Deformable Shape Completion with Graph Convolutional Autoencoders

Or Litany[1,2], Alex Bronstein[1,3], Michael Bronstein[4], Ameesh Makadia[2]
[1]Tel Aviv University [2]Google Research [3]Technion [4]USI Lugano

Abstract

The availability of affordable and portable depth sensors has made scanning objects and people simpler than ever. However, dealing with occlusions and missing parts is still a significant challenge. The problem of reconstructing a (possibly non-rigidly moving) 3D object from a single or multiple partial scans has received increasing attention in recent years. In this work, we propose a novel learning-based method for the completion of partial shapes. Unlike the majority of existing approaches, our method focuses on objects that can undergo non-rigid deformations. The core of our method is a variational autoencoder with graph convolutional operations that learns a latent space for complete realistic shapes. At inference, we optimize to find the representation in this latent space that best fits the generated shape to the known partial input. The completed shape exhibits a realistic appearance on the unknown part. We show promising results towards the completion of synthetic and real scans of human body and face meshes exhibiting different styles of articulation and partiality.

1. Introduction

The problem of reconstructing 3D shapes from partial observations is central to a broad spectrum of applications, ranging from virtual and augmented reality to robotics and autonomous navigation. Of particular interest is the setting where objects may undergo articulations or more generally non-rigid deformations. While several methods based on (volumetric) convolutional neural networks have been proposed for completing man-made rigid objects (see [12, 51, 55, 61, 49]), they struggle at handling deformable shapes. However, this is not a limitation specific to volumetric approaches. The same difficulties with deformable shapes, irrespective of the completion task, are present for other 3D shape representations utilized in deep learning frameworks, such as view-based [52, 59] and point clouds [42, 43].

The main reason for this is that for methods based on Euclidean convolutional operations (e.g. volumetric or view-based deep neural networks), an assumption of self-similarity under rigid transformations (in most cases, axis-aligned) is implied. For example a chair seat will always be parallel to the floor. Non-rigid deformations violate this assumption, effectively making each pose a novel object. Thus, tackling such data with a standard CNN requires many network parameters and a prohibitively large amount of training. Although model-based methods such as [2] have shown good performance, they are restricted to a specific class of shape with manually constructed models.

To explicitly enable robustness towards non-rigid deformations, the approach advocated in this paper adopts recent advances for in CNNs on graphs which directly exploit the 3D mesh structure. This allows the learning of a powerful non-rigid shape representation from data without an explicit model.

Another shortcoming of deep learning shape completion techniques stems from their end-to-end design. A network trained to perform completion would be biased towards the type of missing data introduced at training, and may not generalize well to previously unseen types of missing information. To allow generalization to any style of partiality we choose to separate the task of completion from the training procedure altogether. As a result, we also avoid a significant amount of preprocessing and augmentation that is typically done on the training data.

Finally, when a complete mesh is desired as the output, producing a triangulation from point clouds or volumetric grids is itself a challenging problem and may introduce undesired artifacts (although recent advances such as [12] address this by directly producing implicit surfaces). Conversely, by utilizing a mesh-convolutional network our method will produce complete and plausible surfaces by design.

Contribution. The main contribution of this work is a method for deformable shape completion that decouples the task of partial shape completion from the task of learning a generative shape model, for which we introduce a novel graph convolutional autoencoder architecture. Compared to previous works the proposed method has several advantages. First, it can handle any style of partiality without needing to see any partial shapes during training. Second,

