






Mesh Draping: Parametrization-Free Neural Mesh Transfer

A. Hertz,¹  O. Perel,¹  R. Giryes,¹ O. Sorkine-Hornung²  and D. Cohen-Or¹

¹Tel Aviv University, Tel Aviv, Israel
amirhertz@mail.tau.ac.il, raja@tauex.tau.ac.il, cohenor@gmail.com
²ETH Zurich, Zurich, Switzerland
sorkine@inf.ethz.ch

Abstract

Despite recent advances in geometric modelling, 3D mesh modelling still involves a considerable amount of manual labour by experts. In this paper, we introduce *Mesh Draping*: a neural method for transferring existing mesh structure from one shape to another. The method drapes the source mesh over the target geometry and at the same time seeks to preserve the carefully designed characteristics of the source mesh. At its core, our method deforms the source mesh using progressive positional encoding (PE). We show that by leveraging gradually increasing frequencies to guide the neural optimization, we are able to achieve stable and high-quality mesh transfer. Our approach is simple and requires little user guidance, compared to contemporary surface mapping techniques which rely on parametrization or careful manual tuning. Most importantly, *Mesh Draping* is a parameterization-free method, and thus applicable to a variety of target shape representations, including point clouds, polygon soups and non-manifold meshes. We demonstrate that the transferred meshing remains faithful to the source mesh design characteristics, and at the same time fits the target geometry well.

Keywords: shape synthesis, shape modelling, neural networks

CCS Concepts: • Computing methodologies → Shape analysis

1. Introduction

Polygonal meshes are *de facto* the most common discrete representation of surfaces in computer graphics. For geometric modelling purposes, their expressiveness allows to capture adequate approximations of many object surfaces. At the same time, their modular design has sparked a plethora of interactive editing tools. That, in turn, makes them appealing to 3D artists for various creative tasks, such as modelling, sculpting and rigging.

The flexible nature of meshes allows artists to conceive shapes of varying densities by diligently employing local operations. Detailed mesh areas, and in particular curved parts and features relevant for articulation, are usually carefully crafted with higher polygonal count and certain connectivity structures. For example, a mesh of a humanoid may contain both a detailed, dense polygon fan of an eye as well as a low-poly, flat torso. Much of this delicate design involves meticulous repetitive work, oftentimes when jointly drafting multiple mesh models. Experts may resort to the usage of templates, but this is limited to certain use cases. A scanned point cloud, for instance, is not generated from such a template and therefore cannot

directly inherit a desired mesh topology. To alleviate the hard labour of artists, a wealth of research efforts has been focusing on *meshing* and *remeshing* techniques, which, given a target polygonal shape, generate an alternative, coherent and enhanced mesh topology. This is usually achieved by optimization methods with various objectives that aim to optimize the polygon distribution while avoiding degenerate faces [Cam17].

While these methods are useful, they are inadequate at times, as they neither allow users full control over the resulting connectivity, nor directly support polygon soups and point cloud targets. As a consequence, they are missing the opportunity to re-use sophisticated meshes carefully crafted by experts.

In this paper, we introduce an alternative approach, namely, a method for transferring an existing, high-quality reference mesh structure to a comparable target shape. Unlike previous methods that optimize both the connectivity and the geometry of the target shape mesh, our method re-uses the source mesh characteristics while deforming it to best fit the geometry of the target. Our algorithm relies on a neural optimization process at its core: We learn the weights of

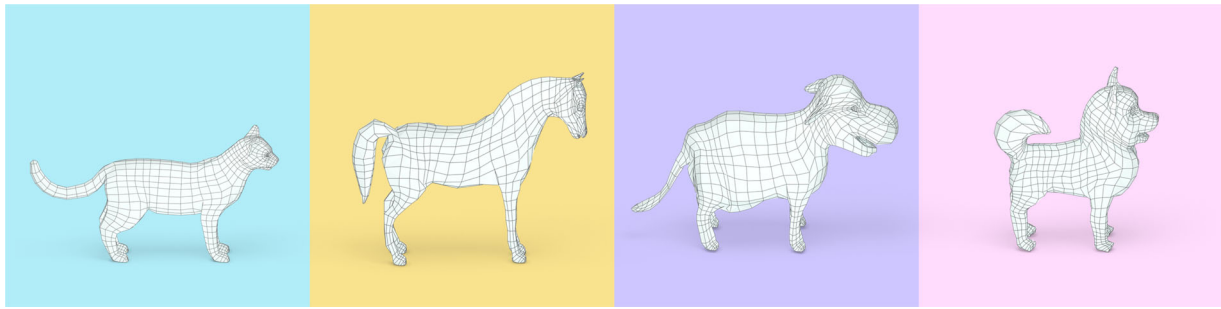


Figure 1: Mesh Draping in action: the source mesh of the cat model (left) is transferred with minimal distortion onto other target shapes.

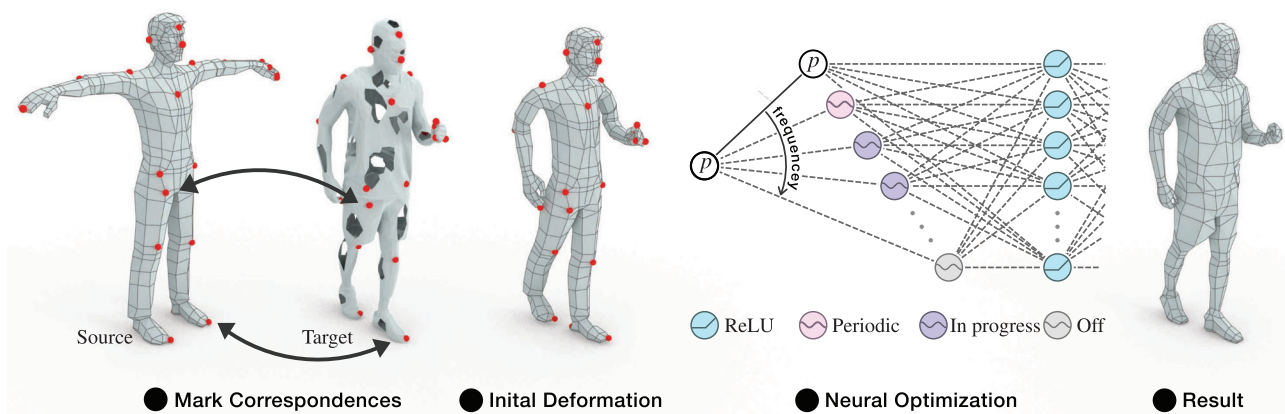


Figure 2: Method overview. Our pipeline begins with a manual step of marking sparse correspondence points between the source and the target. Then, source mesh deformation takes place, to obtain an initial solution for the mesh transfer neural optimization. After that, the optimization commences, until it finally converges to the resulting transferred mesh.

a lightweight network to deform a given source mesh to the target shape. In Figure 1, we show a mesh transfer from a source surface (on left) to three different target shapes. We follow contemporary literature, and overcome the shortcomings of simple neural nets by employing varying positional encodings (PEs). More specifically, we gradually map the source mesh vertex positions through encoding functionals of increasing frequencies [1] from low to high, before feeding them into the network. This progressive scheme introduces an *inductive bias* of slowly changing global deformations. That allows us to maintain a stable optimization through the earlier, critical steps of the optimization. Towards the end of the optimization, it also allows us to fit delicate details, where higher frequencies pertaining to fine local features are mapped (Figure 2).

We propose Mesh Draping, a parameterization-free, neural approach for mesh transfer between shapes. Mesh Draping is an interactive tool, where the user merely guides the displacement procedure. That is, the source mesh is ‘draped’ over the target geometry. Our optimizing technique yields high quality meshes, faithful to both the source design and the geometry of the target shape. Most importantly, it is not limited by the target shape representation, rendering our approach suitable for transferring mesh structures to an assortment of shape formats, such as point clouds, polygon soups and nonmanifold meshes. We demonstrate results

of transferring triangle and quad meshes on a variety of examples, ranging from 3D models generated by artists to synthetic objects comparable with other methods.

2. Related Work

2.1. Remeshing

The literature on *meshing* and *remeshing* is concerned with methods that directly generate new mesh structures of the desired shapes without assuming any prescribed connectivity. Oftentimes, a generated mesh of good quality ideally comprises of well-shaped triangles or quadrilaterals that are aligned to principal directions.

