OperatorNet: Recovering 3D Shapes From Difference Operators

Ruqi Huang, Marie-Julie Rakotosaona, Panos Achlioptas, Leonidas Guibas, Maks Ovsjanikov

OperatorNet: Recovering 3D Shapes From Difference Operators

Ruqi Huang*
LIX, Ecole Polytechnique
rqhuang88@gmail.com

Marie-Julie Rakotosaona*
LIX, Ecole Polytechnique
mrakotos@lix.polytechnique.fr

Panos Achlioptas
Stanford University
optas@cs.stanford.edu

Leonidas Guibas
Stanford University
guibas@cs.stanford.edu

Maks Ovsjanikov
LIX, Ecole Polytechnique
maks@lix.polytechnique.fr

Abstract

This paper proposes a learning-based framework for reconstructing 3D shapes from functional operators, compactly encoded as small-sized matrices. To this end we introduce a novel neural architecture, called OperatorNet, *which takes as input a set of linear operators representing a shape and produces its 3D embedding. We demonstrate that this approach significantly outperforms previous purely geometric methods for the same problem. Furthermore, we introduce a novel functional operator, which encodes the extrinsic or pose-dependent shape information, and thus complements purely intrinsic pose-oblivious operators, such as the classical Laplacian. Coupled with this novel operator, our reconstruction network achieves very high reconstruction accuracy, even in the presence of incomplete information about a shape, given a soft or functional map expressed in a reduced basis. Finally, we demonstrate that the multiplicative* functional algebra *enjoyed by these operators can be used to synthesize entirely new unseen shapes, in the context of shape interpolation and shape analogy applications.*

1. Introduction

Encoding and reconstructing 3D shapes is a fundamental problem in Computer Graphics, Computer Vision and related fields. Unlike images, which enjoy a canonical representation, 3D shapes are encoded through a large variety of representations, such as point clouds, triangle meshes and volumetric data, to name a few. Perhaps even more importantly, 3D shapes may undergo a diverse set of transformations, ranging from rigid motions to complex non-rigid and articulated deformations, that impact these representations.

The representation issues have become even more prominent with the recent advent of learning-based techniques, leading to a number of solutions for learning directly on geometric 3D data [7]. This is challenging, as point clouds and meshes lack the regular grid structure exploited by convolutional architectures. In particular, devising representations that are well-adapted for both shape analysis and especially *shape synthesis* remains difficult. For example, several methods for shape interpolation have been proposed by designing deep neural networks, including auto-encoder architectures, and interpolating the latent vectors learned by such networks [35, 1] . Unfortunately, it is not clear if the latent vectors lie in a linear vector space, and thus linear interpolation can lead to unrealistic intermediate shapes.

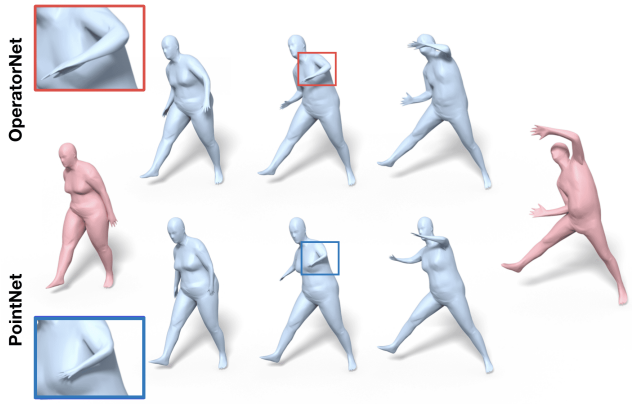

Figure 1. Shape interpolation via OperatorNet (top) and PointNet autoencoder (bottom). Our interpolations are more smooth and less distorted.

In this paper, we show that 3D shapes can not only be compactly encoded as linear functional operators, using the previously proposed shape difference operators [32], but that this representation lends itself very naturally to learning, and allows us to *recover* the 3D shape information, using a novel neural network architecture which we call OperatorNet. Our key observations are twofold: first we show that since shape difference operators can be stored as canonical matrices, for a given choice of basis, they enable the use

*denotes equal contribution.

of a convolutional neural network architecture for shape recovery. Second, we demonstrate that the *functional algebra* that is naturally available on these operators can be used to synthesize new shapes, in the context of shape interpolation and shape analogy applications. We argue that because this algebra is well-justified theoretically, it also leads to more accurate results in practice, compared to commonly used linear interpolation in the latent space (see Figure 1).

The shape difference operators introduced in [32], have proved to be a powerful tool in shape analysis, by allowing to characterize each shape in a collection as the "difference" to some base geometry. These difference operators encode precise information about how and where each shape differs from the base, but also, due to their compact representation as small matrices, enable efficient exploration of *global* variability within the collection. Inspired by the former perspectives, purely geometric approaches [5, 10] have been proposed for shape reconstruction from shape differences. Though theoretically well-justified, these approaches rely on solving difficult non-linear optimization problems and require strong regularization for accurate results, especially when truncated bases are used.

Our OperatorNet, on the other hand, leverages the information encoded at both the pairwise level and the collection level by using the shape collection to guide the reconstruction. It is well-known that related shapes in a collection often concentrate near a low-dimensional manifold in shape space [33, 19]. In light of this, the shape difference operators can help to both encode the geometry of the individual shapes, but also help to learn the constrained space of realistic shapes, which is typically ignored by purely geometric approaches. Finally, they also allow to encode differences between shapes with different discretizations by relying on *functional* maps, rather than, e.g., pointwise bijections.

