Neural Feature Matching in Implicit 3D Representations: *Supplementary Material*

Yunlu Chen [1] **Basura Fernando** [2] **Hakan Bilen** [3] **Thomas Mensink** [4 1] **Efstratios Gavves** [1]

A. Limitation of global shape matching error

We further clarify the inherent limitation of the global shape distance metric for measuring the shape deformation quality in the presence of inconsistencies in topology or semantics between the source and the target shapes.

The global metrics assign a low error, when the two shapes overlap significantly, even if this implies an unnatural fitting. Figure 1 is a characteristic example. With our feature matching, the source arms can only find the closest points on the seat or the back of the target chair, leading to a larger global fitting error; while, with cross-fitting, the arms are forced very close to the seat and the back in an unnatural and distorted manner, which, however, reduces the whole shape error. By contrast, part-level metrics do not count such errors with inconsistent semantics, which makes more sense when shapes differ significantly.

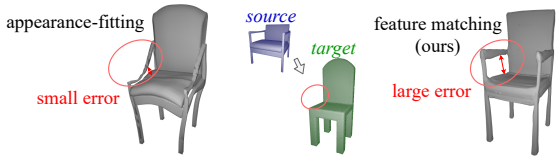

Figure 1. Limitation of global shape error metrics. Mesh deformed from source (blue) to target (green). The appearance-fitting result, generated with MeshODE (Huang et al., 2020), has a lower global matching error from the target shape at the arms (*e.g.* Chamfer distance), with an unnatural fitting.

B. Interpolating mesh deformation

Some additional visual results are provided in Figure 2 for chairs and Figure 3 for shapes in other categories, with deformed shapes from intermediate time steps. We show smooth and meaningful interpolated shapes as our method transfers the vertices of the shape mesh continuously.

[1]Informatics Institute, University of Amsterdam, the Netherlands [2]AI3, IHPC, A*STAR, Singapore [3]School of Informatics, University of Edinburgh, Scotland [4]Google Research, Amsterdam, the Netherlands. Correspondence to: Yunlu Chen <y.chen3@uva.nl>.

C. Implementation Details

Analysis and mesh deformation (§4.1 and §4.2). Our implementation is based on IM-Net or IM-AE from Chen & Zhang 2019 with the codebase available at `https://github.com/czq142857/IM-NET-pytorch`, which is an improved implementation from the authors. We use the preprocessed ShapeNet dataset (Chang et al., 2015) available with the codebase. For evaluation of part-aware measures in Table 1 in the main paper, we take semantic part segmentation annotation from ShapeNetPart (Yi et al., 2016) dataset preprocessed by Chen et al. 2019 (available at `https://github.com/czq142857/BAE-NET`). For each of the shape categories we take the first 200 shapes from the test split, and deform the first shape to the second, the third to the fourth, until the 199th shape to the 200th.

The implicit decoder is 7-layer MLP with Leaky-Relu activation except that the last layer to the output is linear. The negative slope set for Leaky-Relu is -0.02. There are no batch normalization or other normalization layers. The widths of each layers $\{w_l\}$ from input to output are 259-1024-1024-1024-512-256-128-1. Note, $w_0 = 259$ is for the 3-dim input coordinates concatenated with 256-dim latent code. The encoder is a 5-layer 3D ConvNet that takes voxels of shapes as input. Each conv layer is followed by an instance normalization and Leaky-Relu activation (with negative slope -0.02). The widths are 1-32-64-128-256-256 and so the output is a 256-dim latent code.

We train one network per shape category with the same architecture, following the coarse-to-fine progressive training scheme from Chen & Zhang 2019 with the resolutions at 16^3, 32^3, 64^3 respectively for 100, 200 and 800 epochs. The batch size is 32. Adam optimizer is used with learning rate 0.00005. The supervised ℓ_1 loss is used for training. Training of each model takes around 30 hours on one single Nvidia Geforce 1080 Ti.

The optimisation for feature matching uses the following settings: we use $\mathrm{d}t = 0.02$ for a total of 50 intermediate steps with latent code interpolation. We use $N = 3$ Newton's iterations at each time step. The regularisation factor λ is set as 0.01. The entire feature matching process from one source mesh with 3000 vertices to a target shape takes

around 60 seconds. The bottleneck of runtime is mostly at the calculation of Jacobian, which requires iterating over the dimension in the hidden layer feature w_l in modern deep learning frameworks PyTorch or TensorFlow.

Inherent correspondence evaluation (§4.3). We use the released code from OccFlow (Niemeyer et al., 2019) available at `https://github.com/autonomousvision/occupancy_flow` for the preprocessed D-FAUST dataset (Bogo et al., 2017), the evaluation of the ℓ_2 error of the correspondence as well as the implementation of OccNet (Mescheder et al., 2019). The velocity network component is not used.

The implicit function OccNet contains sequentially 5 residual blocks. Each block is with two fully-connected layers followed by ReLU activation, with residual connection from the input to the output of the block. In total, the implicit decoder as 10 fully-connected layers. All hidden layer widths are 256. The input has 259 dimensions and the output is a scalar occupancy probability. The encoder is a PointNet (Qi et al., 2017) that takes point coordinate inputs and output a 256-dim latent code. Following the original setup from Niemeyer et al. 2019 in this evaluation, all vertices of the human shape are taken as the point inputs.

We train the OccNet to reconstruct human shapes with all poses from all training sequences, unlike Niemeyer et al. 2019 that trains the reconstruction network with only the poses in the first frame of each sequence. Other training details follow the original implementation. The batch size is 16. Adam optimizer is used with learning rate 0.0001. Training uses cross-entropy classification loss on the binary occupancy probability and takes around 5 days for 3000 epochs.

