EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner
(Guest Editors)

Volume 42 (2023), Number 2

Appendix for Unsupervised Template Warp Consistency for Implicit Surface Correspondences

Mengya Liu[1] Ajad Chhatkuli[1] Janis Postels[1] Luc Van Gool[1] Federico Tombari[2]

[1]ETH Zurich
[2]Google, TU Munich

I. Overview

We present further experimental results and details regarding unsupervised template warp consistency for implicit surface correspondences. In Section II, we introduce the architecture of the network and its hyperparameters, followed by the evaluation details and dataset pre-processing. In Section III, more experimental results on ShapeNet [CFG*15] are provided.

II. Experimental Details

II.1. Network Architecture

Deform-Net $\mathcal{D}$. The Deform-Net $\mathcal{D}$ is composed of 6 coupling layers and an LSTM cell [HS97]. In each layer, the split dimension first goes through a projection layer to 128 dimensions and then concatenates with the latent code. The result is then mapped to 256 dimensions by an LSTM cell and fed into the scaling and translation layers. The scaling λ and translation δ layers consist of 2 MLPs with 256, 3 neurons respectively. We use ReLU as the activation function. An additional Hardtanh activation is applied to the scaling function to limit the scaling be in [-10, 10]. The latent code is initialized as a 256 dimensions random vector and optimized along with the network.

Implicit Template $\mathcal{T}$. For comparison, we use the same implicit template $\mathcal{T}$ as DIT [ZYDL21]. The module is inherited from DeepSDF [PFS*19], consisting of 6 MLPs with 3, 256, 256, 256, 256, 256 and 1 neurons. We use weight normalization, 0.05 dropout probability and also ReLU activation except for the last layer. In the last layer, a hyperbolic tanh function is used as an activation function.

Explicit Constraints. We use a small SP-Net [CLC*20] to extract structural points on the deformed shapes. The network consists of a PointNet++ [QYSG17] encoder with two set abstraction levels of 128 and 64 grouping centers. Multi-scale grouping (MSG) layers contain (0.1, 0.2, 0.4) and (0.2, 0.4, 0.8) scales in two layers respectively. The output of the encoder is 128 sampled points with the local features of 224 dimensions. Later the sampled points together with the local features are trained through a point integration module combined of 3 MLP layers with the neuron numbers (128, 64, 16) and output 16 structural points. The dropout ratio is 0.2.

Parameter Name	Value
Nr. of Coupling Layers	6
w_0	1
w_1	0.0001
w_2	0.0001
w_3	0.0001
w_4	0.001
w_5 (Epoch $>$ 500)	0.001
Batch Size	40
Number of Points per Shape	4000
Number of Points on the Surface per Shape	2000
Number of Structural Points picked	16
Epochs	2000
Marching Cube [LC87] Resolution	256

Table A: *Hyperparameters. w_4 is 0 when epoch $<$ 500, and 0.01 when epoch $>$ 500.*

Training details. We provide further hyperparameters in Table A. For ShapeNet [CFG*15], we train a separate model for each category, and for DFaust [BRPB17], we train one model for the whole dataset. In the ablation studies, we train one model per sequence. We use Adam optimizer with an initial learning rate of 0.0005 for the Deform-Net and the implicit template respectively. For Shapenet, the initial learning rate for the latent embedding is 0.001, while for DFaust, it is set to 0.0001. The decay factor is 0.5 for every 500 epochs. The total loss is defined as below:

$$\mathcal{L} = w_0\mathcal{L}_{sdf} + w_1\mathcal{L}_{\mathbf{c}} + w_2\mathcal{L}_{pp} + w_3\mathcal{L}_{pw} + w_4\mathcal{L}_{sp} + w_5\mathcal{L}_{ec}, \quad (1)$$

where $\mathcal{L}_{sdf}$ denotes SDF prediction loss, $\mathcal{L}_{\mathbf{c}}$ is the regularization term for latent code in an auto-decoder model, $\mathcal{L}_{pp}, \mathcal{L}_{pw}$ are point-pair and pointwise losses respectively, $\mathcal{L}_{sp}$ is the structural points prediction loss, $\mathcal{L}_{ec}$ is the explicit constraints loss. $w_0 \dots w_5$ are the loss weight hyperparameters. The explicit constraints are applied using 16 structural points on each warped shape after 500 epochs when the trained SP-Net is stable to generate accurate structural points. Detailed hyperparameters are shown in Table A.

