MvDeCor: Multi-view Dense Correspondence Learning for Fine-grained 3D Segmentation

Gopal Sharma[1⋆], Kangxue Yin[2], Subhransu Maji[1], Evangelos Kalogerakis[1], Or Litany[2], and Sanja Fidler[2,3,4]

[1] University of Massachusetts, Amherst
[2] NVIDIA
[3] University of Toronto
[4] Vector Institute

A Supplementary Material

Here we further provide the following supplementary information and results:

- Samples from training datasets
- Experiment on the Shapenet dataset
- Additional qualitative results on RenderPeople, and details of annotation for the dataset
- Training details of our approach and baselines
- Data licenses, and discussion on societal impact and human dataset.

A.1 Samples from training datasets

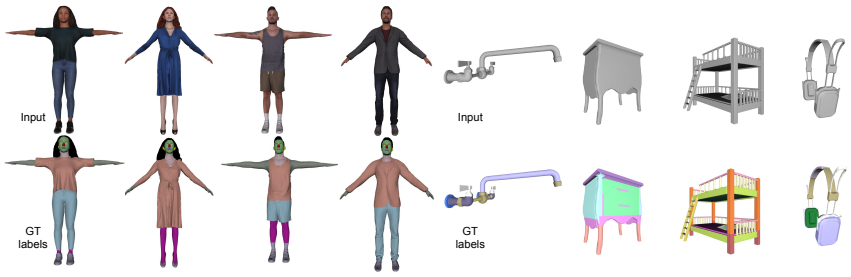

Fig. 1. Examples from datasets used in our experiments. *Left*: RenderPeople [3] dataset. *Right*: PartNet [9] dataset

Figure 1 visualizes different samples from our training dataset used in our experiments.

⋆ The work was mainly done during Gopal's internship at NVIDIA

Methods	instance avg. mIOU	class avg. mIOU
SO-Net [8]	64.0	-
PointCapsNet [14]	67.0	-
MortonNet [10]	-	-
JointSSL [4]	71.9	-
Multi-task [7]	68.2	-
Deformation [12]	68.9	66.2
PointContrast [13]	74.0	-
ACD [6]	**75.7**	74.1
2D Scratch	72.7	74.7
(2D+3D) Ours	74.3	**75.8**

Table 1. Comparison with state-of-the-art few-shot part segmentation methods on ShapeNet. Performance is evaluated using *instance-averaged* and *class-averaged* mIOU while using 1% of the training data.

A.2 Experiment on Shapenet dataset

The main focus of our work is fine-grained semantic segmentation. We also experiment with the Shapenet Semantic Segmentation dataset [5] for the task of few-shot semantic segmentation, which consists of 16,881 labeled point clouds across 16 shape categories, with a total of 50 part categories. We transfer the point labels to triangles of a mesh using nearest neighbor queries to train our models. The evaluation is done by transferring the predicted triangle labels back to original point cloud. We use the same architecture and training strategy for this dataset as used for other datasets. We report our results in Table 1. Note that instance mIOU is highly influenced by the shape categories with large number of testing shapes Chair, Table. Class mIOU, on the other hand gives equal importance to all categories, hence it is a more robust evaluation metric. We evaluate the performance of work by Wang *et al.*[12] all all shape categories from this dataset and average the performance over 5 random runs.

A.3 Additional results on RenderPeople and annotation details

In Figure 2, we provide additional qualitative results on the RenderPeople dataset.

In the data annotation stage, we label Renderpeople shapes using the labeling tool from [11]. We start by rendering multiple RGB images of the textured mesh such that maximum surface area can be covered. Then we label each rendered image and back-project the pixel labels to the surface. We label 13 different parts as shown in Figure 3.

A.4 Training details

Training details for PartNet dataset. For pre-training and fine-tuning stages of our method we use the Adam optimizer with 0.001 learning rate. For pre-training we decay the learning rate by half when validation loss saturates. During pre-training, we use $4k$ matched pairs of points for a pair of views to compute our self

supervision loss. During pre-training on the PartNet dataset, we train our model with batch size of 16 for $200k$ iterations. For fine-tuning, we use the batch size of 8 and exponential learning rate decay (factor=0.99) after every 40 iterations. For $k = 10$, $v = all$ setting, we train our model for $4k$ iterations, and for $k = 10$ and $v = 5$ setting, we train our model for $2k$ iterations.

