Supplementary Material:
NEF: Neural Edge Fields for 3D Parametric Curve Reconstruction from Multi-view Images

Yunfan Ye Renjiao Yi Zhirui Gao Chenyang Zhu Zhiping Cai Kai Xu*
National University of Defense Technology

1. Statistical Analysis for ABC-NEF dataset

We present more statistics of the contributed ABC-NEF dataset, which consists of 115 distinct and complicated CAD models. Each model can be described by its topology (edges and vertices) as well as the geometry (surfaces and curves). *Edges* are the oriented connections between 2 vertices, with the most to be *sharp edges* where normal changes sharply; *Vertices* are the basic entities, corresponding to points in space. We refer to the original ABC dataset [3] for more detailed explanations.

Therefore, we illustrate the distribution of all mentioned attributes in Fig. 1. The selected models all contain only one part, with medium size of a proper number of vertices n ($10000 < n < 30000$). The major types of edge and surface are line and plane, respectively. We also present the histogram of the vertice, edge and sharp edge numbers in Fig. 1, to give an impression of the complexity and variety of the dataset. The distribution of the ABC-NEF dataset is close to the original ABC dataset [3], but as a new benchmark for 3D parametric curve reconstruction, ours focus more on commonly seen objects of medium size with more sharp edges.

2. Additional Experiments

Except for this PDF, we also provide several examples for inference using the provided code in the folder "NEF_test". The video demo "NEF-video-demo.mp4" also contain 10 examples of the rendered images, detected 2D edge maps, re-rendered 2D edge maps, extracted 3D edge points and reconstructed 3D parametric curves.

Here we provide more training details in Sec. 2.1, experimental results including the ablation study about the required number of views in Sec. 2.2, and more comparisons with state-of-the-arts in Sec. 2.3.

*Corresponding author.

2.1. Training Details

Our method is implemented in the Pytorch [7] environment and its neural network API PyTorch Lightning [1]. We sample 1024 rays per batch and train our model for 6 epochs (about 46k iterations) with Adam optimizer [2] and the learning rate of 5×10^{-4}. We use a threshold of 0.7 to extract point cloud edges from the learned neural edge field with a grid size of 256. When optimizing all parametric curves, we set $d = 4$ in to connect endpoints that are already close enough with a learning rate of 0.5. All experiments of our method are conducted on a single NVIDIA RTX3080Ti GPU.

2.2. Ablation Study

In the proposed ABC-NEF dataset, we sample 50 views for each object by evenly placing cameras on a sphere. Here we conduct an extra ablation study about the required number of views to train neural edge fields (NEF) properly, where the vanilla NeRF [6] requires about 100 views. We train the NEF with 5, 10, 30 and 50 views respectively (all evenly distributed) until convergence. As in the main paper, we also observe the spatial distribution of edge density by illustrating rendered depth maps for better visualization. As demonstrated in Fig. 2, 5 views are not enough to cover the whole object, and thus cannot get complete and clear edge densities. 10 views can already recover the geometrical shapes for simple cases, but may miss several curves or generate extra noise in objects with relatively complicated shape structures (e.g. the last two rows in Fig. 2). 30 views and 50 views are both complete and identical to the real geometrical shape for most cases, which are satisfactory enough.

Considering that the results of 50 views are slightly clearer, and the time consumption is close for training NEF by 30 views and 50 views until convergence, we finally decide to sample a unified number of 50 views for all cases for better performance, although $10-30$ views are enough for most simpler cases.

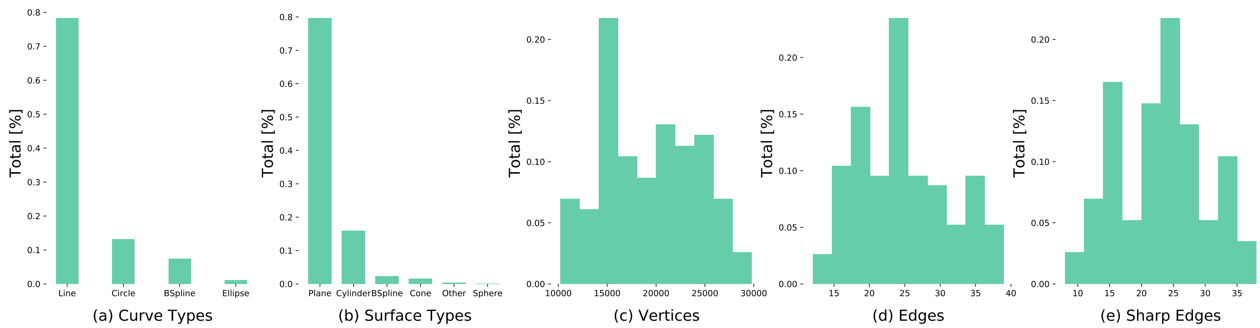

Figure 1. Each model in our dataset is composed of multiple surfaces and feature curves. The first two images show the distribution of types of curves (a) and surfaces (b) over the current ABC-NEF dataset. Histograms over the numbers of vertices (c), edges (d) and sharp edges (e) are presented in last three images. Most edges of the selected models are sharp edges in ABC-NEF dataset, which is qualified as a benchmark of 3D parametric curve reconstruction.