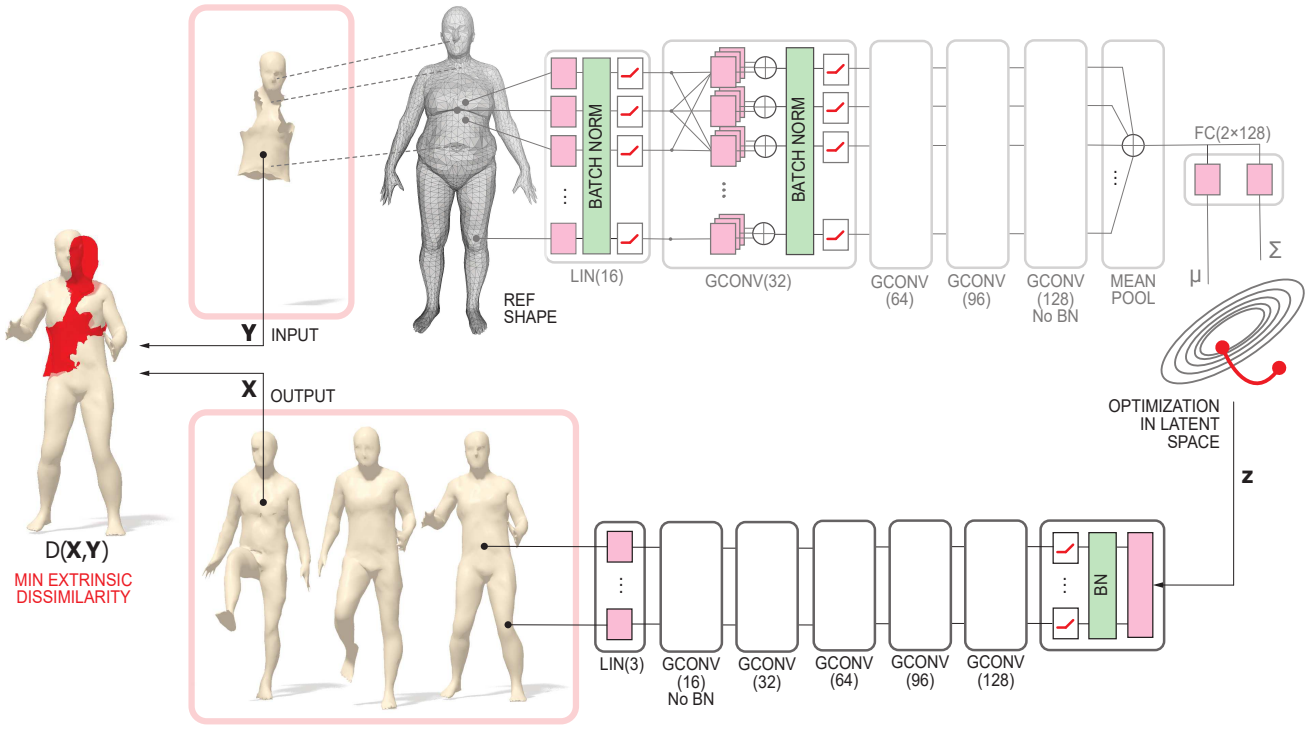

Figure 1. **Schematic description of our approach.** A variational autoencoder is first trained on full shapes with vertex-wise correspondence to create a reference shape and a latent space parameterizing the embedding of its vertices in $\mathbb{R}^3$. At inference, only the decoder (bottom part) is used. A partially missing shape is given as the input together with the correspondence with the reference shape. Starting at a random initialization, optimization is performed in the latent space to minimize the extrinsic dissimilarity between the input shape and the generated output shape.

the method is not limited to a specific class of shapes (e.g. humans) and can be applied to any kind of 3D data. Third, shape completion is an inherently ill-posed problem with potentially multiple valid solutions fitting the data (this is especially true for articulated and deformable shapes), thus making deterministic solutions inadequate. The proposed method reflects the inherent ambiguities of the problem by producing multiple plausible solutions.

2. Related work

3D shape completion. The application addressed in this paper is a very active research area in computer vision and graphics, ranging from completion of small holes [48] and larger missing regions in individual objects [1, 45, 55, 61, 49], to entire scenes [51]. Completion guided by geometric priors has been explored, for example Poisson filling [22] and self-similarity [27, 48, 32]. However, such methods work only for small missing regions, and dealing with bigger occlusions requires stronger priors. A viable alternative is model-based approaches, where a parametric morphable model describing the variability of a certain class of objects can be fit to the observed data [4, 16].

The setting of non-rigid shape completion differs from its rigid counterpart in that at inference, the input partial shape may admit a deformation unseen in the training data. This distinction becomes crucial as large missing regions force the priors to become more complex (see for example the human model designed in [2]).

Generative methods for non-rigid shapes. The state-of-the-art in generative modeling has rapidly advanced with the introduction of Variational Autoencoders (VAE [25]), Generative Adversarial Networks [18], and related variations (e.g. VAEGAN [29]). These advances have been adopted by the 3D shape analysis community for dynamic surface generation through VAE [28] and image-to-shape generation through VAEGAN [60]. In [53], a VAE for non-rigid shapes is proposed. This work differs from ours in that the core operations of our network are graph-convolutional operations as opposed to fully-connected layers, and our network operates directly on raw 3D vertex positions rather than relying on hand-crafted features.

Geometric deep learning. This paper is closely related to a broad area of active research in geometric deep learning (see [9] for a summary). The success of deep learning

(in particular, convolutional architectures [30]) in computer vision has brought a keen interest in the computer graphics community to replicate this progress for applications dealing with geometric 3D data. One of the key difficulties is that for such data it requires great care to define the basic operations constituting deep neural networks, such as convolution and pooling.

Several works avoid this problem by using a Euclidean representation of 3D shapes, such as rendering a collection of 2D views [52, 59], volumetric representations [61], or point cloud [42, 43]. One of the main drawbacks of such extrinsic deep learning methods is their difficulty to deal with shape deformations as discussed earlier. Additionally, voxel representations are often memory intensive and suffer from poor resolution [61], although recent models have been proposed to address these issues: implicit surface representation [12], sparse octree networks [57, 44], encoder-decoder CNN for patch-level geometry refinement [19], and a long-term recurrent CNN for upsampling coarse shapes [58]. Regarding point cloud representations, the PointNet model [42] applies identical operations to the coordinates of each point and aggregates this local information without allowing for interaction between different points which makes it difficult to capture local surface properties. PointNet++ [43] addresses this by proposing a spatially hierarchical model. Additionally, for PointNet to be invariant to rigid transformations the input point clouds are aligned to a canonical space. This is achieved by a small network that predicts the appropriate affine transformation, but in general such an alignment would be difficult for articulated and deformable shapes.

An alternative strategy is to redefine the basic ingredients of deep neural networks in a geometrically meaningful or intrinsic manner. The first intrinsic CNN-type architectures for 3D shapes were based on local charting techniques generalizing the notion of "patches" to non-Euclidean and irregularly-sampled domains [35, 7, 36]. The key advantage of this approach is that the generalized convolution operations are defined intrinsically on the manifold, and thus automatically invariant to its isometric deformations. As a result, intrinsic CNNs are capable of achieving correspondence results with significantly less parameters and a very small training set. Related independent efforts developed CNN-type architectures for general graphs [10, 20, 13, 26, 36, 31].

Recently, [56] suggested a dynamic filter in which the assignment of each filter to each member of the k-ring in a graph neighborhood is determined by its feature values. Importantly, this method demonstrated state-of-the-art performance working directly on the embedding features. Thus, in our work we build upon [56] as a basic building block for convolution operations.