In addition to demonstrating the representative power of the shape differences in a learning framework, we also extend the original formulation in [32], which only involves intrinsic (i.e., invariant to isometric transformations) shape differences, with a novel *extrinsic* difference operator that facilitates pose-dependent embedding recovery. Our formulation is both simpler and robuster compared to previous approaches, e.g. [10], and, as we show below, can more naturally be integrated in a unified learning framework.

To summarize, our contributions are as follows:

- We propose a learning-based pipeline to reconstruct 3D shapes from a set of difference operators.
- We propose a novel formulation of extrinsic shape difference, which complements the intrinsic operators formulated in [32].
- We demonstrate that by applying algebraic operations on shape differences, we can synthesize new operators and thus new shapes via OperatorNet, enabling shape manipulations such as interpolation and analogy.

2. Related Work

Shape Reconstruction Our work is closely related to shape reconstruction from intrinsic operators, which was recently considered in [5, 10] where several advanced, purely geometric optimization techniques have been proposed that give satisfactory results in the presence of full information [5] or under strong regularization [10]. These works have also laid the theoretical foundation for shape recovery by demonstrating that shape difference operators, in principle, contain complete information necessary for recovering the shape embedding (e.g. Propositions 2 and 4 in [10]). On the other hand, these methods also highlight the practical challenges of reconstructing a shape without any knowledge of the collection or "shape space" that it belongs to. In contrast, we show that by leveraging such information via a learning-based approach, realistic 3D shapes can be recovered efficiently from their shape difference representation, and moreover that entirely new shapes can be synthesized using the algebraic structure of difference operators, e.g., for shape interpolation.

Shape Representations for Learning. Our work is related to the recent techniques aimed at applying deep learning methods to shape analysis. One of the main challenges is defining a meaningful notion on convolution, while ensuring invariance to basic transformations, such as rotations and translations. Several techniques have been proposed based on e.g., Geometry Images [34], volumetric [22, 38], point-based [28] and multi-view approaches [29], as well as, very recently intrinsic techniques that adapt convolution to curved surfaces [21, 6, 27] (see also [7] for an overview), and even via toric covers [20], among many others.

Despite this tremendous progress in the last few years, defining a shape representation that is compact, lends itself naturally to learning, while being invariant to the desired class of transformations (e.g., rigid motions) and not limited to a particular topology, remains a challenge. As we show below, our representation is well-suited for learning applications, and especially for encoding and recovering geometric structure information. We note that a recent work that is closely related to ours is the characteristic shape differences proposed in [14]. That work is primarily focused on *analyzing* shape collections, rather than on shape *synthesis* that we target.

Shape Space Exploring the structure of shape spaces has a long and extensive research history. Classical PCA-based models, e.g. [2, 13], and more recent shape space models, adapted to specific shape classes such as humans [19] or animals [39], or parametric model collections [33], all typically leverage the fact that the space of "realistic" shapes is significantly smaller than the space of all possible embeddings. This has also recently been exploited in the context

of learning-based shape synthesis applications for shape completion [17], interpolation [3] and point cloud reconstruction [1] among others. These techniques heavily leverage the recent proliferation of large data collections such as DFAUST [4] and Shapenet [8] to name a few. At the same time, it is not clear if, for example, the commonly used *linear interpolation* of latent vectors is well-justified, leading to unrealistic synthesized shapes. Instead, the shape difference operators that we use satisfy a well-founded multiplicative algebra, which, as we show below, can be used to create realistic synthetic shapes.

3. Preliminaries and Notations

Discretization of Shapes Throughout this paper, we assume that a shape is given as a triangle mesh $(\mathcal{V}, \mathcal{F})$, where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is the vertex set, and $\mathcal{F} = \{(v_i, v_j, v_k) | v_i, v_j, v_k \in \mathcal{V}\}$ is the set of faces encoding the connectivity information.

Laplace-Beltrami Operator To each shape S, we associate a discretized Laplace-Beltrami operator, $\mathcal{L} := A^{-1}W$, using the standard cotangent weight scheme [23, 26], where W is the cotangent weight (stiffness) matrix, and A is the diagonal lumped area (mass) matrix. Furthermore, we denote by $\mathbf{\Lambda}, \mathbf{\Phi}$, respectively the diagonal matrix containing the k smallest eigenvalues and the corresponding eigenvectors of S, such that $W\mathbf{\Phi} = A\mathbf{\Phi}\mathbf{\Lambda}$. In particular, the eigenvalues stored in $\mathbf{\Lambda}$ are non-negative and can be ordered as $0 = \lambda_1 \leq \lambda_2 \leq \cdots$. The columns of $\mathbf{\Phi}$ are sorted accordingly, and are orthonormal with respect to the area matrix, i.e., $\mathbf{\Phi}^T A \mathbf{\Phi} = \mathbf{I}_{k \times k}$, the $k \times k$ identity matrix. It is well-known that Laplace-Beltrami eigenbasis provides a multi-scale encoding of a shape [16], and allows to approximate the space of functions via a subspace spanned by the first few eigenvectors of $\mathbf{\Phi}$.

Functional Maps The functional map framework was introduced in [24] primarily as an alternative representation of maps across shapes. In our context, given two shapes S_0, S_1 and a point-wise map T from S_1 to S_0, we can express the functional map $\mathbf{C}_{01}$ from S_0 to S_1, as follows:

$$\mathbf{C}_{01} = \mathbf{\Phi}_1^T A_1 \Pi_{01} \mathbf{\Phi}_0. \tag{1}$$

Here, A_1 is the area matrix of S_1, and Π_{01} is a binary matrix satisfying $\Pi_{01}(p, q) = 1$ if $T(p) = q$ and 0 otherwise. Note that $\mathbf{C}_{01}$ is a $k_1 \times k_0$ matrix, where k_1, k_0 is the number of basis functions chosen on S_1 and S_0. This matrix allows to transport functions as follows: if f is a function on S_0 expressed as a vector of coefficients $\mathbf{a}$, s.t. $f = \mathbf{\Phi}_0 \mathbf{a}$, then $\mathbf{C}_{01}\mathbf{a}$ is the vector of coefficients of the corresponding function on S_1, expressed in the basis of $\mathbf{\Phi}_1$.