II.2. Dataset Pre-processing

DIF-Net based test set. We follow DeepSDF [PFS*19] to prepare the SDF samples and split the train/test set in ShapeNet [CFG*15], and results are provided in Table 1 in the main paper. When compared to DIF-Net [DYT21], which also extracted SDF samples, and provides pre-trained models for planes, chairs and cars categories, we can directly evaluate the performance with the provided model and test set which contains 100 shapes in each category providing the points near/on the surface as well as their normals, results are presented in Table 2 in the main paper.

DFaust dataset preparation. DFaust [BRPB17] is a dynamic human shape dataset containing 129 sequences with various motions "shake shoulders", "one leg jump", etc. We pre-process the data in the same way as ShapeNet to obtain the SDF values and split the dataset into 105 training, 6 validation, and 9 test sequences following the practice of OFlow [NMOG19]. In contrast to OFlow, we do not require small deformations between shapes, we prepare the training set by randomly sampling the complete training sequences to 2040 shapes only. As for the test set, we extract SDF samples for all the shapes in the sequence, and evaluate the reconstruction and correspondences following the protocol in OFlow. Noticeably, OFlow provides the ground-truth shapes and correspondences with its data-processing procedure on the original DFaust dataset. For a fair comparison, we evaluate the reconstructed shapes of all methods with OFlow provided ground-truth shapes. In order to keep the scale consistency as in OFlow, we first transform the reconstructed shapes to the original DFaust objects scale, then rescale to OFlow provided ground-truth shapes scale.

II.3. Evaluation Details

Reconstruction. At inference time, a shape code $c_{\mathcal{S}}$ for test shape $\mathcal{S}$ is initialized randomly from $\mathcal{N}(0, 0.01^2)$, and then refined by minimizing the following objective:

$$\widehat{\mathsf{c}} = \operatorname{argmin} \mathcal{L}_{SDF} + w_1 \mathcal{L}_{\mathsf{c}}, \tag{2}$$

where $\mathcal{L}_{SDF}$ is formulated in Eq. (9) in the main paper. For each test shape we first optimize the latent code using Adam with a learning rate of 1×10^{-3} for 800 iterations. Then we apply Marching Cube [LC87] based on the predicted SDF values to obtain the reconstructed shape. We refer to Marching Cube [LC87] for detailed implementation. For training and testing DIT, we follow their original setup. Thus, a learning rate of 5×10^{-4} and 800 iterations are used.

Label Transfer. The source shapes we use for label transfer are selected from the test set in order (the first 5 shapes in the intersection of the test set and ShapeNetPart [MZC*19] dataset), displayed in Figure A.