Different remeshing methods allow varying degrees of control by the user. Automated or semi-automated techniques may utilize high-level inputs, such as density control and crease selection [AMD02, LKH08, FBT*18, JTPSH15]. Other methods allow further interactions with the user. Bommers *et al.* [BZK09] let users specify orientation and alignment constraints. Campen and Kobbelt [CK14] introduce dual strip weaving, a framework where the user can specify the main layout of the quad grid. Marcias *et al.* [MTP*15] learn quadrangulation patterns, applying them by guided user sketches of

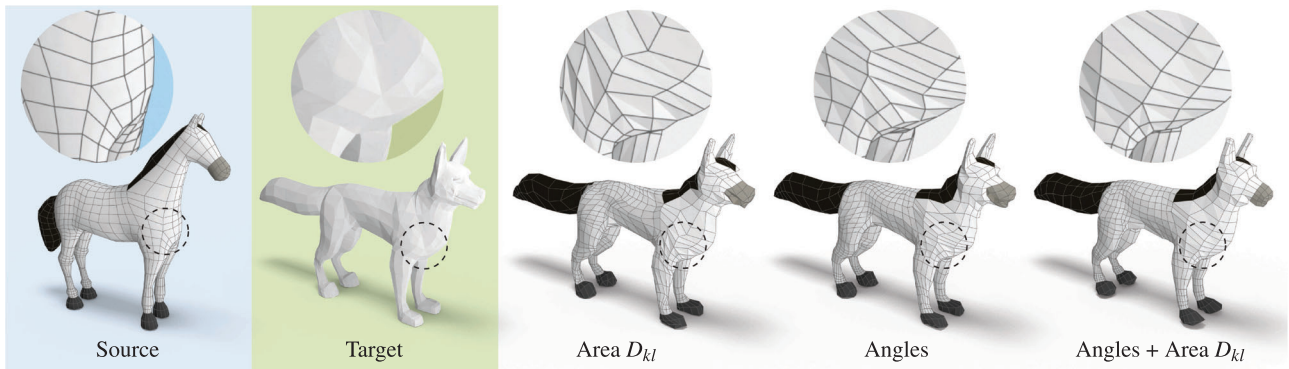


Figure 3: Ablation of the structural loss component: $\mathcal{L}_s(\widehat{\mathcal{M}}|\mathcal{M})$ (Equation 3). Middle: \mathcal{L}_s considers only the area term, D_{kl} , which leads to large distortions with respect to the source mesh (blue). Next to the right, optimizing while minimizing the angle error term achieves better results. Right: combining both losses better preserves the tessellation of the source mesh.

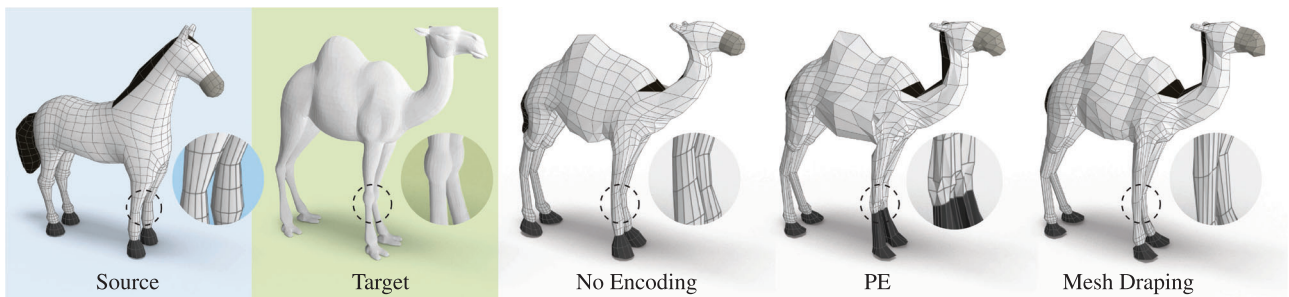


Figure 4: Mesh transfer from source to target, comparing various neural optimization methods. No Encoding, which uses common MLPs with ReLU activations, fails to fit delicate target geometry details. Fourier PE is able to fit fine details, but converges to a sub-optimal minima which introduces distortions. Progressive PE endows a coarse-to-fine inductive bias, which contributes to stable optimization and allows for high-quality mesh transfer.

the edge flows. Ebke *et al.* [ESCK16] suggest a rapid feedback interface that enables more precise control by specifying constraints, such as the placement of singularities and their connections, as well as smooth surface regions, among others. The above methods generate high-quality meshes, however, they do not offer users the option of employing existing, custom mesh structures.

Another class of remeshing methods focuses on transferring the connectivity of the mesh [ACBCO17, MMM*20]. Another line of works deals with compatible meshing. Such methods remesh multiple shapes to generate common connectivity among them [YFC*18, YZL*20]. These methods generally aim to produce meshes of regular connectivity and minimal polygon count that fit both shapes to facilitate applications like morphing or attribute mapping; however, they discard the original mesh connectivity, in contrast to the goal of our work.

For further reading about triangle and quadrilateral remeshing algorithms, we refer the curious reader to the comprehensive surveys by Alliez *et al.* [AUGA08] and Campen [Cam17].

From a practical point of view, Mesh Draping is not just a remeshing method *per se*, but rather, it focuses on transferring a complete

mesh-structure from one shape to another. It, therefore, puts an extra emphasis on minimizing the distortion of the original connectivity, which can benefit practical applications, e.g., to assist artists re-use useful mesh structures from past works. This stands in contrast to other automatic research directions. For example: remeshing often produces a regular mesh endowed with geometrically desired properties. However, its structure is beyond the control of the artist due to its automatic nature (we can keep carefully meshed areas such as the eyes in Figure 5, or varying mesh density in Figure 8).

2.2. Surface mapping

Our work shares similarities with methods that map attributes between surfaces. In the following, we provide a brief summary of relevant surface mapping techniques; a more extensive discussion is available in the survey by Tam *et al.* [TCL*13].

Broadly speaking, most works can be characterized into extrinsic approaches, which leverage surface deformation embedded within a 3D space, and intrinsic approaches, which achieve a complete bijective mapping between the two surfaces via parameterization to some intermediate domain.

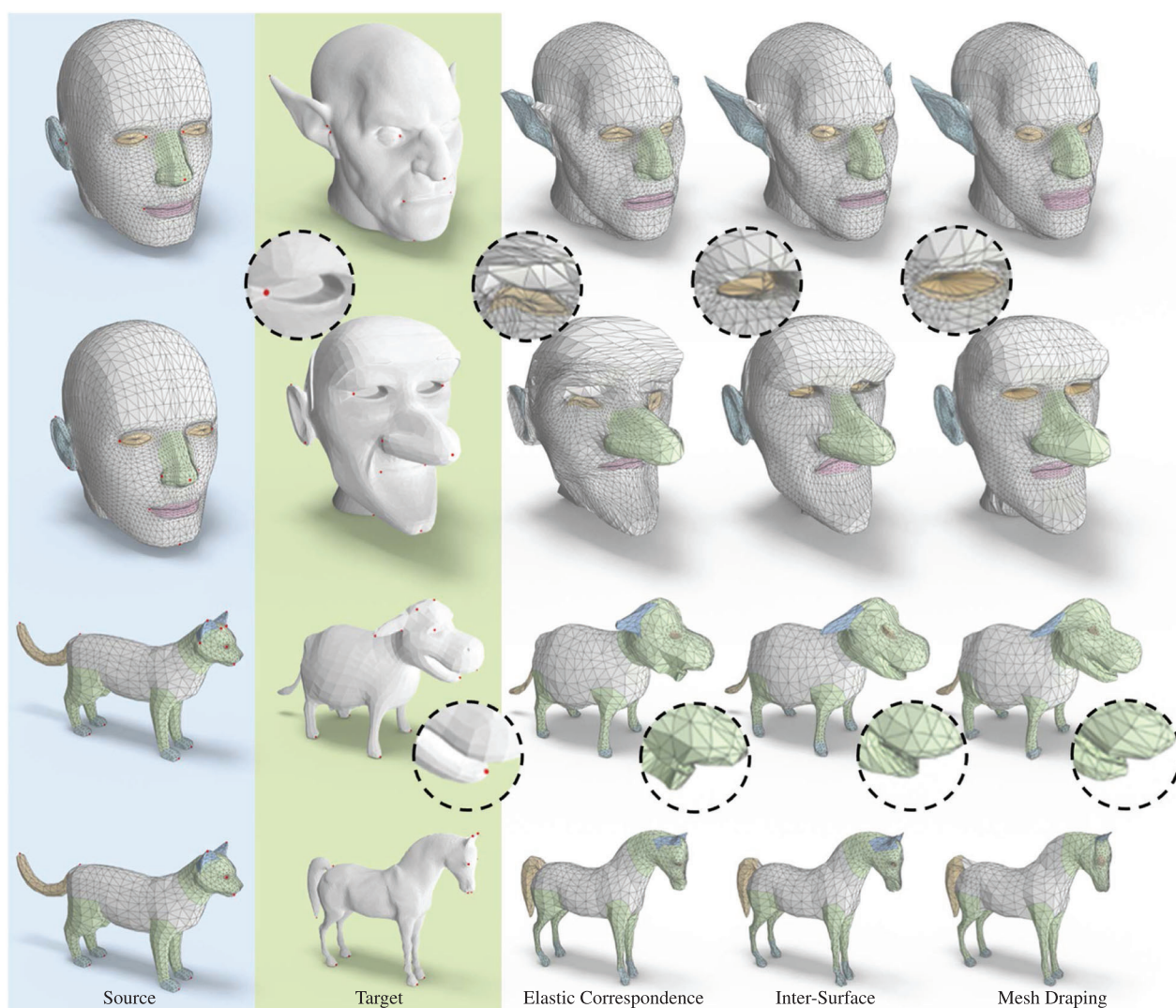


Figure 5: Comparison to direct surface mapping methods. Sparse correspondence points manually marked by users to guide algorithms are indicated by red dots. Face colours indicate semantic areas, showing how the mesh is stretched or contracted to fit the target shape. Mesh Draping is able to handle transfer of fine mesh structure in highly detailed areas.