In general, not every functional map matrix arises from a point-wise map, and the former might include, for example, soft correspondences, which map a point to a probability density function. All of the tools that we develop below can accommodate such general maps. This is a key advantage of our approach, as it does not rely on all shapes having the same number of points, and only requires the knowledge of functional map matrices, which can be computed using existing techniques [25, 18].

Intrinsic Shape Difference Operators Finally, to represent shapes themselves, we use the notion of shape difference operators proposed in [32]. Within our setting, they can be summarized as follows: given a base shape S_0, an arbitrary shape S_i and a functional map $\mathbf{C}_{0i}$ between them, let $\mathbf{K}_0$ (resp. $\mathbf{K}_i$) be a positive semi-definite matrix, which defines some inner product for functions on S_0 (resp. S_i) expressed in the corresponding bases. Thus, for a pair of functions f, g on S_0 expressed as vectors of coefficients $\mathbf{a}, \mathbf{b}$, we have $< f, g >= \mathbf{a}^T \mathbf{K}_0 \mathbf{b}$.

Note that these two inner products $\mathbf{K}_0, \mathbf{K}_i$ are not comparable, since they are expressed in different bases. Fortunately, the functional map $\mathbf{C}_{0i}$ plays a role of basis synchronizer. Thus, a shape difference operator, which captures the difference between S_0 and S_i is given simply as:

$$\mathbf{D}_{0i}^K = \mathbf{K}_0^+ (\mathbf{C}_{0i}^T \mathbf{K}_i \mathbf{C}_{0i}), \tag{2}$$

where $^+$ is the Moore-Penrose pseudo-inverse.

The original work [32] considered two intrinsic inner products, which using the notation above, can be expressed as: $\mathbf{K}^{L^2} = \mathbf{Id}$, and $\mathbf{K}^{H^1} = \mathbf{\Lambda}$. These inner products, in turn lead to the following shape differences operators:

$$\text{Area-based } (L^2): \quad \mathbf{D}_{0i}^A = \mathbf{C}_{0i}^T \mathbf{C}_{0i}, \tag{3}$$

$$\text{Conformal } (H^1): \quad \mathbf{D}_{0i}^C = \Lambda_0^+ \mathbf{C}_{0i}^T \Lambda_i \mathbf{C}_{0i}, \tag{4}$$

These shape difference operators have several key properties. First, they allow to represent an arbitrary shape S_i, as a pair of matrices of size $k_0 \times k_0$, independent of the number of points, by requiring only a functional map between the base shape S_0 and S_i. Thus, the size of this representation can be controlled by choosing an appropriate value of k_0 which allows to gain multi-scale information about the geometry of S_i, from the point of view of S_0. Second, and perhaps more importantly, these matrices are invariant to rigid (and indeed any intrinsic isometry) transformation of S_0 or S_i. Finally, previous works [10] have shown that shape differences *in principle* contain complete information about the intrinsic geometry of a shape. As we show below these properties naturally enable the use of learning applications for shape recovery.

Functoriality of Shape Differences Another useful property of the shape difference operators is *functoriality*, shown

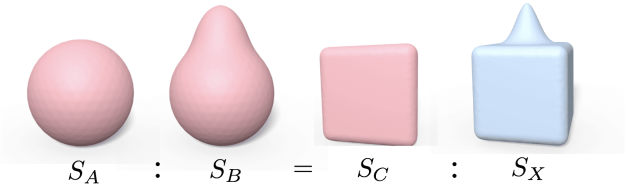

Figure 2. Illustration of shape analogy.

in [32], and which we exploit in our shape synthesis applications in Section 7. Given shape differences $\mathbf{D}_{0i}, \mathbf{D}_{0j}$ of shapes S_i and S_j with respect to a base shape S_0, functoriality allows to compute the difference $\mathbf{D}_{ij}$, without functional maps between S_i and S_j. Namely (see Prop. 4.2.4 in [9]):

$$\mathbf{D}_{ij} = \mathbf{C}_{0i}\mathbf{D}_{0i}^{+}\mathbf{D}_{0j}\mathbf{C}_{0i}^{-1} \tag{5}$$

Intuitively, this means that shape differences naturally satisfy the *multiplicative algebra*: $\mathbf{D}_{0i}\mathbf{D}_{ij} = \mathbf{D}_{0j}$, up to a change of basis ensured by $\mathbf{C}_{0i}$.

This property can be used for *shape analogies*: given shapes S_A, S_B and S_C, find S_X such that S_X relates to S_C in the same way as S_B relates to S_A (see the illustration in Figure 2). This can be solved by looking for a shape X that satisfies: $\mathbf{C}_{0C}^{+}\mathbf{D}_{CX}\mathbf{C}_{0C} = \mathbf{C}_{0A}^{+}\mathbf{D}_{AB}\mathbf{C}_{0A}$. In our application, we first create an appropriate $\mathbf{D}_{0X}$ and then use our network to synthesize the corresponding shape.