Partial shape correspondences. Dense non-rigid shape correspondence [23, 11, 33, 47, 8] is a fundamental challenge as it is an enabler for many high level tasks like pose or texture transfer across surfaces. We refer the interested reader to [54, 3] for a detailed review of the literature. The proposed method in this work builds upon correspondence between a partial input and a canonical shape of the same class, and related to this are several methods that explore partial shape correspondence and matching [46, 36, 34]. The approaches demonstrating state-of-the-art performance on partial human shapes (e.g. [36]) treat correspondence as a vertex classification task. Recently [59] has shown impressive results for correspondence across different human subjects in varied pose and clothing.

Inpainting. The 3D shape completion task is closely related to the analogous structured prediction task of image inpainting [41, 62]. However, our proposed optimization scheme is more reminiscent of style transfer [15] techniques. In our setting we optimize only for the best complete shape with no constraints on the internal feature representation.

3. Method

We propose a shape completion method that detaches the process of learning to generate 3D shapes from the task of partial shape completion. Our method requires a generative model for complete 3D shapes which we construct by training a graph-convolutional variational autoencoder (VAE [25]). Partial shapes can be completed by identifying the shape in the output space of the VAE's generator which best aligns with the partial input. We propose an optimization in the latent space that iteratively deforms (non-rigidly) a randomly generated shape to align with a partial input. In what follows, we describe in more detail both ingredients of our process, the VAE generator and the partial shape completion scheme. A schematic rendition of the method is depicted in Figure 1.

3D shape generator. We fix the number of vertices N and the topology of a reference shape and refer to the three-dimensional vertex embedding $\mathbf{X} \in \mathbb{R}^{3\times N}$ as to a shape. The VAE consists of two networks: the *encoder* that encodes 3D shape inputs $\mathbf{X}$ to a latent representation vector $\mathbf{z} = \mathrm{enc}(\mathbf{x})$, and the *decoder* that decodes the latent vectors into 3D shapes $\mathbf{X}' = \mathrm{dec}(\mathbf{z})$. The variational distribution $q(\mathbf{z}|\mathbf{X})$ is associated with a prior distribution over the latent variables, and the usual choice which we follow here is a centered multivariate Gaussian with unit variance $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Our VAE loss combines the shape reconstruction loss $L_r = ||\mathrm{dec} \circ \mathrm{enc}(\mathbf{X}) - \mathbf{X}||_2$ encouraging the encoder-decoder pair to be a nearly identity transformation, and a

regularization prior loss measured by the Kullback-Leibler divergence, $L_p = D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X})||p(\mathbf{z}))$. The total VAE loss is computed as $L = L_r + \lambda L_p$, where $\lambda \geq 0$ controls the similarity of the variational distribution to the prior.

The choice to measure shape reconstruction loss with pointwise distances is not the only option. For example, the VAE can be combined with a Generative Adversarial Network (VAE-GAN) as in [29, 60], thus introducing an additional discriminator loss on the reconstructed shape. We do not consider a discriminator in the scope of this work to avoid additional model complexity but leave it as future work to investigate different loss functions that can be imposed on reconstructed shapes.

The internal details of the VAE encoder $\mathrm{enc}(\mathbf{X})$ and decoder $\mathrm{dec}(\mathbf{z})$ are largely influenced by the choice of the 3D shape representation. As discussed in Section 2, many representations have been explored ranging from voxels to raw point clouds. Our desire to focus on shape completion for deformable object classes leads us to consider intrinsic mesh and surface models that have shown promising results for deformable shape correspondence among other applications (e.g. [35, 36]). Multiple approaches have been proposed to perform convolution on spatial meshes. The primary factor which distinguishes spatial graph convolutional operations is how correspondence is determined between convolutional filters and the local graph neighborhoods. Rather than relying on properties of the underlying geometry to map filters to surface patches, we adopt data-adaptive models which learn the mapping from the neighborhood patch to filters weights. Specifically, our VAE is primarily composed of the dynamic filtering convolutional layers proposed in FeaStNet [56]. The input to the layer is a feature vector field on the mesh vertices, attaching to a vertex i a vector $\mathbf{x}_i$. The output is also a vector field $\mathbf{y}_i$, possibly of a different dimension, computed as

$$\mathbf{y}_i = \mathbf{b} + \sum_{m=1}^{M} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} q_m(\mathbf{x}_i, \mathbf{x}_j) \mathbf{W}_m \mathbf{x}_j, \qquad (1)$$

where $\mathcal{N}_i$ denotes a patch around the vertex i, and $q_m(\mathbf{x}_i, \mathbf{x}_j) \propto \exp(\mathbf{u}_m^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j) + c_m)$ are positive edge weights in the patch normalized to sum to one over m. The trainable weights of the layer are $\mathbf{W}_m$, $\mathbf{u}_m$, c_m and $\mathbf{b}$, while the number of weight matrices M is a fixed design parameter. Note that the mapping from neighborhood patch to weights is translation invariant in the input feature space, as q operates only on the differences $\mathbf{x}_i - \mathbf{x}_j$. Refer to Figure 1 and [56] for further details.

Partial shape completion. Once the encoder-decoder pair has been trained, the encoder is essentially tossed away, while the decoder acts as a complete shape generator, associating to each input latent vector $\mathbf{z}$ an $\mathbb{R}^3$ embedding of the reference shape, $\mathbf{X} = \mathrm{dec}(\mathbf{z})$. Importantly, this acts as a strong shape prior, generating plausible looking shapes (see Figure 2).