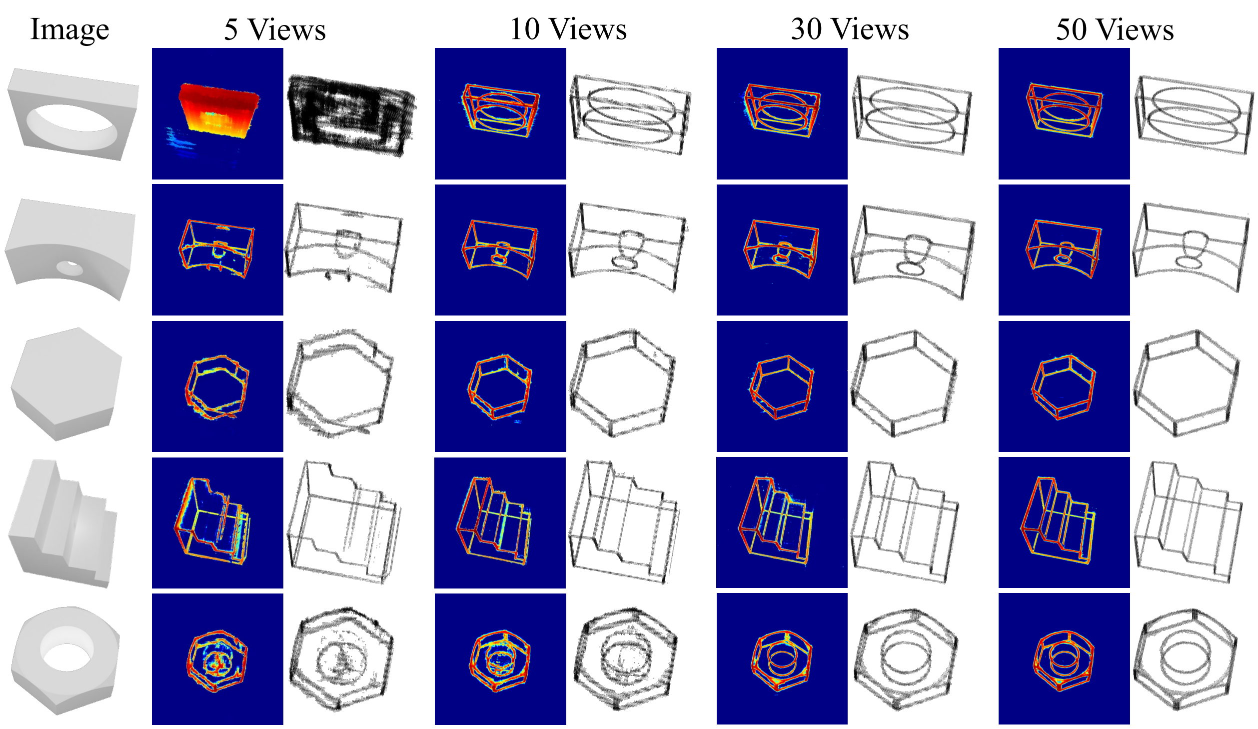

Figure 2. From left to right, we present 2D images in a given view, followed by the rendered depth map and extracted 3D edge points from NEF of 5, 10, 30 and 50 views respectively. Rendered depth maps convey the spatial distribution of the edge density field, and 3D edge points show the extracted geometrical shape. For simple cases, results of 10 views are close to satisfactory, while for complex cases (e.g. the last two rows), more views are required for better performance.

2.3. More Comparisons

We provide more qualitative comparisons with state-of-the-art methods of parametric curve reconstruction, including PIE-NET [10], PC2WF [4] and DEF [5]. The results are illustrated in Fig. 3.

3. Limitations

To foster additional works in this field, we briefly demonstrate several limitations of NEF, which are also potential directions for future work.

Training speed. Currently, it takes about one hour for NEF to train each model with 50 views, one can reduce the number of views to speed up with minor performance drops in most cases, as shown in 2.2. Also, the edge densities in spatial positions are highly sparse and could be accelerated by decreasing the samples along rays or integrating other voxel-based NeRF works for speedup. The coarse and fine optimization stages cost about 30 and 4 seconds on average, respectively.

Textured objects. 3D edges exactly lie in areas where normal changes sharply, while 2D edges also contain other edge types (e.g. shadow, surface texture). Objects with rich textures could bring much noise on 2D edge maps and consequently influence extracting 3D edge points and reconstructing curves. Those noisy edges could be suppressed from both the image level (classify which edge pixel is caused by texture discontinuity) and NEF level (recognize texture edge densities by locating object surfaces).

Edges inside the object. We cannot detect unseen edges hidden inside the object from only 2D images, and thus cannot reconstruct the corresponding curves. This is a natural drawback of our method, and could be tackled by integrating extra 3D cues (e.g. point cloud, mesh, shape prior).