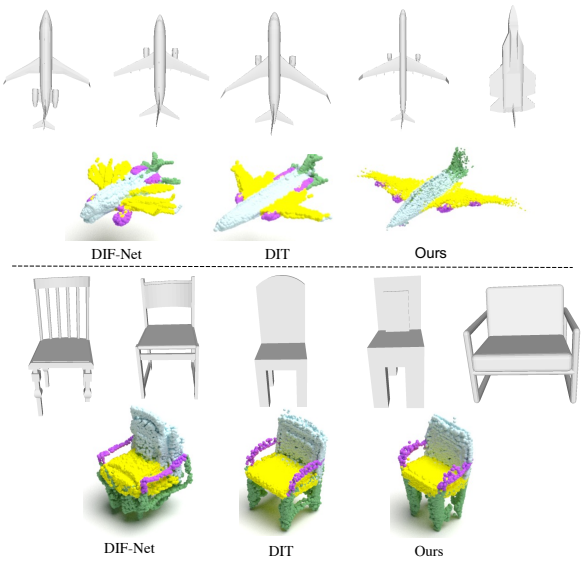

Figure A: *Source shapes for label transfer. The figure shows the input source shapes and their deformed labels in template space from different methods. The first row and the third row are 5 source shapes in airplanes and chairs categories. The second row and the last row are deformed labels in DIF-Net [DYT21], DIT [ZYDL21] and our methods respectively.*

III. Additional Experimental Results on ShapeNet

III.1. Template Consistency Results

We provide additional results on ShapeNet [CFG*15]. we evaluated template consistency in DIF-Net [DYT21], DIT [ZYDL21] and our methods by computing CD Temp $\ell 2$, results are displayed in Table B. We can observe that in all categories, our method can generate a more consistent template shape. Furthermore, We show some qualitative results of deformed shapes in chairs and sofas in Figure C where shapes have large variations. Our method is able to keep warp consistency when sofas shapes contain some structural variations, like the $2^{rd} - 4^{th}$ columns, while DIT fails in these shapes. However, chairs consisting of many large structural variations are quite challenging, we show the failure cases of both our method and DIT in the last two columns.

Method	Airplanes	Cars	Chairs	Sofas
DIF-Net [DYT21]	2.465	1.123	2.823	-
DIT [ZYDL21]	2.551	0.026	3.293	0.677
Ours	**1.400**	**0.024**	**1.806**	**0.609**

Table B: *Template consistency in airplanes, cars, chairs and sofas categories, evaluated with CD Temp $\ell 2$, multiplied by* 10^3.

III.2. Shape Interpolation

To explore the ability to interpolate two shapes, we present the experiment of interpolating m shapes between two random shapes $\mathcal{X}_1, \mathcal{X}_2$, each with their optimized global code $\mathsf{c}_1, \mathsf{c}_2$, and their local code $\mathsf{f}_1, \mathsf{f}_2$. We linearly interpolate in the latent space to generate m shapes. Thus, for shape $\mathcal{X}_i, i \in [0, m+1]$, its global latent code c_i can be computed

$$\mathsf{c}_i = \mathsf{c}_1 + \frac{(\mathsf{c}_2 - \mathsf{c}_1)}{(m+1)} \times i. \tag{3}$$

While for local code, we extracted the local context f from its closer input shape. That is,

$$\mathsf{f}_i = \begin{cases} \mathcal{N}(\mathsf{x}, \mathcal{X}_1), & if \quad i \leq (m+1)//2 \\ \mathcal{N}(\mathsf{x}, \mathcal{X}_2). \end{cases} \tag{4}$$

Here, x represents each point, $\mathcal{N}$ is the SP-Net [CLC*20]. We present further visualization results on shape interpolation ($m = 6$) in Figure B. The interpolated shapes are textured with their UV map based texture images. The same colored checkerboard with the same number on it denotes correspondences. However, given the fact that the local context is not accurately extracted from the interpolated shape, it will influence the correspondence accuracy. From Figure B, we see our method is still able to keep dense correspondences.

III.3. Runtime and Computation Complexity

We compare the model size and the runtime for training/inference in Table C. We set up our network parameters (1.89M) similar to the one used in DIT [ZYDL21](1.85M). When using only one single RTX 1080ti GPU, the time costs for training one shape in each epoch is 0.068s and 0.188s for DIT and ours respectively. Although We are slower in training time, we are faster at inference with fewer LSTM cells when reconstructing one shape.