In a purely extrinsic approach, the deformation process is preceded by global alignment defined by corresponding points on the source and target shapes [ZSCO*08, LSP08]. Local transformations calculated per element depend on the global orientation of both shapes. Early approaches, such as the renowned iterative closest point method (ICP) [CM92, BM92], assume the scenario of rigid registration, that is, the transformation between the source and target shapes maintains pairwise point distances. Follow up works remove this restriction and perform non-rigid registration [SBSCO06], possibly in a setting of partially overlapping surfaces with outliers [LSP08, HAWG08, BTP13].

By contrast, in the intrinsic line of works, mapping is achieved by means of parameterization to an intermediate domain. The actual mapping is achieved using a composition of mappings from source shape to the intermediate domain and an inverse

mapping from the intermediate domain to the target shape. Representative examples include Refs. [KS04, KLF11, APL15, AL15, AL16, BCK18, SBCK19]. Many of these works aim for either continuous or bijective mappings, or both. To handle non-isometric cases, Mandad *et al.* [MCSK*17] suggest using an optimal transport plan without requiring an intermediate domain, geared towards as-conformal-as-possible mapping. A recent work by Schmidt *et al.* [SCBK20] presents a novel method for continuous bijective mapping (homeomorphism) of attributes between surfaces of the same genus. Their method is able to obtain low intrinsic distortion, as well as generalize to arbitrary genus. Deng *et al.* [DBSF20] present a neural approach to reconstruct a target shape via an atlas of stitched patch mappings. Unlike our method, these methods rely on surface parameterization and cannot be easily extended to domains beyond manifold meshes, such as point clouds.



Figure 6: Comparison to deep deformation methods. The usage of progressive positional encoding allows Mesh Draping to deform the source mesh and remain faithful to delicate details of the target shape.

Ezuz *et al.* [EHA*19] propose a hybrid extrinsic approach that builds upon an intrinsic initialization. Their design combines an optimization scheme of elastic thin-shell deformation of the source mesh with projection over the target. They can handle non-isometric shapes, but struggle on highly complex meshes. References [ETLTC20, ELC20] produce a good shape correspondence between two shapes using an iterative alignment of smooth shells.

The problem of non-isometric mapping has also been studied from a spectral point of view. References [OBCS*12, OCB*17] propose using functional maps to transfer functions between two meshes using eigendecomposition. Newer neural variants of this

method also exist [LRR*17, GR20, DSO20]. These works create vertex-to-area mappings, rather than vertex-to-vertex mappings, which renders them less suitable for the purpose of mesh transfer.

The work of Tierny *et al.* [TDN*11] is closest to ours in spirit, as it also allows for direct quad mesh transfer between source and target meshes. In their work, they construct a corpus of source meshes, and use cross-parameterization with as-rigid-as-possible deformations to automatically generate a mapping between the shapes. Their method assumes that a parameterization of the target shape is available, as well as the existence of a homeomorphism. Their method requires a preprocessing phase for generating the aforementioned

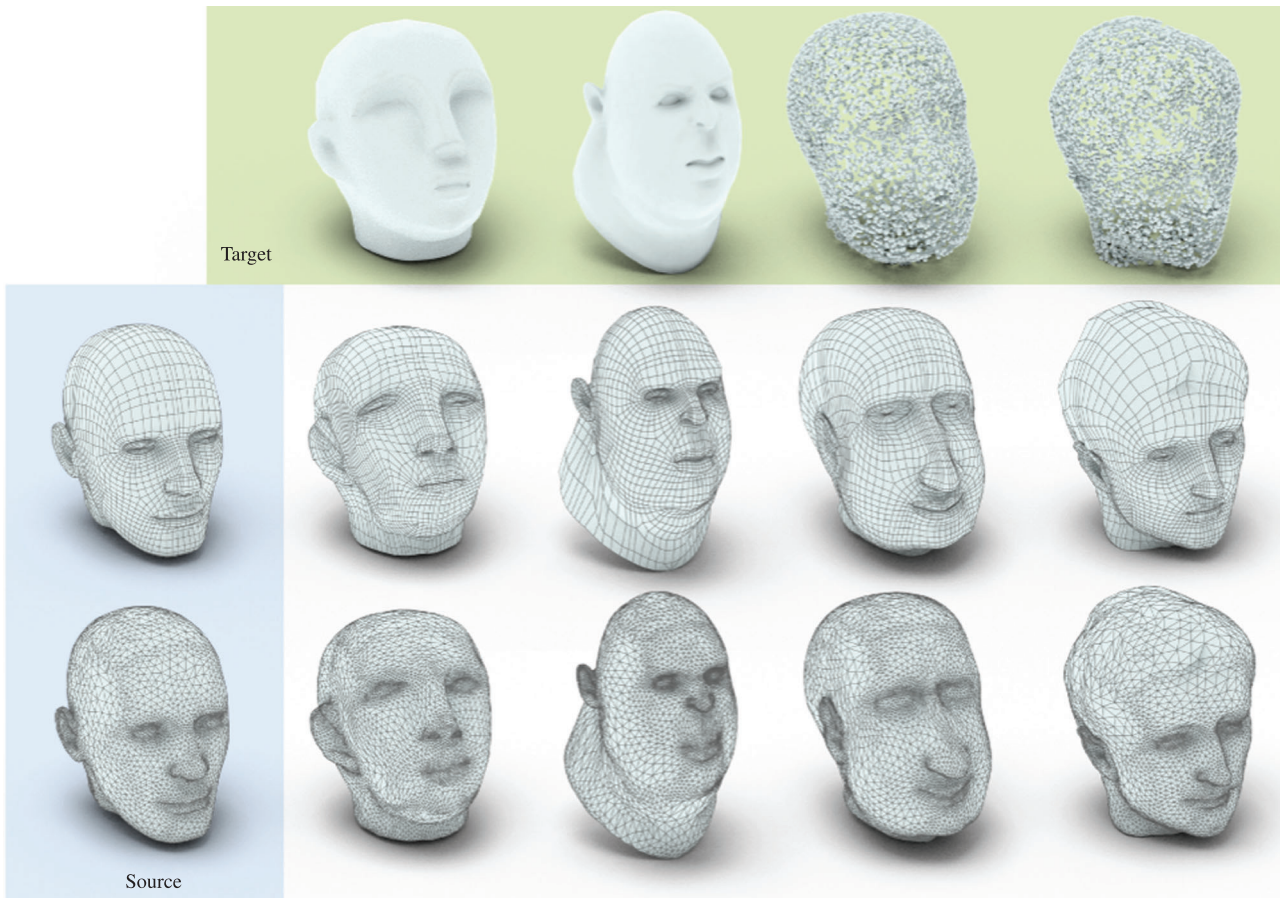


Figure 7: Applying Mesh Draping to transfer the structure of the mesh faces on left to the target meshes or point clouds on top. Our method supports direct mesh transfer to non-manifold shapes.

corpus, where boundary conditions are provided and segmentation masks are marked. Their method then requires to stitch the back of the segments in a postprocess.

A common application for surface mapping methods is texture mapping, where the emphasis is on minimizing global distortion of the mapped textures. In contrast, our work is parameterization-free, global and differs by focusing on correlative face transfer between similar areas of the two shapes, denoted by users, as well as maintaining the original mesh integrity. While mesh transfer can be implemented via a surface mapping method, we show in Section 4 that such methods produce inferior results in the extrinsic case, and cannot be used when a manifold mesh is not available and bijective parameterization cannot be directly achieved.

2.3. Shape deformation

Applying unconstrained deformations to mesh vertices introduces a risk of geometric artifacts, such as intersecting faces. Our work shares common ground with shape deformation methods, and similarly to them, we cope with the challenge of maintaining the mesh coherency under geometric warping. To that end, many of the shape deformation methods are concerned with the type of regularization

imposed on the mesh transformations, or proper design of interactive editing tools offered to the user. We highlight however, that the end goal of shape deformation works is somewhat different: they aim to preserve a single source shape under some set of user constraints. It can be used for performing shape registration and interpolation [GFK*18, ELC19, ENK*21].