Training details for RenderPeople dataset. For pre-training and fine-tuning stages of our method, we use the Adam optimizer with 0.001 learning rate. For pre-training, we decay the learning rate by half when validation loss saturates. During pre-training, we use $4k$ matched pairs of points for a pair of views to compute our self supervision loss. During pre-training for the RenderPeople dataset, we train our model with batch size of 16 for $100k$ iterations. For fine-tuning, we use the batch size of 8 and exponential learning rate decay (factor=0.99) after every 40 iterations. For the RenderPeople dataset for $k = 5$, $v = all$ setting we train our model for $2K$ iterations, and for $k = 5$ and $v = 3$ setting, we train our model for 400 iterations.

DeepLabv3+. We use the DeepLabV3+ as our 2D CNN backbone for learning pixel level features. We modify the last layer of DeepLabV3+. In the original version, the (64×64) feature map is directly $4\times$ upsampled to a (256×256) feature map using bilinear interpolation, since the input image has a size of $(256 \times 256 \times 3)$. We instead gradually upsample the (64×64) feature map to (256×256) resolution in two upsampling stages to preserve fine-grained details in the following way: $\texttt{Up}(2) \rightarrow \texttt{BN}(256) \rightarrow \texttt{Relu} \rightarrow \texttt{Conv2D}(256, 128, 3) \rightarrow \texttt{Up}(2) \rightarrow \texttt{BN}(256) \rightarrow \texttt{Relu} \rightarrow \texttt{Conv2D}(128, 64, 3)$. We also use bilinear up-sampling. $\texttt{Conv2D}(i, o, k)$ is a 2D convolution layer with i input channels, o output channels and k kernel size, `Relu` is rectified linear unit, $\texttt{Up}(x)$ is bilinear up-sampling by a factor of x and `BN` is a batch normalization layer.

DenseCL. We keep all the hyper parameters same as proposed in the original work. When depth map and normal maps are also input to the network, the spatial augmentations applied to the RGB image are also applied to the normal and depth maps. We do not augment normal and depth maps in any other way. The models are trained until convergence. Once the DenseCL baseline is pre-trained using their proposed approach on our dataset, we use the backbone ResNet weights to initialize our DeepLabv3+ architecture as described above and add a 2D convolution layer as a segmentation head.

PointContrast. To implement our 3D baseline, we use a 3D ResNet with U-Net based architecture with 42 layers as proposed in the original paper [13]. We use a voxel size of 0.01. We use a batch size of 16 and $10k$ pairs of matched points to compute their self supervision loss. The implementation of the loss is done using the source code provided by the authors. We use the SGD optimizer with learning rate 0.1 with 0.9 momentum and 0.0001 weight decay. We train this model for $100k$ iterations. The validation loss saturates after $100k$ iterations.

A.5 Dataset, code, and ethics discussions

Dataset and code licenses The PartNet dataset [9] is a collection of labeled shapes from ShapeNet [5]. The license can be found on the website of ShapeNet. We obtained the license for using the Renderpeople dataset [3] through an agreement with Renderpeople. To run the comparison with baseline methods, we use the source code provided by the authors of DenseCL [1], PointContrast [2]. The licenses of the codebases are provided in their original GitHub repositories.

Potential negative societal impacts We present a method for labeling detailed parts of 3D models given a provided training set of shapes. Like many other learning-based methods, our results can be biased by training datasets. For purpose of deploying the method for human shapes, one would need to carefully de-bias the dataset to cover the distribution of a wide range of body shapes, clothing, skin tones, race, and gender.

Personal data and human subjects Our paper uses human 3D models from Renderpeople for training and evaluation. The data collection and ethics approvals were taken care of by the dataset provider. We carefully inspected the dataset and did not find identifiable information or offensive content. More information about the dataset can be found on the websites of the data provider.

