Supplementary Material
—SGCR: Spherical Gaussians for Efficient 3D Curve Reconstruction

Xinran Yang[1], Donghao Ji[1], Yuanqi Li[1], Jie Guo[1], Yanwen Guo[1*], Junyuan Xie[1]
[1]Nanjing University, Nanjing, China
{xryang,donghaoji}@smail.nju.edu.cn
{yuanqili,guojie,ywguo,jyxie}@nju.edu.cn

A. More Discussions on Anisotropic Gaussians v.s. Isotropic/Spherical Gaussians

The traditional *3D Gaussian Splatting* and its subsequent research mainly center on the technique of 'Splatting', which involves utilizing different anisotropic Gaussian splats to accurately represent the entire scene on a global scale. However, our work focuses on each 'Gaussian' itself, essentially displaying a reverse process: decomposing the entire 'Splatting' into small 'atoms' (atoms can form everything) and assigning convenient geometrical meanings to every individual Gaussian.

Let's draw an analogy in mathematics—a specific function $f(x)$ can be effectively approximated using polynomials—$x^0, x^1, x^2, x^k ... x^n$, each term with coefficient holds distinct algebraic property (anisotropy). But $f(x)$ can also be transformed in its Fourier expansion, where all terms share the same form of trigonometric functions (isotropy), with only variations in frequency. Uniform characteristics often reveal the deeper essence of things.

Our Spherical Gaussians choose a fixed radius to regulate the scale and rotation properties for all Gaussians, similar to resetting a basis function. Given the isotropic property, the gradients from the Gaussian rasterizer can be concentrated more on optimizing the splitting and positioning of each Gaussian. The global structure decomposed by these Gaussian bases is highly interpretable, with each Gaussian having a clear geometric meaning.

In fact, the degree of freedom in *3D Gaussian Splatting* is large, as one scene can be represented by completely different sets of Gaussians. By fixing specific properties of Gaussians and restricting the splatting gradients to optimizing other attributes (similar to base transformations intrinsically), the process of optimizing scene representation still works, but it results in different explanations. A thin and elongated Gaussian seems to be more efficient in representing curves, but it can be split into smaller 'basis-Gaussians' with few sacrifices but for better interpretation and convenience in certain tasks. Similarly, one can develop a sophisticated method to handle various anisotropic Gaussians for a specific task, but this inevitably come at the expense of robustness and generalization.

Currently, the full potential of explicit representation in 3D Gaussian Splatting has not been fully exploited. The core of our work is to provide meaningful interpretations to Gaussians through scenes, rather than overfitting scenes with arbitrary Gaussians. This transformation sets our work apart from others. Particularly, our method explores the utilization of isotropic bases—Spherical Gaussians—in curve reconstruction tasks, which turn out to be both efficient and structured. The isotropic and structured nature of Spherical Gaussians also provides significant advantages when correlated with today's advanced 3D manipulation techniques, such as point cloud processing methods (Gaussian attributes viewed as isotropic point features). Once the explicit representation property of 3D Gaussian Splatting aligns with appropriate geometrical meanings, it will spark great convenience for numerous tasks, far beyond curve reconstruction. We hope that our work can bring some insight to perceive the true sense and potential of *3D Gaussian Splatting*.

B. Visual Differences between Gaussians

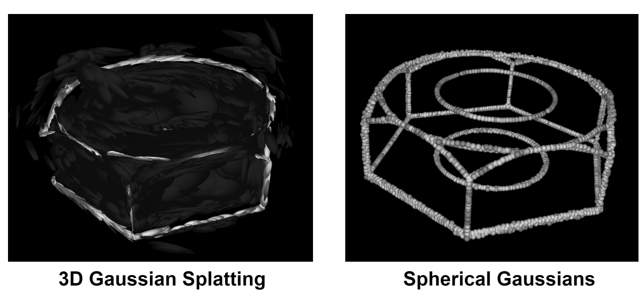

Figure 1. Visual differences between original 3D Gaussian Splatting [2] and our Spherical Gaussians.

*Corresponding author.

In our main paper, we have presented the quantitative results of ablation experiments on the design of Spherical Gaussians. Here, we visually compare the differences between the original 3D Gaussian Splatting [2] and our Spherical Gaussians. Results are shown in Figure 1. Although trained with edge maps, 3D Gaussian Splatting produces Gaussians in shape of various ellipsoids containing significant noise and low opacity values (the dark ellipsoids), making it quite inconvenient for geometric reconstruction tasks. However, our Spherical Gaussians are well aligned and neat, preserving the details of 3D structures and making edge reconstruction quite convenient. This accounts for the significant enhancement in 3D edge reconstruction metrics (as presented in Table 2 of our main paper). Spherical Gaussians may serve as a great tool for connecting 2D and 3D modalities, and further exploration into their capabilities is worthwhile.