At inference, a partial shape $\mathbf{Y}$ is given. We first use an off-the-shelf method (MoNet) [36] to compute a dense partial intrinsic correspondence between $\mathbf{Y}$ and the reference shape. Representing this correspondence as a partial permutation matrix $\mathbf{\Pi}$ and applying it to any shape $\mathbf{X}$ generated by the decoder produces a subset of points in $\mathbb{R}^3$, $\mathbf{X\Pi}$), ordered compatibly with their counterparts in $\mathbf{Y}$. We therefore define an extrinsic dissimilarity between the input shape and the generated full shape as $\mathrm{D}(\mathbf{X}, \mathbf{Y}) = \|\mathbf{X} - \mathbf{Y}\|$, possibly weighed by the confidence of the correspondence at each point.

Inference consists essentially of finding a latent vector $\mathbf{z}^*$ minimizing the dissimilarity between the input and the output shape,

$$\min_{\mathbf{z}, \mathbf{T} \in \mathrm{SE}(3)} \mathrm{D}(\mathrm{dec}(\mathbf{z})\mathbf{\Pi}, \mathbf{TY}), \qquad (2)$$

where $\mathbf{T}$ denotes a rigid transformation. Alternating steps are performed over $\mathbf{z}$ (non-rigid deformation) and $\mathbf{T}$ (rigid registration). When the ℓ_2 norm is used to define the shape dissimilarity, the rigid registration step has a closed-form solution via the singular value decomposition of the covariance matrix of $\mathbf{Y}$ and $\mathbf{X\Pi}$, while the non-rigid deformation step is performed using stochastic gradient descent.

Shape completion is an inherently ill-posed problem that can have multiple plausible solutions. In cases where there exists more than one solution consistent with the data, sampling a result from our proposed generative model allows us to explore this space. The results in Section 4.2 illustrate the variability in completed shapes when repeating the optimization procedure (2) with random initializations.

4. Experiments

Dataset. The majority of our experiments are performed on human shapes. The VAE is trained on registered 4D scans from the DFAUST dataset [6] comprising 10 human subjects performing 14 different activities. Scans are captured at a high frame rate and registered to a canonical topology. Due to the high frame rate, we subsample the data temporally by a factor of 4. We consistently subsample each mesh by the factor of 2 down to $N = 3446$ vertices. Refer to the supplemental info for details on the data processing. The training set is created by holding out all scans for two human subjects and two activities leaving approximately 7000 training shapes. Details for additional experiments with face meshes is provided in Section 4.6.

Network parameters. The structure of our graph-convolutional VAE is illustrated in Figure 1. We evaluated a number of model parameters on a subset of the training set

to inform our final design choices. We use $M = 8$ and latent dimensionality of 128 for all our DFAUST experiments. A more important and delicate decision is the selection of the parameter λ controlling the emphasis on pushing the variational latent distribution towards the Gaussian prior. Our experiments show, as expected, that a higher weight for the Gaussian prior causes randomly sampled latent vectors to generate realistic shapes more likely, while a lower λ improves reconstruction accuracy over a wider variety of shapes. In the context of our problem, it is more important for the latent space to represent and for the decoder to be able to generate a wide variety of shapes accurately. Sampling from the latent space is less important since the final latent vectors are obtained by means of solving the optimization problem (2). Consequently, we selected $\lambda = 10^{-8}$ at training (see supplemental material for empirical analysis motivating these choices).

Implementation details. We train the model directly on the $3 \times N$ input meshes from the DFAUST dataset as described above; the sparse adjacency matrices (we use a vertex 2-ring as the neighborhood size) are passed as side information to define the graph convolutional layers. Data are augmented by adding normally distributed noise to the vertex positions as well as a global planar translations and scalings. We use the ADAM [24] optimizer with the learning rate set to 10^{-4}, momentum to 0.9, batch size to 2, Xavier initialization [17] for all weights, and train for 3×10^5 iterations. For shape completion optimization we use an SGD optimizer with a 0.1 learning rate. All additional data for training and evaluation will be provided on the authors' websites.

4.1. Representation quality

To understand the generative capabilities of the VAE we show several examples related to shape generation as well as explore the structure of the learned latent space. Figure 2 depicts shapes generated by the decoder fed with latent variables randomly sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. As we have explicitly relaxed the Gaussian prior on the latent variables (small λ) during training. As discussed earlier, the tradeoff is that samples coming from the prior may generate slightly unrealistic shapes.

Figure 3 depicts generated shapes as the result of linear interpolation in the latent space. The source and target shapes are first passed through the encoder to obtain latent representations; applying the decoder to convex combinations of these latent vectors produces a highly nonlinear interpolation in $\mathbb{R}^3$. The top two rows of Figure 3 show interpolation for networks trained with $\lambda = 10^{-6}$ and $\lambda = 10^{-8}$, respectively. The bottom row of Figure 3 highlights the interesting structure of the learned latent space through arithmetic. Applying the difference of a subject with left knee raised and lowered to the same subject with right knee raised results in a lowering of the right knee. The network learned this symmetry without any explicit modeling.