References

1. Densecl. https://github.com/WXinlong/DenseCL 4
2. Pointcontrast. https://github.com/facebookresearch/PointContrast 4
3. Renderpeople. https://renderpeople.com/ 1, 4
4. Alliegro, A., Boscaini, D., Tommasi, T.: Joint supervised and self-supervised learning for 3d real world challenges. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 6718–6725. IEEE Computer Society (2021) 2
5. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q.X., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. CoRR **abs/1512.03012** (2015) 2, 4
6. Gadelha, M., RoyChowdhury, A., Sharma, G., Kalogerakis, E., Cao, L., Learned-Miller, E., Wang, R., Maji, S.: Label-efficient learning on point clouds using approximate convex decompositions. In: European Conference on Computer Vision (ECCV) (2020) 2
7. Hassani, K., Haley, M.: Unsupervised multi-task feature learning on point clouds. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8160–8171 (2019) 2
8. Li, J., Chen, B.M., Hee Lee, G.: So-net: Self-organizing network for point cloud analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9397–9406 (2018) 2
9. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Computer Vision and Pattern Recognition (CVPR) (2019) 1, 4

Fig. 2. Visualization of predicted semantic labels on Renderpeople dataset in the few-shot setting when $k = 5$ fully labeled shapes are used for fine-tuning. We visualize the predictions of all baselines. To visualize the details of predicted segmentations in the facial region, we provide an inset figure.

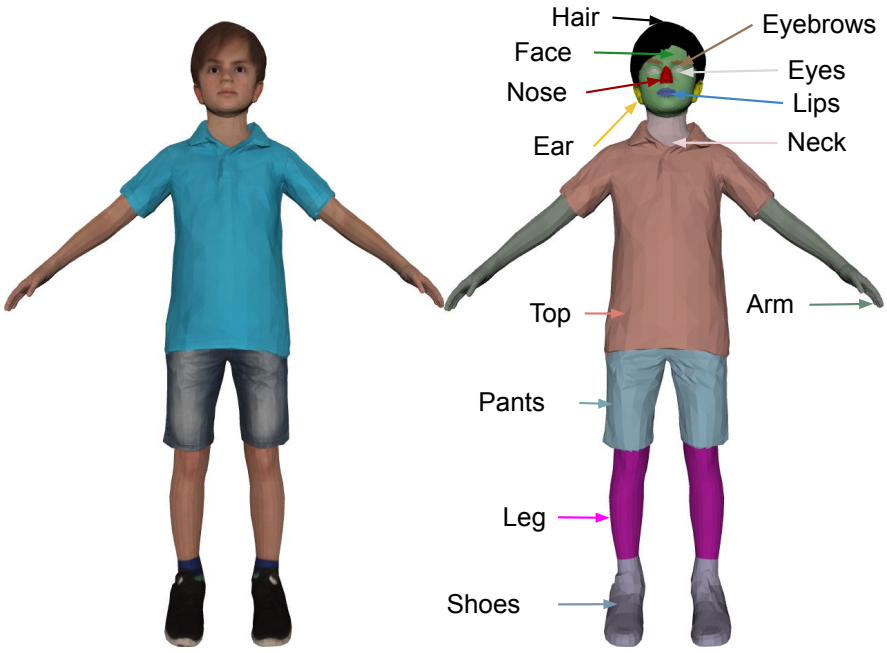

Fig. 3. Semantic labels of a shape from the RenderPeople dataset.

10. Thabet, A., Alwassel, H., Ghanem, B.: MortonNet: Self-Supervised Learning of Local Features in 3D Point Clouds. arXiv (Mar 2019), https://arxiv.org/abs/1904.00230 2
11. Wada, K.: labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme (2016) 2
12. Wang, L., Li, X., Fang, Y.: Few-Shot Learning of Part-Specific Probability Space for 3D Shape Segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4504–4513 (2020) 2
13. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: European Conference on Computer Vision. pp. 574–591. Springer (2020) 2, 3
14. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3D point capsule networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1009–1018 (2019) 2