To tie up the discussion, we mention works most relevant to our problem domain. Yeh *et al.* [YLSL11] suggest deformation by template-based fitting of 3D meshes, using a coarse to fine pipeline. Jacobson *et al.* [JBPS11] present a unified interactive approach combining control handles, skeletons and cages to deform a given shape. Their method uses weighted linear blending for practical purposes of mesh editing and rigging. For an elaborate review of non-neural methods, we refer readers to Sorkine and Botsch [SB09], Jacobson [Jac13] and the references therein.

More recently, Hanocka *et al.* [HFW*18] proposed using free-form-deformations for aligning shapes using a learned shape specific prior. Yifan *et al.* [YAK*20] demonstrated a learned variant of cage-based deformation. Given a source and a target shape, their algorithm warps the source mesh to match the target. However, this work focuses on preserving local details within the source shape, and does not guarantee that the result intimately

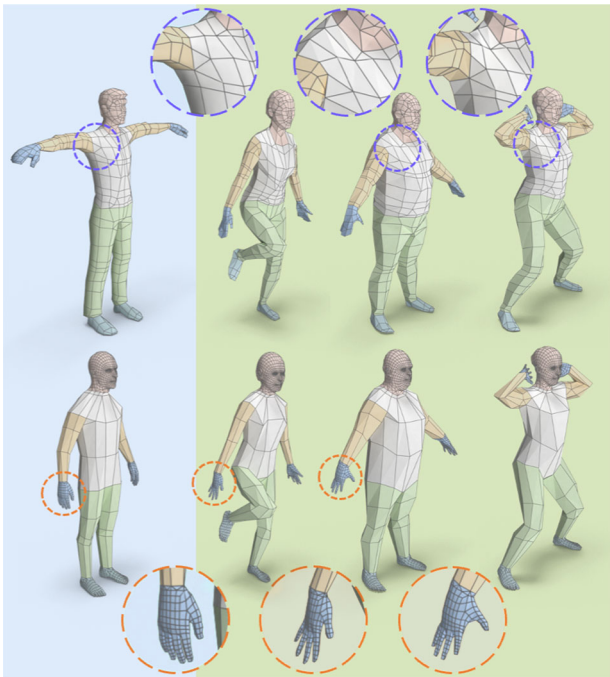


Figure 8: Mapping meshings of varying density, highlighted in different colours. Mesh Draping is able to fit the target shape while retaining the original mesh structure with minimal distortion (zoom in for details).

coincides with the target shape. Another contemporary work by Wang *et al.* [WCMN19] performs neural warping of source to target shapes by concatenating their global representations and predicting per-vertex offsets. Their method relies on a compound loss that assumes symmetry, and is limited to the domain of the training set. The ShapeFlow method [JHTG20] is a scheme for learning a deformation space. Given a target shape, the approach employs a nearest neighbour search for a pre-learned source candidate to guide the warp process. Our method, in contrast, does not assume any learned shape representation, and can be applied directly to novel shapes from unseen domains.

Deformation-based methods are generally susceptible to accurately fitting the target shape as we show hereafter in Figure 6. This is especially true for the small, finer details of the shape, which may have less impact on the overall cost function of the optimization. Mesh Draping relies on the progressive PE for neural optimizations from Hertz *et al.* [] and uses it for optimizing also the fine details in our Mesh Draping strategy.

3. Method Overview

Our framework begins with a preliminary step, where the user specifies a small number of correspondence points on the source and target shapes. After that, the principal part of the algorithm, the optimization, takes place automatically, during which we allow the source mesh vertices to shift their positions to fit the target shape. We optimize an objective function that expresses the Euclidean dis-

tance between the source mesh and the target geometry, regulated by the marked user correspondence points. The objective function encapsulates complementary terms that concurrently fit the target shape and respect the structure of the source mesh. In a nutshell, the process iteratively projects the source mesh onto the target surface, and shifts the projected vertices to preserve face angles and local area entropy.

During the optimization phase, our method learns the parameters of a deep neural network that performs the pairwise mapping of source mesh vertex positions to offsets that fit their target shape locations. To facilitate the learning process, we introduce a progressive PE layer [] at the head of the network. Simply put, a progressive encoding layer maps the input vertex positions to a higher dimensional space by feeding them through periodic functions with gradually increasing frequencies. During optimization, it progressively reveals higher frequencies of source vertices' positional embedding as an input to the mapping network. We assert the claims of Refs. [TSM*20,] and demonstrate how the optimization benefits both from the stability introduced by spectral bias of the network [RBA*19, Gir22, BDG22] and its non-biased solutions pertaining to high-frequency mappings [SMB*20, MST*20].

In Section 3.1, we describe the preliminary setup of our method. In Section 3.2, we describe in detail the optimization terms of our *mesh transfer* solution. Finally, in Section 3.3, we lay out the progressive configuration which allows Mesh Draping to achieve stable optimization and high-quality results.

3.1. Correspondence setup

A key aspect of our method is that it provides the user means to guide the mesh transfer. First, the user marks a small set of correspondence points $\{v_i, u_i\}_{i=1}^k$ between the source mesh \mathcal{M} and the target shape \mathcal{T} . The correspondence points enable the estimation of an initial global affine transformation from \mathcal{M} to \mathcal{T} , followed by a biharmonic deformation [JBPS11] that utilizes the corresponding points on \mathcal{T} as the boundary conditions. In addition, the user can specify the correspondences as *rigid*, for example, in Figure 8, the movement of the body parts is specified by rigid points. In those cases, we apply another *as-rigid-as-possible* deformation [SA07] using the rigid points as the boundary conditions. See Section 4.1 for elaborate implementation details.

The optimization phase starts when the user is satisfied with the initial deformation. Note though that our method is not sensitive to the exact initialization used. Specifically, the main requirement from the initialization is that the two shapes will be aligned and at the same scale. The use of biharmonic deformations together with ARAP for initialization is for the purpose of having a good starting point where the selected points are also aligned. Yet, notice that our method is robust to the initialization as it can provide good results also when no control points are being provided (see e.g., Figure 6).

Given the initialization, the objective of the optimization is to bring the surface of \mathcal{M} close to \mathcal{T} while maintaining the structural properties of the original mesh, as described in Section 3.2.

Interactive mode. Mesh Draping also includes an interactive mode. Between the optimization epochs, the user may pause, and

make additional local modifications by adding or adjusting the correspondence points.

3.2. Neural optimization

Instead of formulating an *explicit* optimization term between the source mesh and the target, we use a neural network parameterization. Specifically, we use fully connected neural networks (FC) with parameters θ . Doing so has been shown to improve the optimization in various works [WSS*19, HMGCO20] as the network serves as an ‘internal prior’ in the optimization.

The mesh deformation is obtained through a direct neural optimization of the parameters θ of a mapping function $f(\mathcal{M}|\theta) = \widehat{\mathcal{M}}$ that receives a source mesh, \mathcal{M} , and outputs an optimized mesh $\widehat{\mathcal{M}}$.

The loss term follows directly from the definition of our problem:

$$\mathcal{L}(\widehat{\mathcal{M}}|\mathcal{T}, \mathcal{M}) = \mathcal{L}_d(\widehat{\mathcal{M}}|\mathcal{T}) + \lambda \mathcal{L}_s(\widehat{\mathcal{M}}|\mathcal{M}). \quad (1)$$

On the one hand, we would like our output mesh $\widehat{\mathcal{M}}$ to fit a given target shape \mathcal{T} as closely as possible, i.e., minimize the distance $\mathcal{L}_d(\widehat{\mathcal{M}}|\mathcal{T})$ between $\widehat{\mathcal{M}}$ and \mathcal{T} . On the other hand, we wish to preserve the structural quality of the source mesh, which is measured by $\mathcal{L}_s(\widehat{\mathcal{M}}|\mathcal{M})$.

The distance loss is given by

$$\mathcal{L}_d(\widehat{\mathcal{M}}|\mathcal{T}) = Ch(\widehat{\mathcal{M}}, \mathcal{T}) + \sum_{i=1}^k \|\widehat{v}_i - u_i\|_2^2, \quad (2)$$

where $Ch(\widehat{\mathcal{M}}, \mathcal{T})$ is a symmetric Chamfer distance between uniformly sampled points on the optimized mesh and the target shape. In addition, we keep the k correspondence points specified by the user close by minimizing the squared distance between them, where \widehat{v}_i and u_i are pairs of corresponding points on $\widehat{\mathcal{M}}$ and \mathcal{T} , respectively.