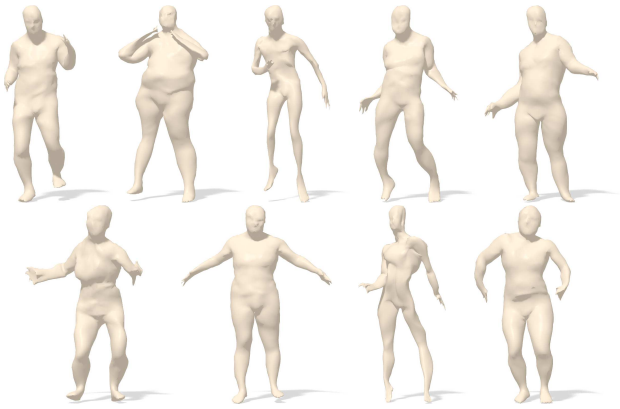

Figure 2. **Random human shapes generated by the VAE.** We have explicitly relaxed the Gaussian prior on the latent variables during training. The tradeoff is that samples coming from the prior may generate slightly unrealistic shapes.

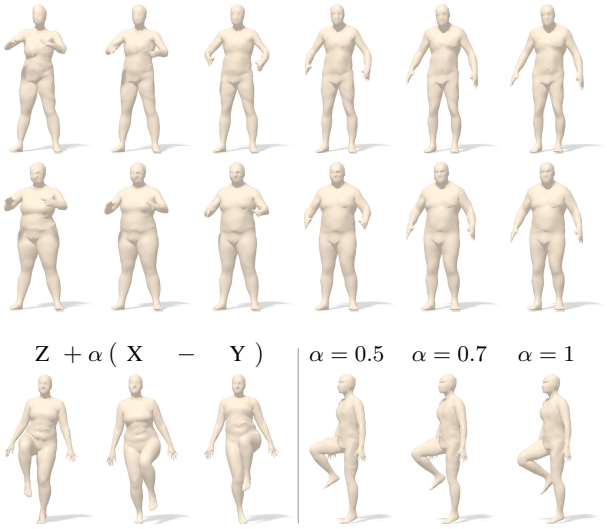

Figure 3. **Latent space interpolation.** Interpolation between two poses (left- and right-most shapes) obtained as convex combinations of the respective representations the the latent space. Bottom row: latent space arithmetic.

4.2. Completion variability

As explained in Section 3, given a partial input with more than one solution consistent with the data, we may explore this space of completions by sampling the initialization of problem 2 at random from the Gaussian prior. For evaluation we consider several test subjects with removed limbs. Figure 4 shows unique plausible completions of the same partial input achieved by random initializations.

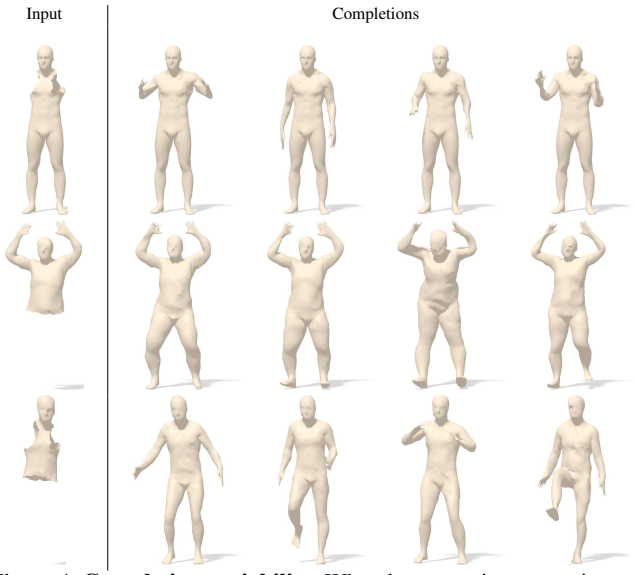

Figure 4. **Completion variability.** When large contiguous regions (e.g. limbs) are missing, the solution to shape completion is not unique. Shown here are different reconstructions with our method obtained using random initializations.

4.3. Synthetic range scans completion

The following experiment considers the common practical scenario of range scan completion. We utilize a test-set of 200 virtual scans produced from 10 viewpoints around 2 human subjects exhibiting 10 different poses. The full shapes were taken from FAUST [5], and are completely disjoint from our train set, as they contain novel subjects and poses. Furthermore, the data is suitable for quantitative comparison as sufficient information is given in the partial shape to make the completion problem nearly deterministic. Keeping the ground truth correspondence from each view to the full shape, we report the mean completion error in table 1 as *Ours (ground truth)*. More interesting are the results of end-to-end completion using partial correspondence obtained by MoNet [36] (reported as *Ours (MoNet)*). For reference we report the performance of other shape completion methods: 3D-EPN [12] which has shown state-of-the-art performance for shape completion using volumetric networks, Poisson reconstruction [22], and nearest neighbor (NN). Note, in order to comply with the architecture of 3D-EPN, we also provide viewpoint information, which is unknown for our method. For NN the completion is considered to be the closest shape from the entire training using the ground truth correspondences. Results in table 1 show mean Euclidean distance (in cm) and relative volumetric error (in %) for the missing region. More results are shown in Figures 5 and 6.

Robust optimization. Our method is able to generalize well to partial shapes in poses unseen during training.

Error	Euclidean distance [cm]	Volumetric err. mean $\pm$ std [%]
Poisson [22]	7.3	24.8 ± 23.2
NN (ground truth)	5.4	34.01 ± 9.23
3D-EPN [12]	4.43	89.7 ± 33.8
Ours (MoNet)	3.40	12.51 ± 11.1
Ours (MoNet with ref. 300)	3.01	10.00 ± 8.83
Ours (MoNet with refinement)	**2.84**	$\mathbf{9.24 \pm 8.62}$
Ours (ground truth)	**2.51**	$\mathbf{7.48 \pm 5.64}$

Table 1. **Synthetic range scans completion.** Comparison of different methods with respect to errors in vertex position and shape volume. Our method is evaluated using ground truth and MoNet [36] correspondences, as well as with and without refinement (details in Section 4.3).

