactions on Graphics (Proc. of SIGGRAPH) **31**(4) (2012)
14. Solomon, J., Peyre, G., Kim, V.G., Sra, S.: Entropic metric alignment for correspondence problems. Transactions on Graphics (Proc. of SIGGRAPH) (2016)
15. Lipman, Y., Funkhouser, T.: Mobius voting for surface correspondence. ACM Transactions on Graphics (Proc. SIGGRAPH) **28**(3) (2009)
16. Kim, V.G., Lipman, Y., Funkhouser, T.: Blended Intrinsic Maps. Transactions on Graphics (Proc. of SIGGRAPH) **30**(4) (2011)
17. Ovsjanikov, M., Mérigot, Q., Mémoli, F., Guibas, L.: One point isometric matching with the heat kernel. (2010)
18. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: A flexible representation of maps between shapes. ACM Trans. Graph. (2012)
19. Allen, B., Curless, B., Popovic, Z.: Articulated body deformation from range scan data. (2002)
20. Allen, B., Curless, B., Popovic, Z.: The space of human body shapes: reconstruction and parameterization from range scans. (2003)
21. Allen, B., Curless, B., Popovic, Z.: Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis. (2006)
22. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. (2015)
23. Rodola, E., Rota Bulo, S., Windheuser, T., Vestner, M., Cremers, D.: Dense non-rigid shape correspondence using random forests. CVPR (2014)
24. Masci, J., Boscaini, D., Bronstein, M.M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. (2015) 37–45
25. Boscaini, D., Masci, J., Rodola, E., Bronstein, M.M.: Learning shape correspondence with anisotropic convolutional neural networks. NIPS (2016)
26. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. CVPR (2017)
27. Wei, L., Huang, Q., Ceylan, D., Vouga, E., Li, H.: Dense human body correspondences using convolutional networks. In: Computer Vision and Pattern Recognition (CVPR). (2016)
28. Litany, O., Remez, T., Rodola, E., Bronstein, A.M., Bronstein, M.M.: Deep functional maps: Structured prediction for dense shape correspondence. ICCV (2017)

29. Tombari, F., Salti, S., Stefano, L.D.: Unique signatures of histograms for local surface description. ECCV (2010)
30. Girdhar, R., Fouhey, D., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. (2016)
31. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. (2015) 1912–1920
32. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. (2017)
33. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. (2017)
34. Fan, H., Su, H., Guibas, L.: A point set generation network for 3D object reconstruction from a single image. (2017)
35. Sinha, A., Bai, J., Ramani, K.: Deep learning 3d shape surfaces using geometry images. (2016)
36. Sinha, A., Unmesh, A., Huang, Q., Ramani, K.: Surfnet: Generating 3d shape surfaces using deep residual networks. (2017)
37. Maron, H., Galun, M., Aigerman, N., Trope, M., Dym, N., Yumer, E., Kim, V.G., Lipman, Y.: Convolutional neural networks on surfaces via seamless toric covers. SIGGRAPH (2017)
38. Ezuz, D., Solomon, J., Kim, V.G., Ben-Chen, M.: Gwcnn: A metric alignment layer for deep shape analysis. SGP (2017)
39. Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. (2018)
40. Zuffi, S., Kanazawa, A., Jacobs, D., Black, M.J.: 3D menagerie: Modeling the 3D shape and pose of animals. (2017)
41. Zuffi, S., Kanazawa, A., Black, M.J.: Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. (2018)
42. : Supplementary material (appendix) for the paper `https://http://imagine.enpc.fr/~groueixt/correspondences/arxiv`
43. Masci, J., Boscaini, D., Bronstein, M.M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. 3dRR (2015)
44. Boscaini, D., Masci, J., Melzi, S., Bronstein, M.M., Castellani, U., Vandergheynst, P.: Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. **34**(5) (2015) 13–23
45. Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.M., Cremers, D.: Anisotropic diffusion descriptors. Computer Graphics Forum **35**(2) (2016) 431–441
46. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. CoRR **abs/1803.07549** (2018)
47. Sorkine, O.: Differential representations for mesh processing. **25** (12 2006) 789–807
48. Meyer, M., Desbrun, M., Schr, P., Barr, A.: Discrete differential-geometry operators for triangulated 2-manifolds. **3** (11 2001)

6 Supplementary

6.1 Quantitative results for perturbations on TOSCA

We evaluate quantitatively the robustness of our method to perturbation on the TOSCA dataset. It consists of one horse shape with different added perturbations, namely noise, shotnoise, sampling, scale, local scale, topology, holes, microholes, and isometry. We report in 5, quantitative results for each perturbation (with a gradual strength from 1 to 5) and show qualitative reconstruction with correspondences suggested by colors for each category with maximum strength in 11. Surprisingly, adding noise can enhance the quantitative error.