Finally, the multiplicative property also suggests a way of interpolation in the space of shape differences. Namely, rather than using basic linear interpolation between $\mathbf{D}_{0i}$ and $\mathbf{D}_{0j}$, we interpolate on the Lie algebra of the Lie group of shape differences, using the exponential map and its inverse, which leads to:

$$\mathbf{D}(t) = \exp((1-t)\log(\mathbf{D}_{0i}) + t\log(\mathbf{D}_{0j})), t \in [0,1]. \tag{6}$$

Here exp and log are matrix exponential and logarithm respectively. Note that, around identity, the linearization provided by the Lie algebra is exact, and we have observed it to produce very accurate results in general.

4. Extrinsic Shape Difference

In our (discrete) setting, with purely *intrinsic* information one at the best can determine the edge lengths of the mesh. Recovering the shape from its edge lengths, while possible in certain simple scenarios, nevertheless often leads to ambiguities, as highlighted in [10]. To alleviate such ambiguities, we propose to augment the existing intrinsic shape differences with a novel *extrinsic* shape difference operator, and in turn boosts our reconstruction.

One basic approach to combine extrinsic information with the multi-scale Laplace-Beltrami basis is to project the 3D coordinate functions onto the basis, to obtain three vectors of coefficients (one for each x, y, z coordinates):

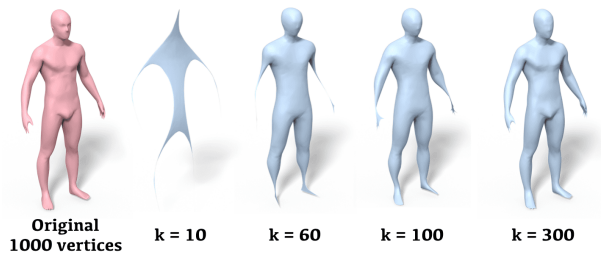

Figure 3. From left to right: original shape with 1000 vertices, the recovered embedding from $\mathbf{G}$ encoded in the leading k = 10, 60, 100 and 300 eigenbasis of the original shape.

$\mathbf{f} = \mathbf{\Phi}^{+}X$, where X is the $n_V \times 3$ matrix of vertex coordinates [16, 15]. Unfortunately representing a shape through $\mathbf{f}$, though being multi-scale and compact, is not rotationally invariant, and does not provide information about intrinsic geometry. For example, interpolation of coordinate vectors can easily lead to loss of shape area.

Another option, which is more compatible with our approach and is rotationally invariant, is to encode the inner products of coordinate functions on each shape using the Gram matrix $G = XX^T$. Expressing G in the corresponding basis, and using Eq. (2) gives rise to a shape difference-like representation of the coordinates. Indeed, the following theorem (see proof in the supplementary materials) guarantees that the resulting representation contains the same information, up to rotational invariance, as simply projecting the coordinates onto the basis.

Theorem 1. *Let $\mathbf{G} = \mathbf{\Phi}^T A X X^T A \mathbf{\Phi}$ be the extrinsic inner product encoded in Φ, then one can recover the projections of the coordinate functions, X, on the subspace spanned by $\mathbf{\Phi}$ from $\mathbf{G}$, up to a rigid transformation. In particular, when $\mathbf{\Phi}$ is a complete full basis, the recovery of X is exact.*

As an illustration of Theorem 1, we show in Figure 3 the embeddings recovered from $\mathbf{G}$ when the number of basis functions in $\mathbf{\Phi}$ ranges from 10 to 300.

However, the rank of the Gram matrix $\mathbf{G}$ of a shape is at most 3, meaning that the majority of its eigenvalues are zero. This turns out to be an issue in applications, where gaining information about the *local* geometry of the shape is important, for example in our shape analogies experiments.

To compensate for this rank deficiency, we make the extrinsic inner product Laplacian-like:

$$E^D(i,j) = \begin{cases} -E(i,j) & \text{if } i \neq j, \\ \sum_{i \neq j} E(i,j) & \text{i = j.} \end{cases} \tag{7}$$

Where $E(i,j)$ is $\|v_i - v_j\|^2 A(i,i)A(j,j)$, i.e., the squared Euclidean distance between points v_i, v_j on the shape, weighted by the respective vertex area measures. Since E^D can be regarded as the Laplacian of a complete graph, all but one of its eigenvalues are strictly positive.

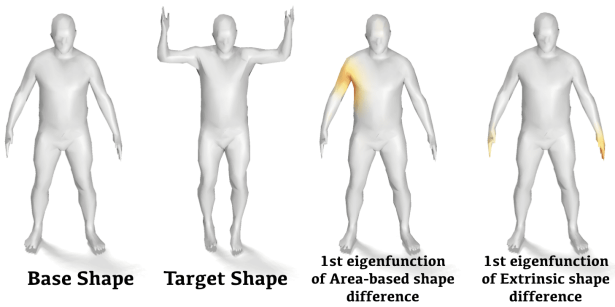

Figure 4. A pair of shapes are compared. The most area (resp. extrinsic) distorted region is captured by the leading eigenfunction of the area-based (resp. extrinsic) shape difference.

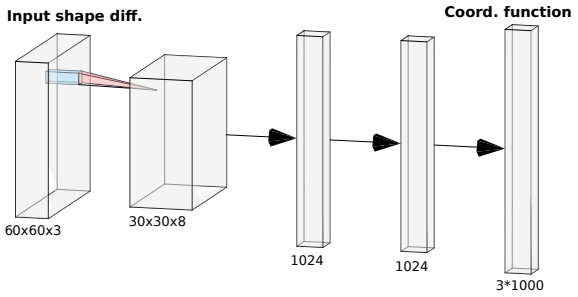

Figure 5. OperatorNet architecture. The inputs of the network are shape difference matrices considered as channels. It outputs the coordinate functions of the shape. The first part (left) of the network consists of a convolutional encoder while the second part (right) is a fully-connected decoder built with dense layers.