The structural loss is given by

$$\mathcal{L}_s(\widehat{\mathcal{M}}|\mathcal{M}) = \frac{1}{N} \sum_{i=1}^N \left(\sum_{j \in R(i)} \|\widehat{\alpha}_j - \alpha_j\|_2^2 + D_{KL}(P_i||Q_i) \right), \quad (3)$$

where the summation is over the N vertices of \mathcal{M} . The first term represents the distortion of the angles $\widehat{\alpha}_j$ on $R(i)$: the 1-ring of \widehat{v}_i with respect to the original angles α_j on \mathcal{M} . The second term measures the local area Kullback–Leibler divergence where P_i are the fixed areas of the faces around vertex i in \mathcal{M} , normalized to have sum 1. Q_i is the equivalent (non-fixed) local area distribution in $\widehat{\mathcal{M}}$. Figure 3 shows the effect of each structural loss term on the final optimization result.

To prevent numerical issues caused by *skinny* faces, we utilize the quality measure for a triangular face from [LKC*20]

$$Q_f = \frac{4\sqrt{3}A_f}{\|e_1\|^2 + \|e_2\|^2 + \|e_3\|^2}, \quad (4)$$

where A_f is the area of the face and $\|e_i\|$ is the length of its i th edge. When $Q \rightarrow 0$, the face approaches to degenerate zero area. To prevent such cases we penalize by $1 - Q_f$ all the faces in $\widehat{\mathcal{M}}$ with quality $Q_f < 0.1$.

3.3. Progressive positional encoding

It has been shown that the learning bias of deep ReLU networks tends to be towards low frequency functions, e.g., they have a spectral bias [RBA*19]. The spectral bias of the network has a positive trait of preventing large deformations during the optimization process. These may be caused by an unstable Chamfer term, which leads to an erroneous local minimum. Unfortunately, the spectral bias also comes with a price: As the optimization proceeds, it prevents local *delicate* deformations that bring the surface of the source close to the target shape. That implies that common FC network with ReLU activations have a hard time mapping a continuous domain to a high-frequency image.

To mitigate the deficiencies of FC network with ReLU activations, previous works by Refs. [TSM*20, MST*20, SMB*20] suggested frequency-based encodings. Specifically in a mesh transfer scenario, source mesh vertex positions are first mapped via PE before feeding them as input to the FC network. However, in the case of mesh transfer, static frequency encoding schemes introduces a new problem to the architecture. Encoding functionals of high frequencies may overfit too quickly, causing the optimization to converge to suboptimal solutions. See for example, Figure 4, for the distortion caused by the PE neural optimization.

Instead, Mesh Draping leverages the progressive PE layer of Hertz et al. []. Progressive PE operates under the assumption that earlier iterations of the optimization benefit from the spectral bias, which is achieved by low-frequency encodings. As the optimization converges, higher frequencies encodings are used to fit the delicate features of the shape. In the context of mesh transfer, this formulation enforces an inductive bias which ensures that the optimization is both stable and accurate.

In our experiments, we adopt a lightweight version of Hertz et al. []: we leverage progressive PEs, and trace their progression in a global manner, e.g.: all vertices are exposed to higher frequencies at the same time-step. Particularly, we have opted to neglect the spatially adaptive part [] as we have found that for the task at hand it does not provide a remarkable benefit that justifies its extra computation and memory. Therefore, our progressive PE layer incurs no additional memory cost compared to the static encoding.

Figure 4 shows the advantage of progressive PE over the common static PE [TSM*20]. Progressive PE maintains the spectral bias property of neural networks, which is crucial to stabilize the optimization at early steps. When spectral bias is not present, the output mesh may present distorted or flipped faces as the network attempts to ‘jump’ right to the solution, thus irrecoverably destroying the mesh structure. Our solution imposes an inductive bias which does not allow the mesh to change too quickly at early iterations. That, in turn, keeps our solution stable, and allows some resilience to transferring challenging geometric features, i.e., cavities.

4. Evaluation

Unless specified otherwise, we use the same optimization configuration detailed in Section 4.1 for all described experiments.

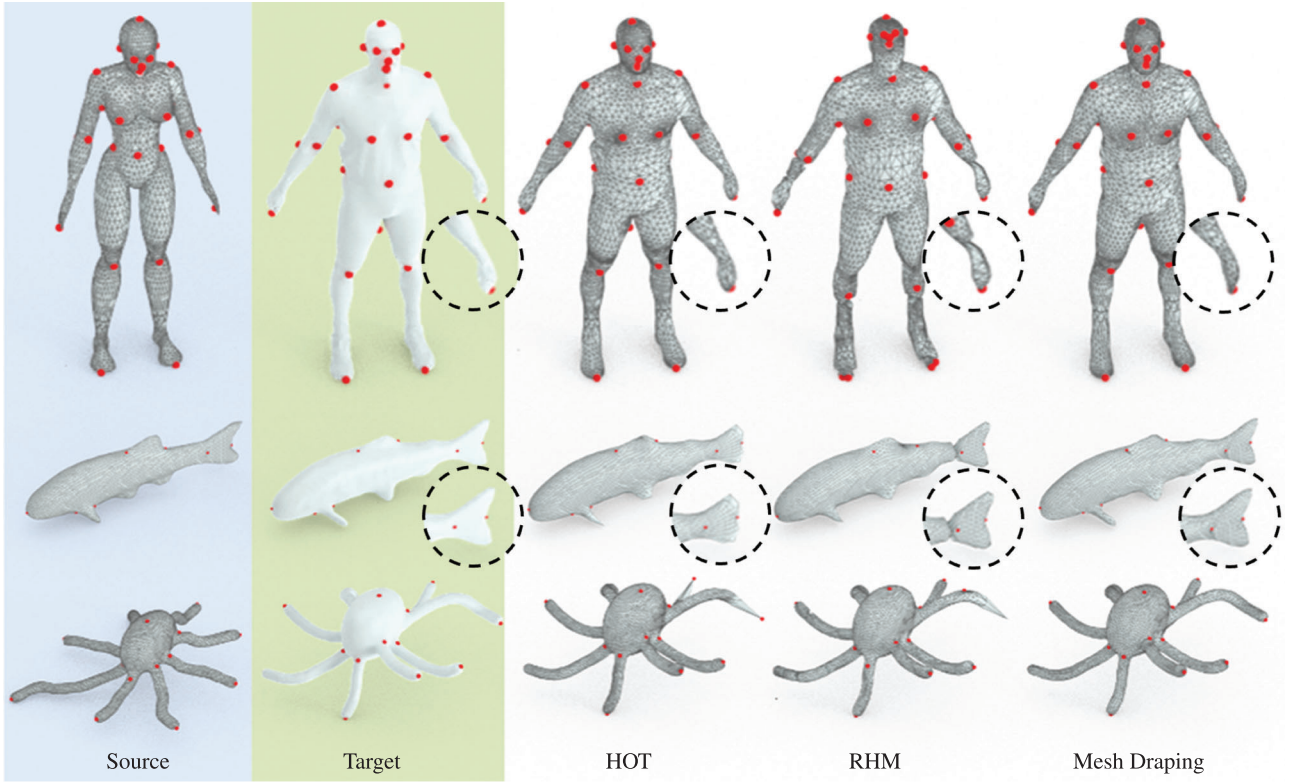


Figure 9: Comparison to additional surface mapping methods on the SHREC-BIM benchmark. In this comparison, correspondence points are provided with the data and not by manual user interactions. Red dots indicate correspondence points used by each method.

4.1. Implementation details

Correspondence setup. For most results shown in this paper, our algorithm requires users to mark 8–15 pairs of correspondence points on average. This number may vary with the complexity of the models. As a good measure, up to 10 correspondence points is the amount used to generate all face figures within this paper and up to 20 for the human bodies in Figure 8. For processing the influence area of correspondence points, we use the implementation of Jacobson *et al.* [JBPS11] and Sorkine and Alexa [SA07] from *libigl* [JP*18].

Network architecture and optimization. Our architecture consists of an FC network with four hidden layers, of size 256, where the first layer is a progressive PE [] layer divided into six blocks. Our optimization runs for 1500 iterations, alternating between backpropagations on $\mathcal{L}_d(\widehat{\mathcal{M}}|\mathcal{T})$ and $\lambda\mathcal{L}_s(\widehat{\mathcal{M}}|\mathcal{M})$, where the weight λ is set to 1 for the first 1000 iterations and later lowered to 0.2 for the rest of the optimization.

Latency. On a machine equipped with Nvidia GTX 1080, the optimization takes up to 45 s to converge for 1500 iterations.