Table 5: Quantitative results for perturbations on TOSCA for the horse category

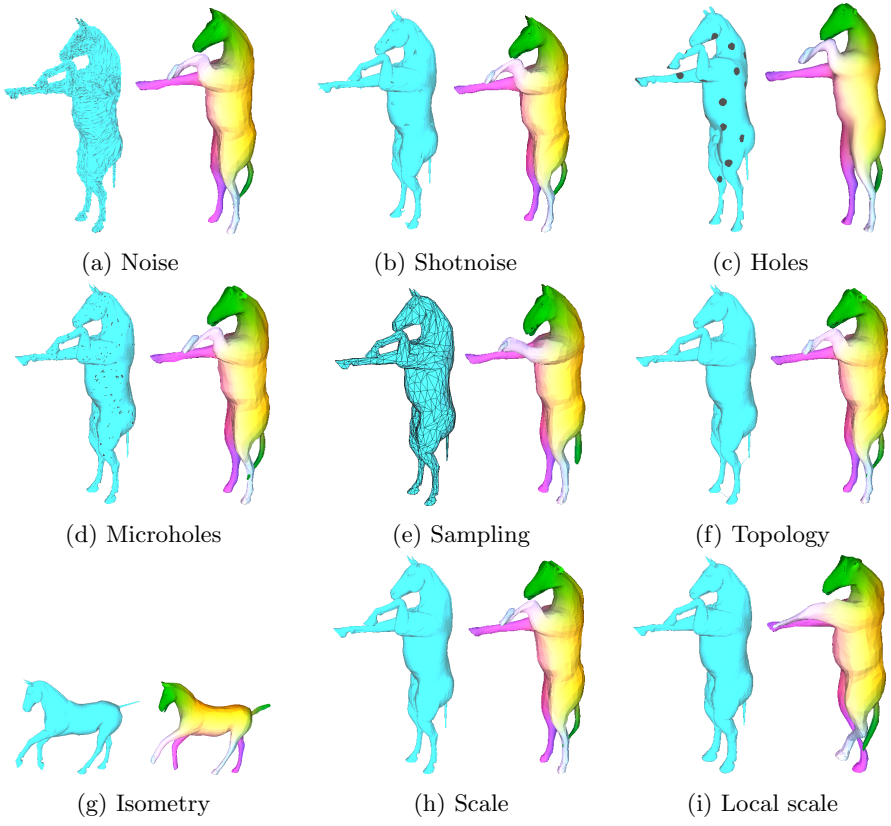
Perturbation		Error (cm)	Perturbation		Error (cm)	Perturbation		Error (cm)
Noise	1	4.58	Scale	1	4.73	Holes	1	4.71
	2	3.87		2	4.78		2	4.71
	3	3.93		3	4.66		3	4.72
	4	3.67		4	4.62		4	4.69
	5	3.91		5	4.67		5	4.84
ShotNoise	1	4.66	Local scale	1	4.18	Microholes	1	4.71
	2	2.64		2	3.65		2	4.72
	3	3.03		3	3.62		3	4.82
	4	2.72		4	3.75		4	4.69
	5	3.00		5	3.56		5	3.53
Sampling	1	4.82	Topology	1	3.99	Isometry	1	4.72
	2	4.78		2	4.38		2	4.69
	3	4.61		3	4.37		3	4.79
	4	3.72		4	4.31		4	4.85
	5	9.93		5	7.53		5	4.74

6.2 Cross-category correspondances on animals

SMAL synthetic are in correspondences across categories. Hence the template for two different categories are in correspondences and our approach can be trivially extended to get correspondences for animals from different species. Qualitative evidence of this is show in Figure 12.

6.3 Regularization for the unsupervised case

We observe some distortion when the network is trained using the Chamfer distance alone. For example the left foot is propagated on left hand in Figure 10. Even if this distortion is consistent across shapes, we hope that by regularizing the generator, the learned deformation on the template would respect the connectivity of the points of the templates. To achieve this, we tried two methods.



(a) Noise (b) Shotnoise (c) Holes

(d) Microholes (e) Sampling (f) Topology

(g) Isometry (h) Scale (i) Local scale

Fig. 11: **Robustness to perturbations on TOSCA for the horse category.** Correspondences are suggested by color. Notice the overall robustness to all perturbations, with small errors on the ears, tail or legs.

Ratio preservation Let (V, E) be the graph of the template and V^g the reconstructed vertices.

$$E_{ratio}(V^g) = \frac{1}{\#E} \cdot \sum_{i \sim j} \| \frac{V_i^g - V_j^g}{V_i - V_j} - 1 \|$$

This enforces edges to keep the same length in the template and the generated mesh. We use $\lambda_{ratio} = 0.005$. For instance, if the length of an edge doubles the contribution to the loss is $\lambda_{ratio} \cdot 1 = 0.005$ which is equivalent (in terms of contribution to the loss function) to a error of placement of 7.1cm. In other words, in terms on loss for the network, it is equivalent to double an edge's length or to misplace a point by 3.2cm.