It is worth noting that the Gram matrix and the squared Euclidean distance matrix are closely related and can be recovered from each other as is commonly done in the Multi-Dimensional Scaling literature [11].

To summarize, given a base shape S_0, another shape S_i and a functional map $\mathbf{C}_{0i}$ we encode the extrinsic information of S_i from the point of view of S_0 as follows:

$$\mathbf{D}_i^E = (\mathbf{\Phi}_0^T E_0^D \mathbf{\Phi}_0)^+ (\mathbf{C}_{0i}^T \mathbf{\Phi}_i^T E_i^D \mathbf{\Phi}_i \mathbf{C}_{0i}). \quad (8)$$

In Figure 4, we compute $\mathbf{D}^A$ and $\mathbf{D}^E$ of the target shape with respect to the base, and color code their respective eigenfunctions associated with the largest eigenvalue on the shapes to the right. As argued in [32] these functions capture the areas of highest distortion between the shapes, with respect to the corresponding inner products. Note that the eigenfunction of $\mathbf{D}^A$ captures the armpit where the local area shrinks significantly, while that of $\mathbf{D}^E$ captures the hand, where the *pose changes* are evident.

Note that in [10], the authors also propose a shape difference formulation for encoding extrinsic information, which is defined on the shape offset using the surface normal information. However, their construction can lead to instabilities, and moreover, it only gives information about *local* distances, making it hard to recover large changes in pose.

5. Network Details

Problem Setup Our general goal is to develop a neural network capable of recovering the coordinates of a shape, given its representation as a set of shape difference matrices. We therefore aim to solve the same problem considered in [5, 10]. However, unlike these purely geometric methods, we also leverage a collection of training shapes to learn and constrain the reconstruction to the space of realistic shapes.

Thus, we assume that we are given a collection of shapes, each represented by a set of shape difference operators with respect to a fixed base shape. We also assume the presence of a point-wise map from the base shape to each of the shapes in the collection, which allows us to compute the "ground truth" embedding of each shape. We represent this embedding as three coordinate functions on the shape. Our goal then is to design a network, capable of converting the input shape difference operators to the ground truth coordinate functions.

At test time, we use this network to reconstruct a target shape given only the shape difference operators with respect to the base shape. Importantly, these shape difference operators only require the knowledge of a *functional map* from the base shape, and can thus arise from shapes with different discretizations, or can be synthesized directly for shape analogies or interpolations applications.

Architecture To solve the problem above we developed the OperatorNet architecture, which takes as input shape difference matrices and outputs coordinate functions. Our network has two modules: a shallow convolutional encoder and a 3 layer dense decoder as shown in Figure 5.

The grid structure of shape differences is exploited by the encoder through the use of convolutions. Note however that translation invariance does not apply to these matrices.

After comparing multiple depths of encoders, we select a shallow version as it performs the best in practice, implying that the shape difference representation already encodes meaningful information efficiently. Moreover, as shown in [10] the edge lengths of a mesh can be recovered from intrinsic shape differences through a series of least squares problems, hinting that increasing the depth of the network and thus the non-linearity might not be necessary with shape differences.

On the other hand, the decoder is selected for its ability to transform the latent representation to coordinate functions for reconstruction and synthesis tasks.

Datasets We train OperatorNet on two types of datasets: humans and animals. For human shapes, our training set consists of 9440 shapes sampled from the DFAUST dataset [4] and 8000 from the SURREAL dataset [37], which is generated with the model proposed in [19]. The

DFAUST dataset contains scan of human characters subject to a various of motions. On the other hand, the SURREAL dataset injects more variability to the body types.

For animals, we use the parametric model proposed in SMAL [39] to generate 1800 animals of 3 different species – lions, dogs, and horses. The meshes of the humans (resp. animals) are simplified to 1000 vertices (resp. 1769 vertices).

Input Shape Differences We construct the input shape differences using a truncated eigenbasis of dimension 60 on the base shape, and the full basis on the target one, in all experiments, regardless of the number of vertices on the shapes. The functional maps from the base to the targets are induced by the identity maps, since our training shapes are in 1-1 correspondence. This implies that each of the shapes is represented by three 60×60 matrices, representing the area-based, conformal and extrinsic shape differences respectively. The independence among the shape differences allows flexibility in selecting the combination of input shape differences, in Section 6 we compare the performance of several combinations, and present a more detailed ablation study in the supplementary materials.

It is worth noting that recent learning-based shape matching techniques enable efficient (functional) maps estimation. In particular, we use the unsupervised matching method of [31] and evaluate OperatorNet trained with *computed* shape differences in Section 6.

Loss Function OperatorNet reconstructs coordinate functions of a given training shape. Our shape reconstruction loss operates in two steps. First, we estimate the optimal rigid transformation to align the ground truth point cloud X_{gt} and the reconstructed point cloud X_{recon} using the Kabsch algorithm [36] with ground truth correspondences. Secondly, we estimate the mean squared error between the aligned reconstruction and the ground truth.

$$L(X_{\text{gt}}, X_{\text{recon}}) = \frac{1}{n_V} \sum_{i=1}^{n_V} \|R(X^i_{\text{recon}}) - X^i_{\text{gt}}\|^2. \quad (9)$$

Here R is the function that computes the optimal transformation between X_{recon} and X_{gt}. We align the computed reconstruction to the ground truth embedding, so that the quality of the reconstructed point cloud is invariant to rigid transformations. This is important since the shape difference operators are invariant to rigid motion of the shape, and thus the network should not be penalized, for not recovering the correct orientation. On the other hand, this loss function is differentiable, since we use a closed-form expression of $R_{X_{\text{gt}}}$, given by the SVD, which enables back-propagation in neural network training.