Method	Model size (M)	Train time(s)	Inf time (s)
DIT [ZYDL21]	**1.85**	**0.0675**	48.39
Ours	1.89	0.188	**34.18**

Table C: *Model size and runtime comparison. The table shows the model size of DIT [ZYDL21] and ours. Moreover, the time costs for training and inference are presented.*

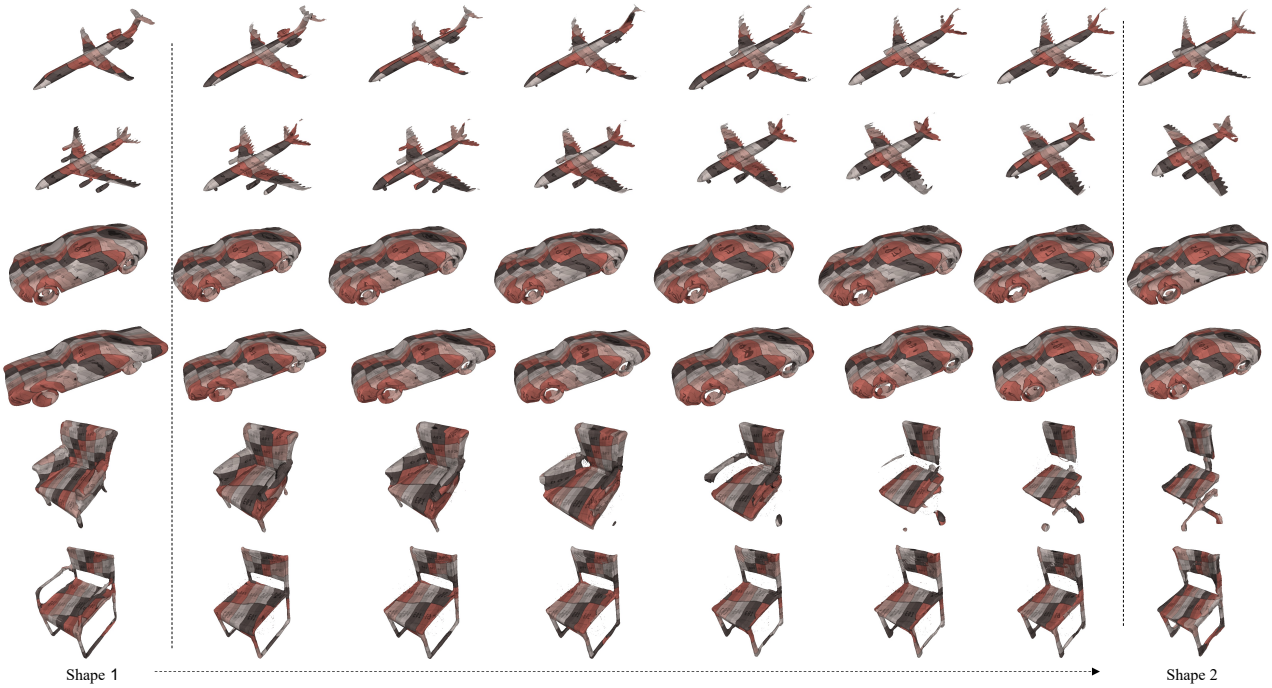

Figure B: *More examples of shape interpolation between "shape 1" to "shape 2" in ShapeNet airplanes, cars, chairs categories. The same colored checkerboard with the same number on it represents the corresponding mesh.*