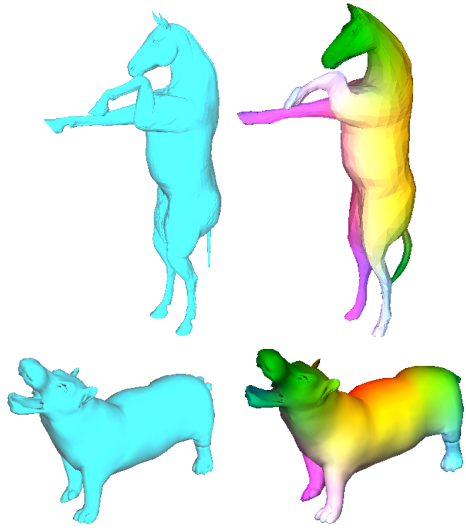


Fig. 12: **Inter-class correspondences on animals.** Correspondences are suggested by color.

Laplacian regularization Similar to Kanazawa et. al. [46], we use the Laplacian regularization. The Laplacian matrix L is defined as :

$$L_{i,j} = \begin{cases} d_i \text{ if } i = j \\ -1 \text{ if } (i,j) \in E \\ 0 \text{ otherwize} \end{cases}$$

$$LV_i = \sum_{i \sim j} V_i - V_j$$

$$LV_i = d_i \cdot (V_i - \frac{\sum_{i \sim j} V_j}{d_i})$$

This is an approximation of the following integral as explained in [47].

$$\lim_{\gamma \to 0} \frac{1}{\mid \gamma \mid} \int_{v \in \gamma} (v_i - v) dl(v) = -H(v_i) \cdot n_i$$

where:

- $H(v_i)$ is the mean curvature
- n_i is the surface normal

We follow [48] and use cotangent weights in the Laplacian to have better geometric discretization property.

$$L^c V_i = \frac{1}{\Omega_i} \sum_{i \sim j} \frac{1}{2} (cot\alpha_{ij} + cot\beta_{ij})(V_i - V_j)$$

where :

- Ω_i is the size of the Voronoi cell of i
- α_{ij} and β_{ij} denote the two angles opposite of edge (i, j)

Our Laplacian loss is thus written :

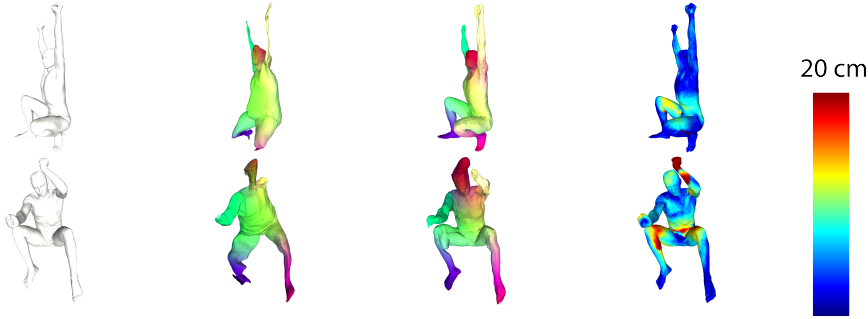
$$E_{laplace}(V^g) = \mathbb{1}^t \cdot L^c \cdot (V^{template} - V^g)$$

We use $\lambda_{laplace} = 0.005$. In practice we notice that using Laplacian regularization constrain the network to keep sound surfaces. It may still suffer from error in symmetry and can still invert right and left, and front and back.

6.4 Failure cases

Figure 13 shows the two main sources of error our algorithm faces. It can be an error in the nearest neighbor step in overlapping regions; here, a point is matched with the closest point in Euclidean distance but the match is very far in geodesic distance. This could be addressed by enforcing matches between the input mesh and its reconstruction in a way that takes into account the regularity of the matching. We leave this to future work.

The other source of error comes from failures in reconstruction: in such cases, the initial guess of the autoencoder is just too far away from the input, and the regression step fails.



(a) Input (b) Rec. 1 (c) Rec. 2. (d) Error.

Fig. 13: **Error visualization** Given the input mesh (a), our autoencoder makes an initial reconstruction (b), optimized by a regression step (c). The average in centimeters over each vertex of (a), of the Euclidean distance between its projection and the ground truth, is reported (d). We use the jet colormap. Red vertices have an error higher than 10, blue ones lower than 2cm. The largest error are observed in places where the Euclidean distance is small, while the geodesic distance is high, such as touching skin (zoom in on the leg). In such region, the nearest neighbors step is match a vertex in mesh A in a distant (in terms of geodesic distance) vertex in mesh A's reconstruction. High error can also come from bad reconstruction. See the head of the second example.