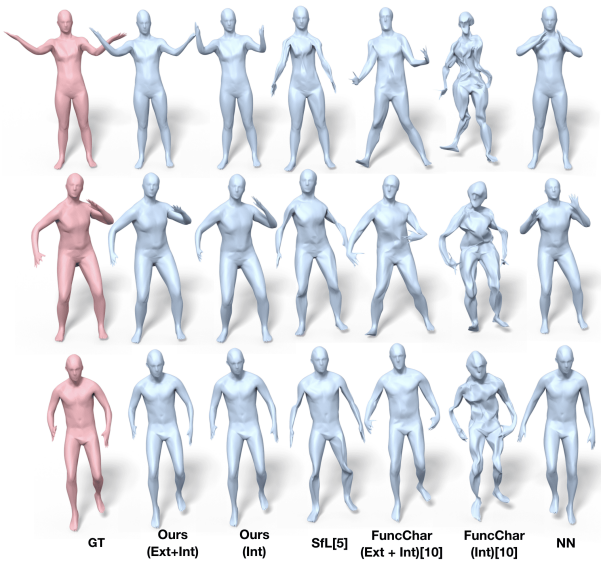

Figure 6. Qualitative comparison of our method and the baselines.

6. Evaluation

In this section, we provide both qualitative and quantitative evaluations of the results from OperatorNet, and compare them to the geometric baselines.

Evaluation Metrics We denote by S_{gt} and S_{recon} the ground-truth and the reconstructed meshes respectively. We let $d_R = L(X_{\text{gt}}, X_{\text{recon}})$, where L is the rotationally-invariant distance defined in Eq. (9) and X is the vertex set of S. Since OperatorNet is trained with the loss defined in Eq. (9), we introduce the following new metrics for a comprehensive, unbiased evaluation and comparison: (1) $d_V = |V(S_{\text{gt}}) - V(S_{\text{recon}})|/V(S_{\text{gt}})$, i.e., the relative error of mesh volumes; (3) $d_E = \text{mean}_{(i,j)}|l^{\text{gt}}_{ij} - l^{\text{recon}}_{ij}|/l^{\text{gt}}_{ij}$, where l_{ij} is the length of edge (i, j).

Baselines Two major baselines are considered: (1) the intrinsic reconstruction method from [5], in which we evaluate with the 'Shape-from-Laplacian' option and use the full basis in *both* the base shape and the target shape; (2) the reconstruction method from [10], where the authors construct offset surfaces that also capture extrinsic geometry. Moreover, this method also provides a purely intrinsic reconstruction version. We evaluate both cases with the same basis truncation as our input. Beyond that, we also consider the nearest neighbor retrieval from the training set with respect to distances between shape difference matrices.

Test Data We use 800 shapes from the DFAUST dataset as the test set, which contains 10 sub-collections (character + action sequence, each consisting of 80 shapes) that are isolated from the training/validation set. For the efficiency

of baseline evaluation, we further sample 5 shapes via furthest point sampling regarding the pair-wise Hausdorff distance from each of the sub-collection, resulting in a set of 50 shapes that covers significant variability in both styles and poses in the test set.

Qualitative Results We demonstrate the reconstructed shapes from OperatorNet and the aforementioned baselines in Figure 6, where the red shape in each row is the ground truth target shape. The base shape in this experiment (also the base shape we compute shape differences on) is shown in Figure 4, which is in the rest pose. The geometric baselines in general perform worse under significant pose changes from the base (see the top two rows in Figure 6), but give relatively more stable results when the difference is mainly in the shape style (see the bottom row).

Our method, on the other hand, produces consistently good reconstructions in all cases. Note also that, as expected, OperatorNet using all 3 types of shape differences gives both the best quantitative and qualitative results. We provide more reconstruction examples in the supplementary materials highlighting the generalization power of our method.

Quantitative Results We report all the quantitative metrics defined above in Table 1. First, we observe that OperatorNet using both intrinsic and extrinsic shape differences achieves the lowest reconstruction error, while the purely extrinsic version is the second best. Secondly, OperatorNet trained on shape differences from computed functional maps achieves competing performances, showing that our method is efficient even in the absence of ground truth bijective correspondences. Lastly, all the versions of OperatorNet significantly outperform the baselines.

Regarding the volume and edge recovery accuracy, either complete or intrinsic-only versions of OperatorNet achieve second to the best result. We remark that since the nearest neighbor search in general retrieves the right body type, therefore the volume is well-recovered. On the other hand, since the *full* Laplacian is provided as input for the Shape-from-Laplacian baseline, it is expected to preserve intrinsic information.

Reconstructions of Shapes with Different Discretizations Lastly, we show that our approach is capable of encoding differences between shapes with different discretizations. In Figure 7, we compute the functional maps from the fine meshes (top row, with 5k vertices) by projecting them to a lower resolution base mesh with 1k vertices. We then reconstruct them with OperatorNet trained on lower resolution shapes. This, on the other hand, is extremely difficult for purely geometric methods. In the supplementary materials we provide examples of reconstructions in the same setting using the method of [10], and reconstructions with OperatorNet trained with shapes having 2k vertices.