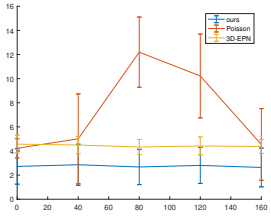

Figure 5. **Reconstruction error as a function of view angle.** Our method produces a consistently accurate reconstruction independent of the view angle.

However, there is a gap in performance when using correspondences from an oracle versus an off-the-shelf technique (MoNet). To handle noisy correspondences better, we propose a robust enhancement to (2). Observing that our method may not converge to the ideal completion if the alignment is guided by poor correspondences. However, if the partial shape is somewhat well aligned with the generated shape we can recalculate the correspondence $\mathbf{\Pi}$ using a simple Euclidean closest-vertex assignment. We find that recalculating when SGD plateaus leads to improved completions (results are reported in Table 1 as *Ours (MoNet with refinement)*). A pleasant side-effect of this refinement step is that our shape completion method can be used to obtain a de-noised (albeit sparser) set of correspondences (see the supplemental material for analysis). We also evaluate shape completion when the optimization steps are capped at 300 (reported as *Ours (MoNet with refinement* 300)) as opposed to running until convergence. Note, for simplicity we did not explore tuning different aspects of the method (learning rate, different reconstruction losses, etc).

4.4. Dynamic Fusion

A common use case of depth scanners is object reconstruction form multiple viewpoints. For static scenes, this problem was explored extensively, e.g., in [38, 39, 14]. Non-rigid deformations pose a much bigger challenge. The

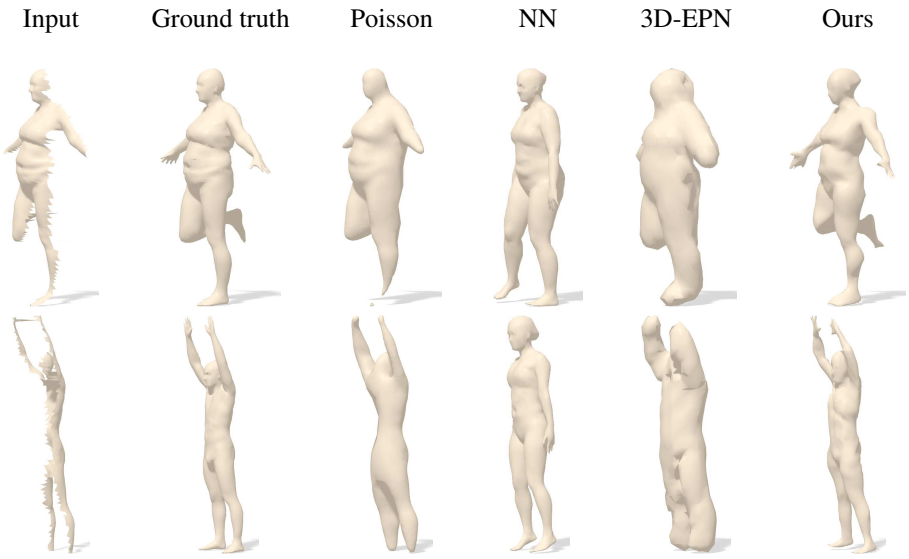

Figure 6. **Comparison of different synthetic range scan completion methods.** From left to right: input range scan, ground truth complete shape, Poisson reconstruction, 3D-EPN, and our method.

works of [37] and [21] have shown very impressive reconstructions, however they are limited to small motions between consecutive frames. This limitation was addressed in [50] by introducing a damped Killing motion constraint. These methods are focused on reconstructing only the observed dynamic surfaces and cannot convincingly hallucinate large unseen regions. Having developed a completion method for non-rigid shapes, we propose its extension to multiple partial inputs. Registering the partial inputs, or the individual reconstructed shapes, is challenging. Instead, we propose to merge shapes in the latent space: we obtain $\mathbf{z}$ by averaging the completed shape latent variables for each partial input, and $\mathrm{dec}(\mathbf{z})$ produces the fused 3D shape. Since the latent representation mixes body shape and pose, the reconstructed pose will generally not adhere to any of the input poses, but rather will be an interpolation thereof.

For a quantitative analysis, we perform fusion on three partial views from a static shape. We use the same FAUST shapes used for testing in Section 4.3. Table 2 shows mean reconstruction errors for all 20 test shapes when fusing three different partial views. The results show how reconstruction accuracy changes according to the viewpoint, and consistently improves with latent space fusion. A qualitative evaluation of the fusion problem is shown for the dynamic setting in Figure 7. Each row shows three partial views of the same human subject from a different viewpoint *and* a different pose. The latent space fusion of the completed shapes is shown in column 4.

4.5. Real range scan completion

The MHAD dataset [40] provides Kinect scans from 2 viewpoints of subjects performing a variety of actions. We apply our completion method to the extracted point cloud (correspondences were initialized through coarse alignment to a training shape, see the supplemental for details). Figure 8 depicts examples of scan completion on the Kinect data as well as on real scans from the DFAUST dataset.

View 1	View 2	View 3	Fused
2.78	2.94	2.93	**2.59**
3.03	3.39	2.73	**2.61**

Table 2. **Fusion in the latent space.** Reported is the mean Euclidean error in cm for three partial views (0^o, 120^o and 240^o for the first row and 80^o, 200^o and 320^o for the second row).