We match the last hidden layer implicit feature from OccNet as it outcomes points that are most close to the target shape surface, and find the nearest point on the target shape surface for correspondence. We use a total of 8 intermediate steps with latent code interpolation. The ℓ_2 error of the correspondence is evaluated on the test split with the author's code. For the number of Newton's iterations at each time step we use $N = 4$. The regularisation factor λ is set as 0.001.

More architectural and training details can be referred to the original implementations, since the architecture and training processes highly rely on the existing standard implicit function methods.

References

Bogo, F., Romero, J., Pons-Moll, G., and Black, M. J. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6233–6242, 2017.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019.

Chen, Z., Yin, K., Fisher, M., Chaudhuri, S., and Zhang, H. Bae-net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8490–8499, 2019.

Huang, J., Jiang, C. M., Leng, B., Wang, B., and Guibas, L. Meshode: A robust and scalable framework for mesh deformation. *arXiv preprint arXiv:2005.11617*, 2020.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5379–5389, 2019.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., and Guibas, L. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

Figure 2. Mesh deformation interpolation over time, chairs. Every two rows are a group of examples. The blue mesh is the source shape and the green mesh is the target shape. The odd row shows interpolation from source to target (*left* to *right*), and the even row from target to source (*right* to *left*).

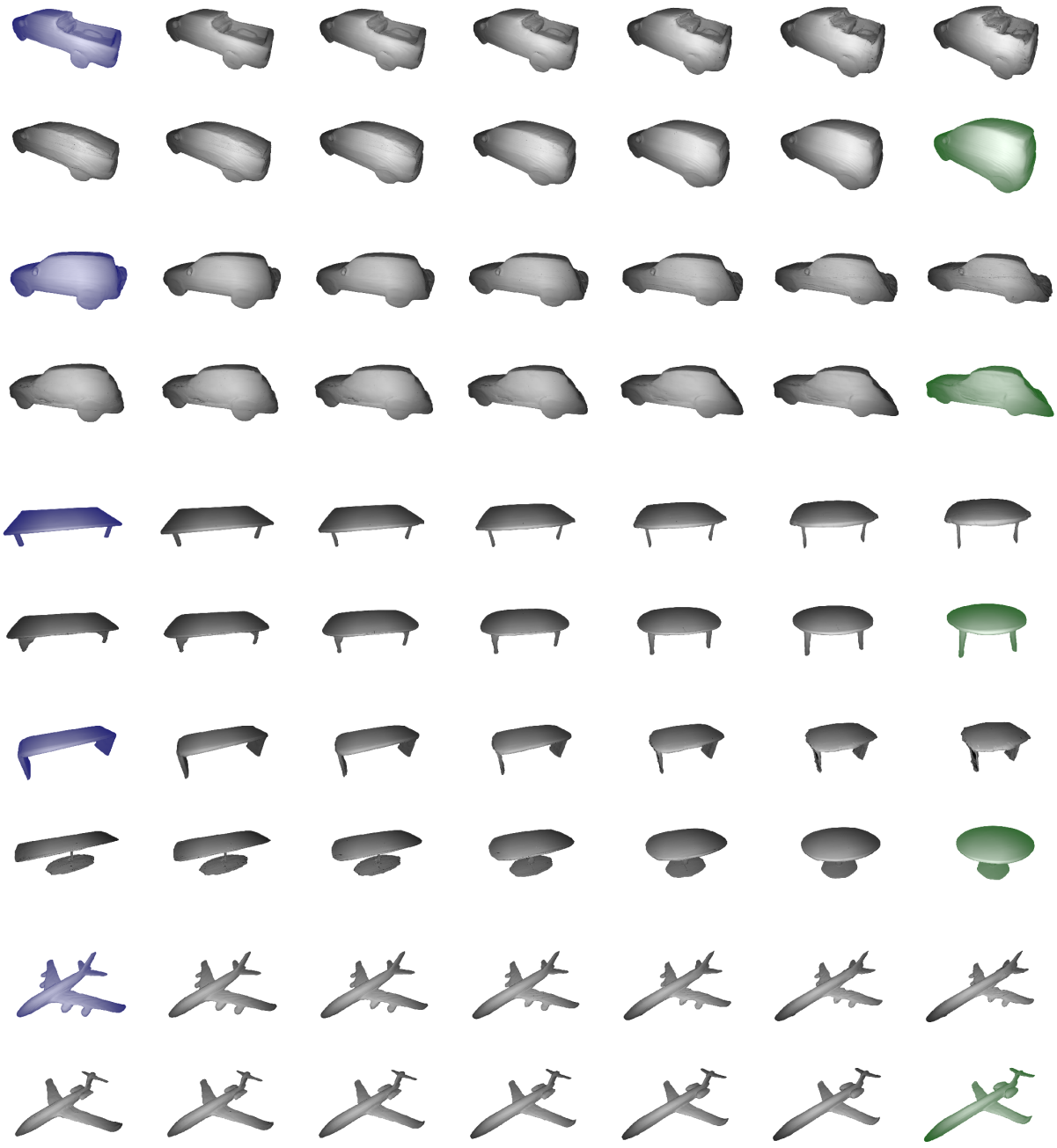

Figure 3. Mesh deformation interpolation over time, other categories. Every two rows are a group of examples. The blue mesh is the source shape and the green mesh is the target shape. The odd row shows interpolation from source to target (*left* to *right*), and the even row from target to source (*right* to *left*).