Table 1. Quantitative evaluation of shape reconstruction (d_R is at the scale of 10^{-4}).

	d_R	d_V	d_E
Op.Net (Int+Ext)	**1.11**	0.014	0.045
Op.Net (Int)	2.41	0.013	0.046
Op.Net (Ext)	1.25	0.017	0.046
Op.Net (Comp)(Ext)	3.86	0.021	0.052
Op.Net (Comp)(Int+Ext)	6.22	0.022	0.053
SfL [5]	48.8	0.081	**0.012**
FuncChar [10](Int)	65.1	0.356	0.118
FuncChar [10] (Int+Ext)	28.4	0.028	0.110
NN	25.5	**0.005**	0.043

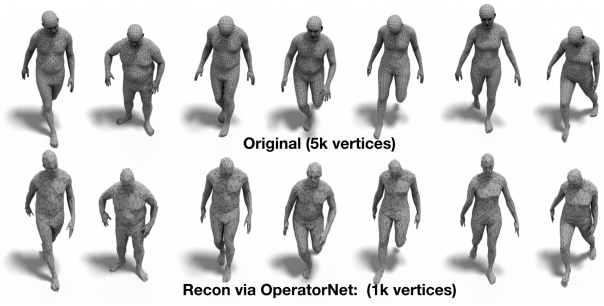

Figure 7. Top row: input shapes with different number of vertices than that of the base shape; Bottom row: reconstructions via OperatorNet.

7. Applications

In this section, we present all of our results using OperatorNet trained with all 3 types of shape differences.

Shape Interpolation Given two shapes, we first interpolate their shape differences using the formulation in Eq.(8), and then synthesize intermediate shapes by inferring the interpolated shape differences with OperatorNet.

We compare our method against nearest neighbor retrieval and PointNet autoencoder. PointNet autoencoder is trained with the encoder architecture from [28] and with our decoder. Two versions of PointNet are trained: one autoencoder with spatial transformers and one without. Since the autoencoder without spatial transformers performs better in our experiments, we select it for the comparisons. Nearest neighbor interpolation retrieves the nearest neighbor of the interpolated shape differences in the training set and uses the corresponding embedding. As expected, (see the second row of Figure 9), nearest neighbor interpolation is less continuous.

As shown in Figure 1, our method produces smooth interpolations, without significant local area distortions compared to PointNet. Similarly, in Figure 9, we observe that the interpolation via PointNet suffers from local distortion on the arms. In contrast, interpolation using OperatorNet is continuous and respects the structure and constraints of the body, suggesting that shape differences efficiently en-

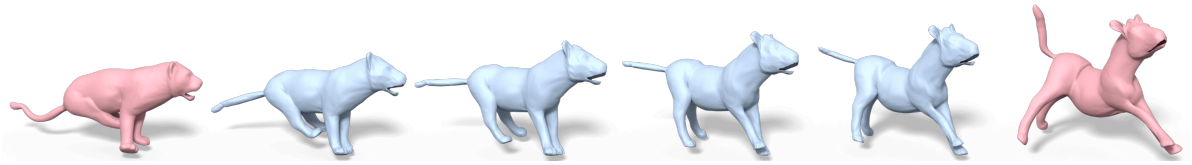

Figure 8. Shape interpolation from a tiger (left) to a horse (right) using OperatorNet trained on animals dataset.

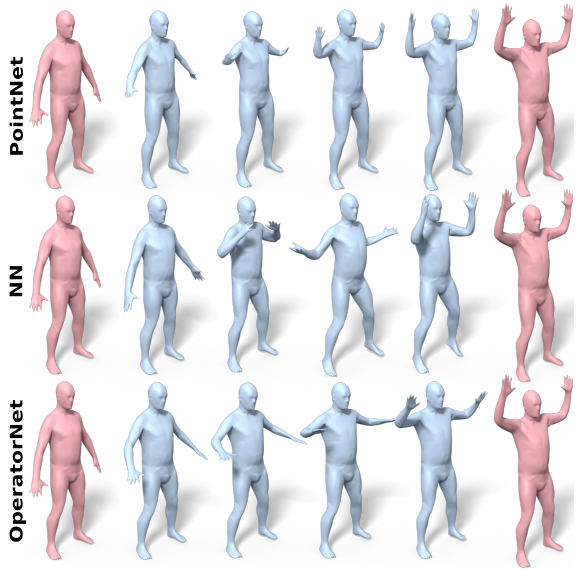

Figure 9. Shape interpolation between two humans. Note that PointNet autoencoder produces shapes with local area distortion, while the interpolation from nearest neighbor (NN) retrieval is not continuous.

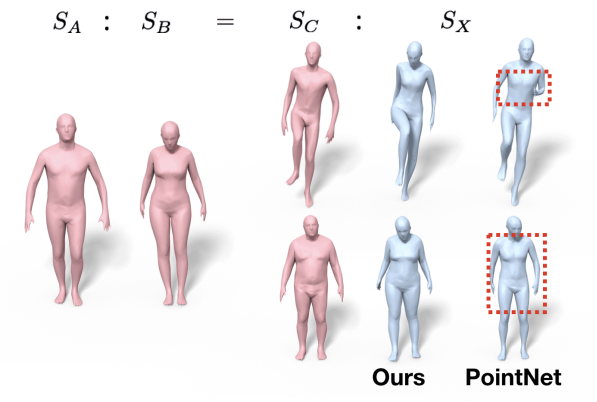

Figure 10. Transferring gender via shape analogies: S_A and S_B are a fixed pair of human shapes with similar poses and styles, but of different genders. We generate S_X, which is supposed to be a "female" version of the varying S_C. Our analogies are semantically meaningful, while PointNet can produce suboptimal results (see the red dotted boxes for the discrepancies).

code the shape structure. We provide further comparisons to other baselines including [30, 3, 12] and to linear interpolation of shape differences in the supplementary materials.

We also train OperatorNet on the animals dataset as described in Section 5 and show in Figure 8 an interpolation from a tiger to a horse.