4.2. Evaluation metrics

Our quantitative and qualitative evaluations highlight the importance of jointly optimizing a source mesh distortion metric, as well

as the alignment with the target shape. We measure the quality of distortion of $\widehat{\mathcal{M}}$ with respect to the source mesh \mathcal{M} by discrete Dirichlet energy [PP93]. The alignment integrity of the optimized mesh $\widehat{\mathcal{M}}$ over the target shape \mathcal{T} is evaluated using Chamfer and Hausdorff distances. It should be noted that the distortion and alignment metrics are complementary to one another, i.e., optimizing one but not the other provides inferior results as either the source structure is distorted, or the result does not perfectly align with the target shape. Readers may confirm the last statement by cross-comparing the visualizations of Figures 5, 6 and 9 with the quantitative evaluation in Table 1.

The aforementioned metrics are staple, and are commonly used on an individual basis. However, when measured concurrently for mesh transfer quality assessment, one has to account for their scale differences, their specific ranges, and how to effectively quantify them into one, comparable measurement. To that end, we propose a novel evaluation metric, $\mathbf{Q}_{\text{transfer}}$, which combines source distortion and target alignment measurements to a single score.

We define $\mathbf{Q}_{\text{transfer}}$ using the following template function:

$$\mathbf{Q}_{\tau}(\widehat{\mathcal{M}}|\mathcal{M}, \mathcal{T}) = 1 - \exp\left(\frac{-\tau}{\|\mathcal{F}_d(\mathcal{M}, \widehat{\mathcal{M}}) + \mathcal{F}_d(\mathcal{T}, \widehat{\mathcal{M}})\|_2}\right). \quad (5)$$

Here, the notations follow Section 3.2, where \mathcal{M} , \mathcal{T} and $\widehat{\mathcal{M}}$ represent the source mesh, the target shape and the transferred mesh, respectively.

Table 1: Quantitative comparison of our approach to other surface mapping and deformation methods, on the task of mesh transferring. We measure the mean Chamfer distance, Hausdorff distance, Dirichlet energy and our proposed $\mathbf{Q}_{\text{transfer}}$ metric over the shapes shown in Figures 5, 6 and 9.

Evaluation metric	Alignment		Distortion Dirichlet (\downarrow)	Joint $\mathbf{Q}_{\text{transfer}}$ (\uparrow)
	Chamf. ($1e^{-4}$) (\downarrow)	Hausd. ($1e^{-2}$) (\downarrow)		
ELC	1.97	4.23	1.52	0.690
ISM	1.39	3.14	2.29	0.770
Ours	1.18	1.91	1.84	0.900
HOT	13.78	4.57	12.76	0.327
RHM	8.78	3.49	6.54	0.534
Ours	8.34	2.22	3.29	0.791
Neural Cages	38.68	11.16	1.1	0.361
ShapeFlow	12.81	8.77	4.8	0.407
Ours	1.84	1.74	1.6	0.933

\mathcal{F}_d and \mathcal{F}_a are, respectively, the source distortion and target alignment measure functions, which are chosen in this paper as the Dirichlet energy and Hausdorff distance defined as

$$\mathcal{F}_d(\mathcal{M}, \widehat{\mathcal{M}}) = \frac{1}{2} \left(\sum_{f \in \mathcal{M}} |d\phi_{\mathcal{M}, \widehat{\mathcal{M}}}(f)|_2^2 A_f \right) - 1, \quad (6)$$

$$\mathcal{F}_a(\mathcal{T}, \widehat{\mathcal{M}}) = w_a \max \left\{ \sup_{x \in \mathcal{T}} \inf_{y \in \widehat{\mathcal{M}}} d(x, y), \sup_{x \in \widehat{\mathcal{M}}} \inf_{y \in \mathcal{T}} d(x, y) \right\}, \quad (7)$$

where for the discrete Dirichlet energy, we follow the definition of Ezuz *et al.* [ESBC19] and assume that $d\phi_{\mathcal{M}, \widehat{\mathcal{M}}}(f) \in \mathbb{R}^{2 \times 2}$ is the linear transformation that maps face $f \in \mathcal{M}$ to its image in $\widehat{\mathcal{M}}$, and a_f is the face area. To map the Dirichlet energy to the range of $[0, \infty]$, we also subtract a single unit. To align the magnitude of the Hausdorff distance, we scale it by a constant $w_a = 1e^2$, which we set empirically.

The calibration hyper-parameter τ is used to tone down the scale of the metric to better distinguish how different methods compare to each other. Throughout this paper, we use $\mathbf{Q}_{\text{transfer}} = \mathbf{Q}_\tau$ with $\tau = 5$. In practice, prior to metrics computation, all 3D models are normalized to the unit cube to ensure scale invariance.

To conclude, $\mathbf{Q}_{\text{transfer}}$ exhibits the following attributes:

- (i) $\mathbf{Q}_{\text{transfer}} \in [0, 1]$, where a higher score correlates with better quality.
- (ii) A perfect score of 1.0 is obtainable when $\mathcal{F}_d = 1$ (no distortion occurs) and $\mathcal{F}_a = 0$ (optimized shape perfectly aligns with target).
- (iii) When $\mathcal{F}_d \rightarrow \infty$ or $\mathcal{F}_a \rightarrow \infty$, then $\mathbf{Q}_{\text{transfer}} = 0$.
- (iv) A lower τ yields a ‘harder’ evaluation metric, which penalizes to a greater extent both misalignment and high distortion.

To tie up the discussion of metrics, we acknowledge that one of the terms in our evaluation metrics, the Chamfer distance, is directly used as one of the optimization terms. We emphasize this choice does not impact the integrity of the evaluation due to the optimization objective encompassing additional terms, and our evaluation consisting of other metrics as well.

4.3. Comparisons

We compare our method to other methods that we found as the most relevant to the task of mesh transfer. The quantitative results are summarized in Table 1.

SHREC-BIM benchmark. We evaluate our method on the BIM benchmark [KLF11] which includes 200 pairs of meshes from SHREC dataset [GBP07]. Each pair is supplied with corresponding points (between 2 and 36 points). We compare our method with two parameterization-based methods: Hyperbolic Orbifold Tutte Embeddings [AL16] (HOT) and Reversible Harmonic Maps between Discrete Surfaces [ESBC19] (RHM). Both methods receive as input the corresponding points and output a parameterization between the meshes that maps the vertices of one mesh to barycentric coordinates on the another mesh. As can be seen in Figure 9, the parameterization methods may struggle to keep on the triangulation of the source mesh. For example, see artifacts RHM outputs such as flipping faces on the fish tail and humanoid hand.

Additional mapping methods. We compare our method to additional two surface mapping methods on our custom test set. The first is Elastic Correspondence (ELC) [EHA*19] using the initialization scheme of Aigerman *et al.* [AL16], and the second is Inter-Surface Maps due to (ISM) [SCBK20]. Both methods are based on parameterization. Consequentially, they are only applicable for pairs of meshes, and also assume the existence of some bijectivity between the surfaces.

The comparisons are shown in Figure 5 where the input to the three methods are pairs of source and target meshes with their marked corresponding points. As evident by careful observation of the corresponding segmented parts marked on the source and the optimized mesh, ELC causes semantic distortion, for example, on the top face ears or the cow mouth. Both ISM and our method preserve the semantic correspondence between the meshes. These observations are also reflected in the quantitative results. Due to some distortions in the mapping of ISM, our method achieves *slightly better* results.

It is important to mention, that both methods, ISM and ELC, satisfy a complete bijective map between the input pair of meshes. However, an error evolves in places where the new discrete tessellation of the source does not properly cover the target when projecting

the map onto the target mesh. We suspect that these kind of errors appear mostly at delicate places, i.e., at the eyes of the topmost face and the cow mouth (Figure 5). In comparison, our direct optimization avoids such types of distortion.

Comparison to deformation methods. We also compare our method to two deformation methods. Both methods are based on a pre-training step over some specific dataset. Those methods can be categorized by the priors they inject to the deformation allowed between the source to the target shapes. The first method, Neural Cages (NC) [YAK*20], allows only coarse deformations by displacing the control points of a *cage* that wraps the source mesh. The second method, ShapeFlow (SF) [JHTG20], enables a more flexible deformation by allowing displacement for all vertices of the source mesh. Their deformation is regularized through the loss function that includes a volume conservation term and a penalization term for a non-isometric deformation.

The results are shown in Figure 6, where we trained and tested each method on selected classes from the ShapeNet dataset [CFG*15]. We apply our method without providing correspondence points, e.g.: we set $k = 0$ in Equation (2). In other words, we assume only a global orientation alignment between the source and the target meshes, as both other methods assume. Both qualitative and quantitative comparisons clearly highlight the difference between the methods. Our proposed progressive optimization better aligns the target mesh with the target shape, doing so with few distortions. By utilizing only global deformations, NC minimizes the distortion term but the resulted mesh is unable to satisfactorily obtain the form of the target mesh. Finally, SF brings the source mesh closer to the form of the target shape, but the resulting meshes usually suffer from undesired distortions.