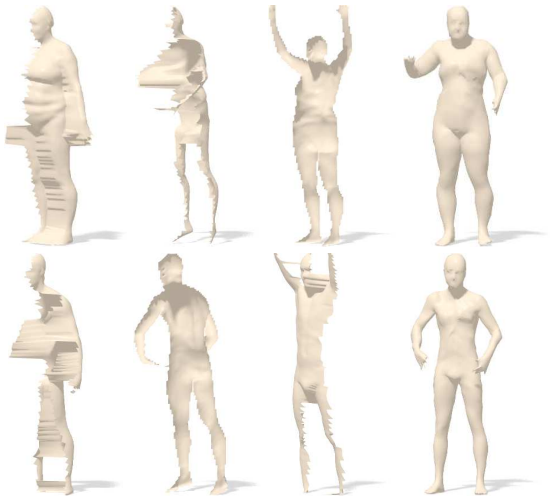

Figure 7. **Dynamic fusion.** Three partial views (columns 1-3) and the reconstructed complete shape (rightmost column).

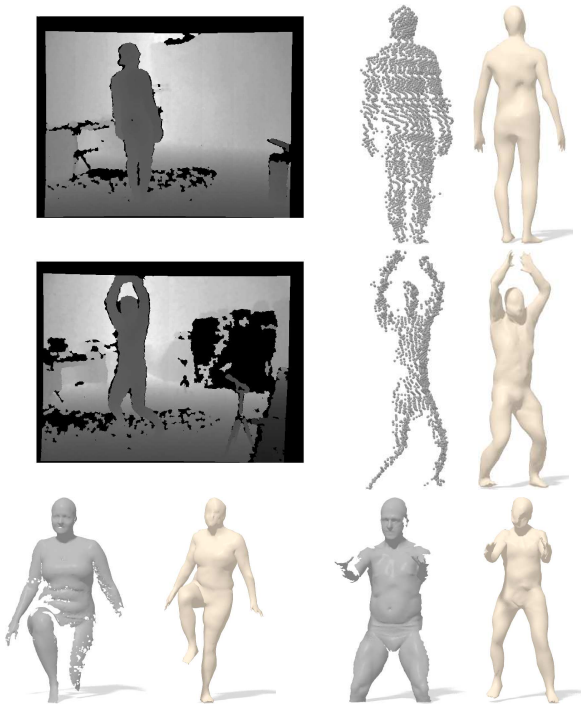

Figure 8. **Completion of real range scans.** First two rows: completion of Kinect scans (left: depth image; middle: extracted point cloud; right: completed shape). Last row: completion of scans from the DFAUST dataset.

4.6. Face completion

A strength of our fully data-driven approach is that by avoiding explicit shape modeling it generalizes easily to different classes of shapes. This is illustrated by an evaluation on deformable faces. 2000 training face meshes, each with 525 vertices, are generated from the model provided by [16]. These face models exhibit less variability in pose relative to the human meshes, so we use a much smaller VAE network (only two convolutional layers and a latent dimensionality of 32). Figure 9 shows completion for different styles of simulated partiality as well as simulated correspondence noise (see the supplemental for more details).

5. Conclusions and future work

This paper introduces a novel graph-convolutional method for shape completion. Its important properties include a model robust to non-rigid deformations, small sample complexity when training, and the ability to reconstruct any style of missing data. Evaluations indicate this is a promising first step towards shape completion from real-world scans, and the analysis reveals directions for future work. Firstly, exploring a representation that disentangles shape and pose would allow for more control in the completion and likely improve dynamic fusion results. Secondly, for initialization we require correspondences between the partial and canonical shape model. Although we show resilience to poor correspondences, improving this initialization for noisy real-world data would be beneficial. Finally, the proposed formulation assumes the desired shape topology (i.e. vertex connectivity) is known when decoding shapes. We leave to future work the task of completion with unknown topology.

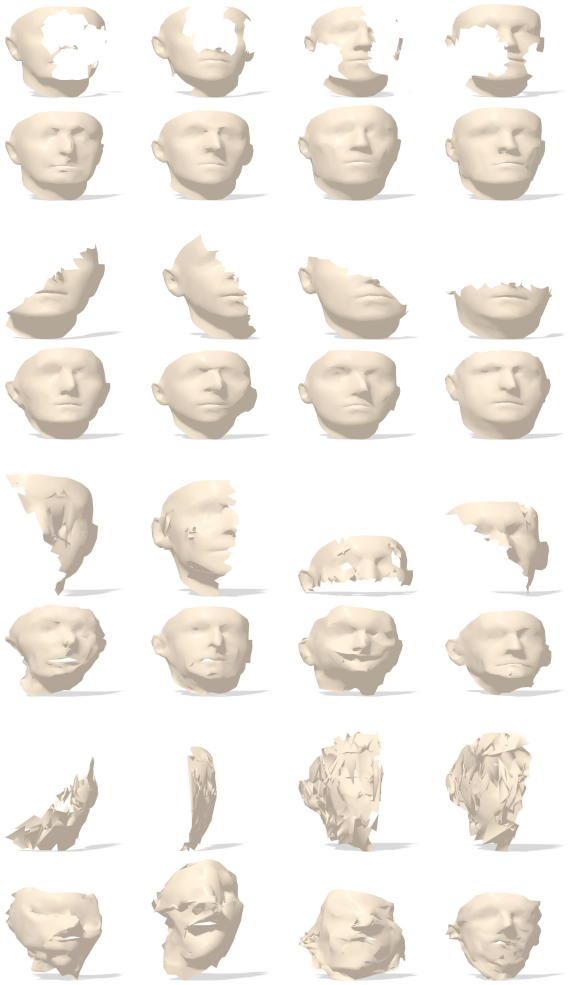

Figure 9. **Completion of faces.** Inputs and outputs are shown in the odd and even rows, respectively. Rows 1-2 show completion for missing patches, rows 3-4 show completion for hyperplane cuts, rows 5-6 show completion for hyperplane cuts and 5% correspondence error, and rows 7-8 show 30% correspondence error. Results indicate completion is plausible even under large missing regions and robust to reasonable correspondence error.