Shape Analogy Our second application is to construct semantically meaningful new shapes based on shape analogies. Given shapes S_A, S_B, S_C, our goal is to construct a new shape S_X, such that S_C relates to S_X as S_A to S_B.

Following the discussion in Section 3, the functoriality of shape differences allows an explicit and mathematically meaningful way of constructing the shape difference of S_X, given that of S_A, S_B and S_C. Namely, $\mathbf{D}_X = \mathbf{D}_C \mathbf{D}_A^+ \mathbf{D}_B$. Then, with our OperatorNet, we reconstruct the embedding of the unknown S_X by feeding $\mathbf{D}_X$ to the network.

We compare our results to that of the PointNet autoencoder. In the latter, we reconstruct S_X by decoding the latent code obtained by $l_X = l_C - l_A + l_B$, where l_A is the latent code of shape S_A (and similarly for S_B, S_C).

In Figure 10, we show a set of shape analogies obtained via OperatorNet and PointNet autoencoder. It is evident that our results are both more natural and intuitive. We also refer the readers to the supplementary materials for more examples of analogies.

8. Conclusion & Future Work

In this paper we have introduced a novel learning-based technique for recovering shapes from their difference operators. Our key observation is that shape differences, stored as compact matrices lend themselves naturally to learning and allow to both recover the underlying shape space in a collection and encode the geometry of individual shapes. We also introduce a novel extrinsic shape difference operator and show its utility for shape reconstruction and other applications such as shape interpolation and analogies.

Currently our approach is only well-adapted to shapes represented as triangle meshes. Thus, in the future we plan to extend this framework to both learn the optimal inner products from data, and adapt our pipeline to other shape representations, such as point clouds or triangle soups.

Acknowledgements Parts of this work were supported by the ERC Starting Grant StG-2017-758800 (EXPROTEA), KAUST OSR Award CRG-2017-3426 a gift from the Nvidia Corporation, a Vannevar Bush Faculty Fellowship, NSF grant DMS-1546206, a Google Research award and gifts from Adobe and Autodesk. The authors thank Davide Boscaini and Etienne Corman for their help with baseline comparisons.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 1, 3

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. 2

[3] Heli Ben-Hamu, Haggai Maron, Itay Kezurer, Gal Avineri, and Yaron Lipman. Multi-chart generative surface modeling. In *Proc. SIGGRAPH Asia*, page 215. ACM, 2018. 3, 8

[4] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *CVPR*, July 2017. 3, 5

[5] Davide Boscaini, Davide Eynard, Drosos Kourounis, and Michael M Bronstein. Shape-from-operator: Recovering shapes from intrinsic operators. In *Computer Graphics Forum*, volume 34, pages 265–274. Wiley Online Library, 2015. 2, 5, 6, 7

[6] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016. 2

[7] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 1, 2

[8] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 3

[9] Etienne Corman. *Functional representation of deformable surfaces for geometry processing*. PhD thesis, 2016. PhD thesis. 4

[10] Etienne Corman, Justin Solomon, Mirela Ben-Chen, Leonidas Guibas, and Maks Ovsjanikov. Functional characterization of intrinsic and extrinsic geometry. *ACM Trans. Graph.*, 36(2):14:1–14:17, Mar. 2017. 2, 3, 4, 5, 6, 7

[11] Trevor F. Cox and M.A.A. Cox. *Multidimensional Scaling, Second Edition*. Chapman and Hall/CRC, 2 edition, 2000. 5

[12] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3d-coded : 3d correspondences by deep deformation. In *ECCV*, 2018. 8

[13] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and H-P Seidel. A statistical model of human pose and body shape. In *Computer graphics forum*, volume 28, pages 337–346. Wiley Online Library, 2009. 2

[14] Ruqi Huang, Panos Achlioptas, Leonidas Guibas, and Maks Ovsjanikov. Limit Shapes - A Tool for Understanding Shape Differences and Variability in 3D Model Collections. *Computer Graphics Forum*, 2019. 2

[15] Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448. Wiley Online Library, 2013. 4

[16] Bruno Levy. Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13, June 2006. 3, 4

[17] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *Proc. CVPR*, pages 1886–1895, 2018. 3

[18] Or Litany, Tal Remez, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proc. ICCV*, pages 5659–5667, 2017. 3

[19] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, Oct. 2015. 2, 5

[20] Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, VLADIMIR G KIM, and Yaron Lipman. Convolutional neural networks on surfaces via seamless toric covers. 2017. 2

[21] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proc. ICCV workshops*, pages 37–45, 2015. 2

[22] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015. 2

[23] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. 3

[24] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012. 3

[25] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, 2017. 3

[26] Ulrich Pinkall and Konrad Polthier. Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics*, 2(1):15–36, 1993. 3

[27] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):236:1–236:14, 2018. 2

[28] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. 2, 7

[29] 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 *Proc. CVPR*, pages 5648–5656, 2016. 2

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. 8

[31] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. *CoRR*, abs/1812.03794, 2018. 6

[32] Raif M. Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)*, 32(4):1, 2013. 1, 2, 3, 4, 5

[33] Adriana Schulz, Ariel Shamir, Ilya Baran, David I. W. Levin, Pitchaya Sitthi-Amorn, and Wojciech Matusik. Retrieval on parametric shape collections. *ACM Trans. Graph.*, 36(4), Jan. 2017. 2

[34] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3d shape surfaces using geometry images. In *European Conference on Computer Vision*, pages 223–240. Springer, 2016. 2

[35] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[36] Arun K. Somani, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987. 6

[37] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 5

[38] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72, 2017. 2

[39] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, July 2017. 2, 6