Polygon soups and point clouds targets. Finally, we highlight the strengths of our progressive optimization by presenting visual examples of mesh-transfer in non-trivial scenarios. In Figure 7, we demonstrate a transfer of face meshes to point clouds, a task which surface mapping methods relying on bijective mappings cannot directly perform. The quality of our results are on par with the mesh-to-mesh case. This setting pertains to a real-life application, where experts may re-use pre-defined meshes on recently scanned objects.

Varying density meshes. In Figure 8, we illustrate the example of transferring meshes of varying density. For clarity, different areas of interest are segmented by colour. Though we suggest a global approach, it is able to transfer both sparsely and densely detailed polygonal areas with minimal distortion, maintaining the original intention of the expert artist. This example alludes to the case where specific parts of the mesh are highly detailed for a particular purpose, such as animation.

Limitations. While Mesh Draping exhibits very good results in many cases, there are still some things to improve in it. First and foremost, Mesh Draping is intended for pairs of shapes of similar parts. For example – mesh transfer will succeed between different four legged animals (Figure 1), but will under-perform for significant non-isometric deformations [DLR*20], e.g., when transferring meshing from humans to animals, or from a dinosaur to a giraffe as shown in Figure 10. Here, the horns shape of the giraffe requires special meshing, which does not exist on the dino head model. This

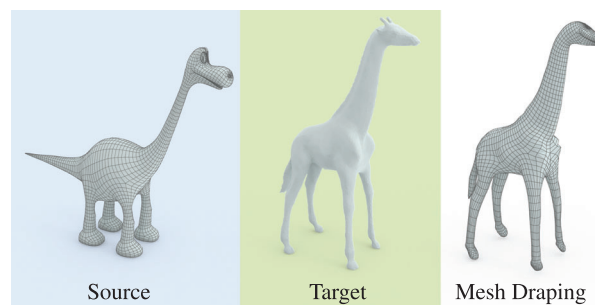


Figure 10: Limitation example: mesh transfer from the dinosaur mesh to the giraffe shape yields sub-optimal results unfaithful to the original target shape. Due to large shape differences, the tessellation of the dinosaur inherently struggles to accurately fit the giraffe ears or the delicate structure of its feet.

phenomenon is also demonstrated in Figure 6, row 4: the target truck shape contains a deep depression right above the right-front wheel, whereas the source shape is smooth. As a result, part of the resulting wheel mesh appears to be deformed. Note also that Figure 10 shows also that our method is not bijective as some of the points in the source shape are not mapped to any point in the target.

In addition, on extreme cases of sharp edges (so called ‘spikes’), the method may struggle to achieve perfect alignment between the optimized mesh and the target shape. To maintain high-quality results on very detailed meshes, users may be required to define additional correspondence points. Since we do not use a pre-defined train set, in such cases, user intervention is necessary.

By design, our method does not alter local connectivity at all. However, where needed, one may perform local subdivision or remeshing as a postprocess. Finally, the latency measures in this paper, are presented for an unoptimized implementation. Subsequent efforts may reduce the runtime cost (detailed in Section 4.1) by further optimizing the Mesh Draping logic.

5. Conclusion and Future Work

We presented a parameterization-free approach for transferring mesh structure to comparable shapes. The proposed method uses neural optimization with progressive frequency PEs, which contribute to both stable optimization and high-quality fitting to fine details. We demonstrated the applicability of our method on a range of examples, including the re-use of triangular and quad meshes to target meshes and point clouds.

Mesh Draping relies on the progressive PE for neural optimizations from Hertz et al. [], which is in particular geared for optimizing fine details. Specifically, in this work, we advocate that the progressive PE presented there makes simple fully connected neural networks an attractive optimization tool for optimizing subtle geometries, and show its advantage in neural mesh transfer. Figure 6 zooms in on some examples which highlight the ability of our approach for keeping the fine details in this task. We believe the results presented in this work would benefit future works that perform geometry optimization.

Future work may leverage an unpaired training set of shapes to obtain priors on common shapes for faster and more robust optimization. Another direction for future work may improve the computation time of our optimization, or its flexibility by allowing users to guide mappings between different topologies, or jointly deform a shape from several sources. Additional interesting direction is combining Mesh Draping with registration methods, e.g., Refs. [ETLTC20, ELC20] that can provide a good initialization, to remove the need for marking the corresponding points for non-rigid objects [BRLB14, MMRC20], add robustness to missing parts and also for extending our approach to dynamic shapes [BRPMB17].

Acknowledgement

This work is supported in part by the European Research Council (ERC) Consolidator Grant No. 101003104 and Starting Grant No. 757497, by the NSF-BSF Grant No. 2017729, by the Israel Science Foundation Grant No. 2492/20 and by the Innovation Programme Grant No. 101003104.

References

- [ACBCO17] AZENCOT O., CORMAN E., BEN-CHEN M., OVSJANIKOV M.: Consistent functional cross field design for mesh quadrangulation. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.
- [AL15] AIGERMAN N., LIPMAN Y.: Orbifold tutte embeddings. *ACM Transactions on Graphics* 34, 6 (2015), 1–12.
- [AL16] AIGERMAN N., LIPMAN Y.: Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics* 35, 6 (2016), 1–14.
- [AMD02] ALLIEZ P., MEYER M., DESBRUN M.: Interactive geometry remeshing. *ACM Transactions on Graphics* 21, 3 (2002), 347–354.
- [APL15] AIGERMAN N., PORANNE R., LIPMAN Y.: Seamless surface mappings. *ACM Transactions on Graphics* 34, 4 (2015), 1–13.
- [AUGA08] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: Recent advances in remeshing of surfaces. In *Shape analysis and structuring*. Springer (2008), pp. 53–82.
- [BCK18] BADEN A., CRANE K., KAZHDAN M.: Möbius registration. *Computer Graphics Forum* 37 (2018), 211–220.
- [BDG22] BAR O., DRORY A., GIRYES R.: A spectral perspective of dnn robustness to label noise. In *Proceedings of the International Conference on Artificial Intelligence and Statistics* (March 2022), G. Camps-Valls, F. J. R. Ruiz and I. Valera (Eds.), vol. 151 of *Proceedings of Machine Learning Research*, PMLR, pp. 3732–3752.
- [BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.
- [BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ, USA, June 2014), IEEE.
- [BRPMB17] BOGO F., ROMERO J., PONS-MOLL G., BLACK M. J.: Dynamic FAUST: Registering human bodies in motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017).
- [BTP13] BOUAZIZ S., TAGLIASACCHI A., PAULY M.: Sparse iterative closest point. *Computer Graphics Forum* 32, 5 (2013), 1–11.
- [BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. *ACM Transactions on Graphics* 28, 3 (July 2009), 1–10.
- [Cam17] CAMPEN M.: Partitioning surfaces into quadrilateral patches: A survey. *Computer Graphics Forum* 36 (2017), 567–588.
- [CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: ShapeNet: An information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012* (2015).
- [CK14] CAMPEN M., KOBBELT L.: Dual strip weaving: Interactive design of quad layouts using elastica strips. *ACM Transactions on Graphics* 33, 6 (2014), 1–10.
- [CM92] CHEN Y., MEDIONI G.: Object modeling by registration of multiple range images. *Image and Vision Computing* 10 (Jan. 1992), 145–155.
- [DBSF20] DENG Z., BEDNAŘÍK J., SALZMANN M., FUA P.: Better patch stitching for parametric surface reconstruction. *arXiv preprint arXiv:2010.07021* (2020).
- [DLR*20] DYKE R. M., LAI Y.-K., ROSIN P. L., ZAPPALÀ S., DYKES S., GUO D., LI K., MARIN R., MELZI S., YANG J.: SHREC’20: Shape correspondence with non-isometric deformations. *Computers & Graphics* 92 (2020), 28–43.
- [DSO20] DONATI N., SHARMA A., OVSJANIKOV M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the CVPR* (2020), pp. 8589–8598.
- [EHA*19] EZUZ D., HEEREN B., AZENCOT O., RUMPF M., BEN-CHEN M.: Elastic correspondence between triangle meshes. *Computer Graphics Forum* 38 (2019), 121–134.
- [ELC19] EISENBERGER M., LÄHNER Z., CREMERS D.: Divergence-free shape correspondence by deformation. *Computer Graphics Forum* 38 (2019), 1–12.
- [ELC20] EISENBERGER M., LAHNER Z., CREMERS D.: Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 12265–12274.
- [ENK*21] EISENBERGER M., NOVOTNY D., KERCHENBAUM G., LABATUT P., NEVEROVA N., CREMERS D., VEDALDI A.: Neuromorph: Unsupervised shape interpolation and correspondence in one go.