Acknowledgement

The authors wish to thank Emanuele Rodolà, Federico Monti, Vikas Sindhwani, and Leonidas Guibas for useful discussions. Much appreciated is the DFAUST scan data provided by Federica Bogo and Senya Polikovsky.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. J. Guibas. Representation learning and adversarial generation of 3D point clouds. *arXiv:1707.02392*, 2017. 2

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *TOG*, 24(3):408–416, 2005. 1, 2

[3] S. Biasotti, A. Cerri, A. M. Bronstein, and M. M. Bronstein. Recent trends, applications, and perspectives in 3D shape similarity assessment. In *Computer Graphics Forum*, 2015. 3

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. Computer Graphics and Interactive Techniques*, pages 187–194, 1999. 2

[5] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, 2014. 6

[6] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic FAUST: Registering human bodies in motion. In *Proc. CVPR*, July 2017. 4

[7] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Proc. NIPS*, 2016. 3

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006. 3

[9] M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond Euclidean data. *arXiv:1611.08097*, 2016. 2

[10] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv:1312.6203*, 2013. 3

[11] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *Proc. ICCV*, 2015. 3

[12] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3D-encoder-predictor cnns and shape synthesis. *arXiv:1612.00101*, 2016. 1, 3, 6

[13] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc. NIPS*, 2016. 3

[14] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the rgb-d slam system. In *Proc. ICRA*, 2012. 6

[15] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. 2015. 3

[16] T. Gerig, A. Morel-Forster, C. Blumer, B. Egger, M. Lüthi, S. Schönborn, and T. Vetter. Morphable face models-an open framework. *arXiv preprint arXiv:1709.08398*, 2017. 2, 8

[17] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 5

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[19] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[20] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv:1506.05163*, 2015. 3

[21] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *Proc. ECCV*. Springer, 2016. 7

[22] M. Kazhdan and H. Hoppe. Screened Poisson surface reconstruction. *TOG*, 32(3):29, 2013. 2, 6

[23] V. G. Kim, Y. Lipman, and T. A. Funkhouser. Blended intrinsic maps. *TOG*, 30(4):79, 2011. 3

[24] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 5

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014. 2, 3

[26] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016. 3

[27] S. Korman, E. Ofek, and S. Avidan. Peeking template matching for depth extension. In *Proc. CVPR*, 2015. 2

[28] I. Kostrikov, J. Bruna, D. Panozzo, and D. Zorin. Surface networks. *arXiv:1705.10819*, 2017. 2

[29] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *Proc. ICML*, ICML'16, 2016. 2, 4

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. 3

[31] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *arXiv:1705.07664*, 2017. 3

[32] O. Litany, T. Remez, and A. Bronstein. Cloud dictionary: Sparse coding and modeling for point clouds. *arXiv:1612.04956*, 2016. 2

[33] O. Litany, T. Remez, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. *Proc. ICCV*, 2017. 3

[34] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2), 2017. 3

[35] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on Riemannian manifolds. In *Proc. 3dRR*, 2015. 3, 4

[36] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Proc. CVPR*, 2017. 3, 4, 6

[37] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. CVPR*, 2015. 7

[38] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proc. ISMAR*. IEEE, 2011. 6

[39] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *TOG*, 32(6):169, 2013. 6

[40] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *Proc. Workshop on Applications of Computer Vision*, 2013. 7

[41] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, 2016. 3

[42] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, 2017. 1, 3

[43] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv:1706.02413*, 2017. 1, 3

[44] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6620–6629, 2017. 3

[45] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3D object shape from one depth image. In *Proc. CVPR*, 2015. 2

[46] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017. 3

[47] E. Rodolà, S. Rota Bulò, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *Proc. CVPR*, 2014. 3

[48] K. Sarkar, K. Varanasi, and D. Stricker. Learning quadrangulated patches for 3D shape parameterization and completion. *arXiv:1709.06868*, 2017. 2

[49] A. Sharma, O. Grau, and M. Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *Geometry Meets Deep Learning Workshop at European Conference on Computer Vision (ECCV-W)*, 2016. 1, 2

[50] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proc. CVPR*, page 7, 2017. 7

[51] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *arXiv:1611.08974*, 2016. 1, 2

[52] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. CVPR*, 2015. 1, 3

[53] Q. Tan, L. Gao, Y.-K. Lai, and S. Xia. Variational autoencoders for deforming 3d mesh models. *arXiv preprint arXiv:1709.04307*, 2017. 2

[54] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. 3

[55] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen. Shape completion enabled robotic grasping. In *IROS*, 2017. 1, 2

[56] N. Verma, E. Boyer, and J. Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proc. CVPR*, 2018. 3, 4

[57] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4), 2017. 3

[58] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[59] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li. Dense human body correspondences using convolutional networks. In *Proc. CVPR*, 2016. 1, 3

[60] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Proc. NIPS*, 2016. 2, 4

[61] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proc. CVPR*, 2015. 1, 2, 3

[62] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proc. CVPR*, July 2017. 3