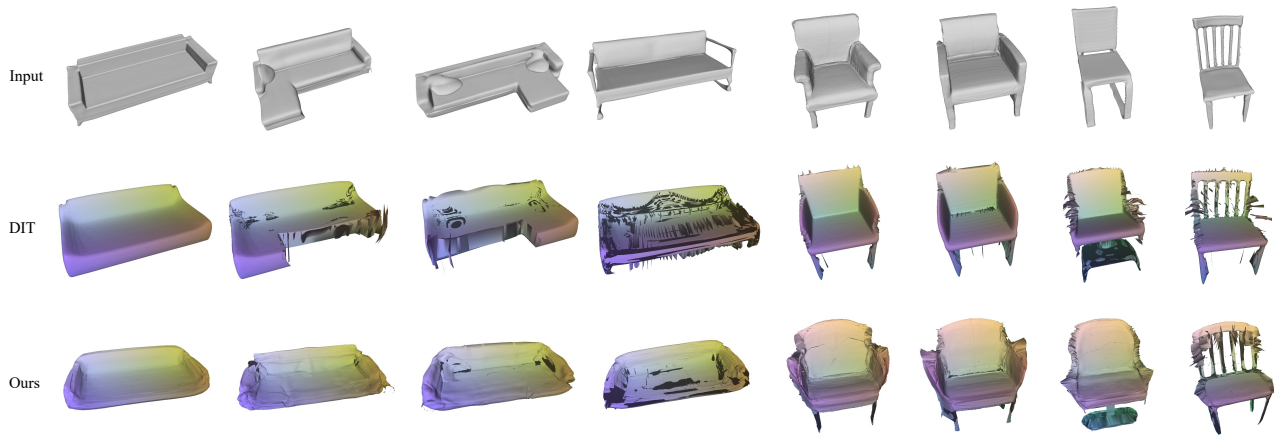

Figure C: *Qualitative results of warped shapes in sofas and chairs from different methods. The first row shows the input shapes, followed by the deformed shapes from DIT [ZYDL21] and our method. Our method can generate the consistent deformed shapes on sofas with large variations. However, both our method or DIT fails in generating a consistent template space in chairs due to its challenging structural changes.*

References

[BRPB17] BOGO, FEDERICA, ROMERO, JAVIER, PONS-MOLL, GERARD, and BLACK, MICHAEL J. "Dynamic FAUST: Registering human bodies in motion". *CVPR*. 2017 1, 2.

[CFG*15] CHANG, ANGEL X, FUNKHOUSER, THOMAS, GUIBAS, LEONIDAS, et al. "Shapenet: An information-rich 3d model repository". *arXiv preprint arXiv:1512.03012* (2015) 1, 2.

[CLC*20] CHEN, NENGLUN, LIU, LINGJIE, CUI, ZHIMING, et al. "Unsupervised learning of intrinsic structural representation points". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 9121–9130 1, 3.

[DYT21] DENG, YU, YANG, JIAOLONG, and TONG, XIN. "Deformed implicit field: Modeling 3d shapes with learned dense correspondence". *CVPR*. 2021 2.

[HS97] HOCHREITER, SEPP and SCHMIDHUBER, JÜRGEN. "Long short-term memory". *Neural computation* 9.8 (1997), 1735–1780 1.

[LC87] LORENSEN, WILLIAM E and CLINE, HARVEY E. "Marching cubes: A high resolution 3D surface construction algorithm". *ACM siggraph computer graphics* 21.4 (1987), 163–169 1, 2.

[MZC*19] MO, KAICHUN, ZHU, SHILIN, CHANG, ANGEL X, et al. "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding". *CVPR*. 2019 2.

[NMOG19] NIEMEYER, MICHAEL, MESCHEDER, LARS, OECHSLE, MICHAEL, and GEIGER, ANDREAS. "Occupancy flow: 4d reconstruction by learning particle dynamics". *ICCV*. 2019 2.

[PFS*19] PARK, JEONG JOON, FLORENCE, PETER, STRAUB, JULIAN, et al. "Deepsdf: Learning continuous signed distance functions for shape representation". *CVPR*. 2019 1, 2.

[QYSG17] QI, CHARLES R, YI, LI, SU, HAO, and GUIBAS, LEONIDAS J. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". *arXiv preprint arXiv:1706.02413* (2017) 1.

[ZYDL21] ZHENG, ZERONG, YU, TAO, DAI, QIONGHAI, and LIU, YEBIN. "Deep implicit templates for 3D shape representation". *CVPR*. 2021 1–4.