- In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7473–7483.
- [ESBC19] EZUZ D., SOLOMON J., BEN-CHEN M.: Reversible harmonic maps between discrete surfaces. *ACM Transactions on Graphics* 38, 2 (Mar. 2019). <https://doi.org/10.1145/3202660>
- [ESCK16] EBKE H.-C., SCHMIDT P., CAMPEN M., KOBELT L.: Interactively controlled quad remeshing of high resolution 3D models. *ACM Transactions on Graphics* 35, 6 (2016), 1–13.
- [ETLTC20] EISENBERGER M., TOKER A., LEAL-TAIXÉ L., CREMERS D.: Deep shells: Unsupervised shape correspondence with optimal transport. In *Proceedings of the Advances in Neural Information Processing Systems* 34 (2020).
- [FBT*18] FANG X., BAO H., TONG Y., DESBRUN M., HUANG J.: Quadrangulation through morse-parameterization hybridization. *ACM Transactions on Graphics* 37, 4 (2018), 1–15.
- [GBP07] GIORGI D., BIASOTTI S., PARABOSCHI L.: Shape retrieval contest 2007: Watertight models track. *SHREC Competition* 8, 7 (2007).
- [GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B. C., AUBRY M.: 3D-coded: 3D correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 230–246.
- [Gir22] GIRYES R.: A function space analysis of finite neural networks with insights from sampling theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022). <https://doi.org/10.1109/TPAMI.2022.3155238>.
- [GR20] GINZBURG D., RAVIV D.: Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *Proceedings of the ECCV* (2020).
- [HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. In *Proceedings of the Symposium on Geometry Processing* (2008), pp. 1449–1457.
- [HFW*18] HANOCKA R., FISH N., WANG Z., GIRYES R., FLEISHMAN S., COHEN-OR D.: ALIGNNet: partial-shape agnostic alignment via unsupervised learning. *ACM Transactions on Graphics (TOG)* 38, 1 (2018), 1–14.
- [HMGCO20] HANOCKA R., METZER G., GIRYES R., COHEN-OR D.: Point2mesh: A self-prior for deformable meshes. *ACM Transactions on Graphics* 39, 4 (2020), 126:1–126:12.
- [Jac13] JACOBSON A.: Algorithms and Interfaces for Real-time Deformation of 2D and 3D Shapes. PhD thesis, ETH Zurich, 2013.
- [JBPS11] JACOBSON A., BARAN I., POPOVIC J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics* 30, 4 (2011), 1–8.
- [JHTG20] JIANG C., HUANG J., TAGLIASACCHI A., GUIBAS L.: Shapeflow: Learnable deformations among 3D shapes. In *Proceedings of the Advances in Neural Information Processing Systems* (2020).
- [JP*18] JACOBSON A., PANOZZO D., et al.: libigl: A simple C++ geometry processing library. <https://libigl.github.io/> (2018).
- [JTPSH15] JAKOB W., TARINI M., PANOZZO D., SORKINE-HORNUNG O.: Instant field-aligned meshes. *ACM Transactions on Graphics* 34, 6 (2015), 1–15.
- [KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *ACM Transactions on Graphics* 30, 4 (2011), 1–12.
- [KS04] KRAEVOY V., SHEFFER A.: Cross-parameterization and compatible remeshing of 3D models. *ACM Transactions on Graphics* 23, 3 (Aug. 2004), 861–869.
- [LKC*20] LIU H.-T. D., KIM V. G., CHAUDHURI S., AIGERMAN N., JACOBSON A.: Neural subdivision. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 124:1–124:16.
- [LKH08] LAI Y., KOBELT L., HU S.: An incremental approach to feature aligned quad dominant remeshing. In *Proceedings of the SPM'08* (2008).
- [LRR*17] LITANY O., REMEZ T., RODOLÁ E., BRONSTEIN A., BRONSTEIN M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE ICCV* (2017), pp. 5660–5668.
- [LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. In *Proceedings of the Symposium on Geometry Processing* (2008), pp. 1421–1430.
- [MCSK*17] MANDAD M., COHEN-STEINER D., KOBELT L., ALLIEZ P., DESBRUN M.: Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics* 36, 4 (2017), 1–14.
- [MMM*20] MELZI S., MARIN R., MUSONI P., BARDON F., TARINI M., CASTELLANI U.: Intrinsic/extrinsic embedding for functional remeshing of 3D shapes. *Computers & Graphics* 88 (2020), 1–12.
- [MMRC20] MARIN R., MELZI S., RODOLÁ E., CASTELLANI U.: Farm: Functional automatic registration method for 3D human bodies. *Computer Graphics Forum* 39, 1 (2020), 160–173.
- [MST*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHY R., NG R.: NeRF: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934* (2020).
- [MTP*15] MARCIAS G., TAKAYAMA K., PIETRONI N., PANOZZO D., SORKINE-HORNUNG O., PUPPO E., CIGNONI P.: Data-driven interactive quadrangulation. *ACM Transactions on Graphics* 34, 4 (2015), 1–10.
- [OBSC*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: A flexible represen-

- tation of maps between shapes. *ACM Transactions on Graphics* 31, 4 (2012), 1–11.
- [OCB*17] OVSJANIKOV M., CORMAN E., BRONSTEIN M., RODOLÀ E., BEN-CHEN M., GUIBAS L., CHAZAL F., BRONSTEIN A.: Computing and processing correspondences with functional maps. In *Proceedings of the ACM SIGGRAPH 2017 Courses* (2017).
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 2, 1 (1993), 15–36.
- [RBA*19] RAHAMAN N., BARATIN A., ARPIT D., DRAXLER F., LIN M., HAMPRECHT F., BENGIO Y., COURVILLE A.: On the spectral bias of neural networks. In *Proceedings of the International Conference on Machine Learning, PMLR* (2019), pp. 5301–5310.
- [SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *SGP'07: Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2007), Eurographics Association, pp. 109–116.
- [SB09] SORKINE O., BOTSCH M.: Tutorial: Interactive shape modeling and deformation. In *Proceedings of the EUROGRAPHICS* (2009).
- [SBCK19] SCHMIDT P., BORN J., CAMPEN M., KOBBELT L.: Distortion-minimizing injective maps between surfaces. *ACM Transactions on Graphics* 38, 6 (2019), 1–15.
- [SBSCO06] SHARF A., BLUMENKRANTS M., SHAMIR A., COHEN-OR D.: Snappaste: An interactive technique for easy mesh composition. *The Visual Computer* 22 (2006), 835–844.
- [SCBK20] SCHMIDT P., CAMPEN M., BORN J., KOBBELT L.: Inter-surface maps via constant-curvature metrics. *ACM Transactions on Graphics* 39, 4 (2020), 119:1–119:15.
- [SMB*20] SITZMANN V., MARTEL J., BERGMAN A., LINDELL D., WETZSTEIN G.: Implicit neural representations with periodic activation functions. In *Proceedings of the Advances in Neural Information Processing Systems 33* (2020).
- [TCL*13] TAM G. K. L., CHENG Z., LAI Y., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X., ROSIN P. L.: Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE TVCG* 19, 7 (2013), 1199–1217.
- [TDN*11] TIERNY J., DANIELS J., NONATO L. G., PASCUCCI V., SILVA C. T.: Inspired quadrangulation. *Computer Aided Design* 43, 11 (Nov. 2011), 1516–1526.
- [TSM*20] TANCIK M., SRINIVASAN P. P., MILDENHALL B., FRIDOVICH-KEIL S., RAGHAVAN N., SINGHAL U., RAMAMOORTHI R., BARRON J. T., NG R.: Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the NeurIPS* (2020).
- [WCMN19] WANG W., CEYLAN D., MECH R., NEUMANN U.: 3DN: 3D deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 1038–1046.
- [WSS*19] WILLIAMS F., SCHNEIDER T., SILVA C., ZORIN D., BRUNA J., PANOZZO D.: Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 10130–10139.
- [YAK*20] YIFAN W., AIGERMAN N., KIM V. G., CHAUDHURI S., SORKINE-HORNUNG O.: Neural cages for detail-preserving 3D deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 75–83.
- [YFC*18] YANG Y., FU X.-M., CHAI S., XIAO S.-W., LIU L.: Volume-enhanced compatible remeshing of 3D models. *IEEE Transactions on Visualization and Computer Graphics* 25, 10 (2018), 2999–3010.
- [YLSL11] YEH I.-C., LIN C.-H., SORKINE O., LEE T.-Y.: Template-based 3D model fitting using dual-domain relaxation. *IEEE TVCG* 17, 8 (2011), 1178–1190.
- [YZL*20] YANG Y., ZHANG W.-X., LIU Y., LIU L., FU X.-M.: Error-bounded compatible remeshing. *ACM Transactions on Graphics* 39, 4 (2020), 113:1–113:15.
- [ZSCO*08] ZHANG H., SHEFFER A., COHEN-OR D., ZHOU Q., VAN KAICK O., TAGLIASACCHI A.: Deformation-driven shape correspondence. In *Proceedings of the SGP* (2008), pp. 1431–1439.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supplemental Video S1