References

[1] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. 1

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[3] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019. 1

[4] Yujia Liu, Stefano D'Aronco, Konrad Schindler, and Jan Dirk Wegner. Pc2wf: 3d wireframe reconstruction from raw point clouds. *arXiv preprint arXiv:2103.02766*, 2021. 2

[5] Albert Matveev, Ruslan Rakhimov, Alexey Artemov, Gleb Bobrovskikh, Vage Egiazarian, Emil Bogomolov, Daniele Panozzo, Denis Zorin, and Evgeny Burnaev. Def: Deep estimation of sharp geometric features in 3d shapes. *ACM Transactions on Graphics (TOG)*, 41(4):1–22, 2022. 2

[6] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, pages 405–421, 2020. 1

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[8] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[9] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5117–5127, 2021.

[10] Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. Pie-net: Parametric inference of point cloud edges. *Advances in neural information processing systems*, 33:20167–20178, 2020. 2

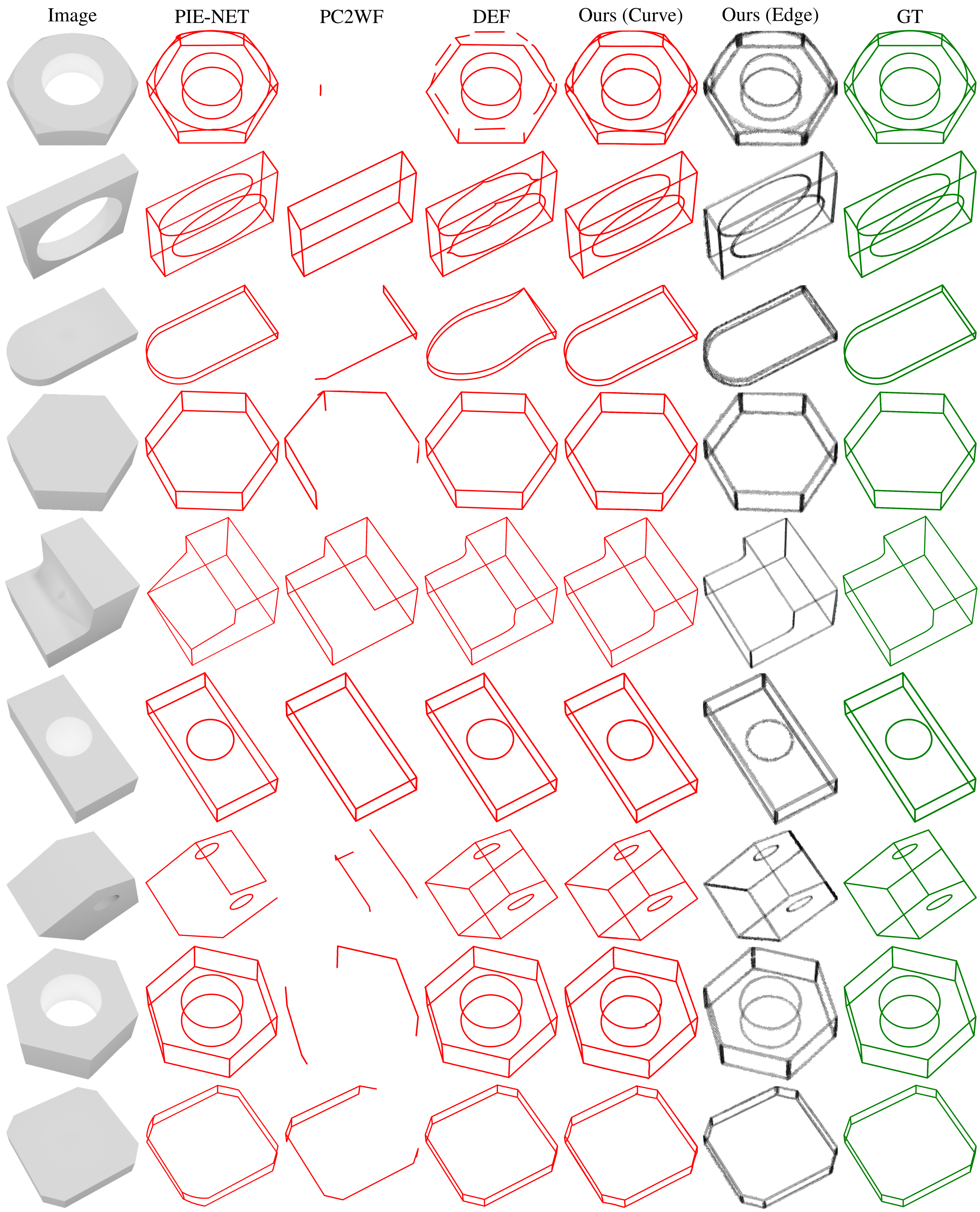

Figure 3. More qualitative comparisons against other state-of-the-arts. From left to right, we present the rendered image, the result curves of PIE-NET, PC2WF, DEF, our reconstructed curves, our 3D edge points obtained from edge densities, and the ground truth edges.