We provide another experimental analysis on anisotropic Gaussians that is thin-shaped with one axis having small covariance. Fig. 2 illustrates the comparison results. Isotropic Gaussians provide a more favorable point cloud distribution, yielding better results in our curve fitting stage. Anisotropic Gaussians are featured in image rendering, but are less convenient for geometry reconstruction tasks.

C. Pseudo Code of Global Optimization

We present the pseudocode of *Part2—Global Optimization* of our SGCR method in the main paper. It is shown in Algorithm 1. This part takes the line endpoints from *Part1—Line Fitting* and our Spherical Gaussians as input, and optimizes them into control points (with weights) of 3rd order rational Bézier curves to achieve the final 3D edge reconstrution.

Algorithm 1 Part2—Global Optimization

Input: a set of Spherical Gaussians G with radius r_0, and the endpoints results $L = \{(p_i, q_i)\}_{i=1}^{|L|}$ from Part 1.

Output: the set of control points $C = \{\{z_i^j, w_i^j\}_{j=1}^4\}_{i=1}^{|L|}$ of 3rd order rational Bézier curves.

Denote: $B(u)_{p,w}$ is the function of rational Bézier curve.

1: $L_{all} = \emptyset$
2: **for** $i = 1$ to $|L|$ **do**
3: $(p_i, q_i) = L_i, \omega_i = (1, 1, 1, 1)$
4: $z_i = (p_i, \frac{3}{4}p_i + \frac{1}{4}q_i, \frac{1}{4}p_i + \frac{3}{4}q_i, q_i)$
5: $L_i = interpolate(B(u)_{z_i,w_i}, N_s) + r_0 * N(0, 1)$
6: $L_{all} = L_{all} \cup L_i$
7: **end for**
8: $C = \underset{\{\{z_i^j, w_i^j\}_{j=1}^4\}_{i=1}^{|L|}}{\arg\min} \mathcal{L}_{WCD}(L_{all}, P(G)) + \lambda \mathcal{L}_{endpoints}$
9: **return** C;

D. More Discussions on Opacity-Color Loss

The opacity attribute of isotropic Gaussians can effectively encode the concept of "edge density." *Opacity-color loss* helps to keep consistency between edge density and color intensity during multi-view optimization. This mechanism ensures that occluded edges—represented by Gaussians with low opacity values—are less susceptible to premature pruning during optimization, resulting in more complete 3D reconstructions. Fig. 3 illustrates the visual ablation results on opacity-color loss.

E. Metrics

To compute the metrics, we densely sample points from ground-truth edges (denoted as $\mathbf{X}_{gt}$) and the predicted curves (denoted as $\mathbf{X}_{pd}$).

Chamfer Distance (CD). The Chamfer distance is computed using:

$$CD = \frac{1}{2}(Comp. + Acc.),$$

$$Comp. = \frac{1}{|\mathbf{X}_{gt}|} \sum_{x_{gt} \in \mathbf{X}_{gt}} \min_{x_{pd} \in \mathbf{X}_{pd}} ||x_{gt} - x_{pd}||,$$

$$Acc. = \frac{1}{|\mathbf{X}_{pd}|} \sum_{x_{pd} \in \mathbf{X}_{pd}} \min_{x_{gt} \in \mathbf{X}_{gt}} ||x_{pd} - x_{gt}||$$

F-Score (FS). The F-Score is defined as follows:

$$FS = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall},$$

where

$$Precision = \frac{|\{x_{pd} \in \mathbf{X}_{pd} | \min_{x_{gt} \in \mathbf{X}_{gt}} ||x_{gt} - x_{pd}|| < \xi\}|}{|\mathbf{X}_{pd}|},$$

$$Recall = \frac{|\{x_{gt} \in \mathbf{X}_{gt} | \min_{x_{pd} \in \mathbf{X}_{pd}} ||x_{pd} - x_{gt}|| < \xi\}|}{|\mathbf{X}_{gt}|}$$

We use $\xi = 0.02$ for all experiments.

Intersection over Union (IoU). Noting TP (resp. FP and FN) the number of true positive, i.e. the number of points correctly predicted as full (resp. the number of points wrongly predicted as full, and the number of points wrongly predicted as empty), the IoU is defined as follows:

$$IoU = \frac{TP}{TP + FP + FN}$$

F. Evaluation on 2D Edge Detectors

Our method requires edgemaps obtained by applying 2D edge detectors on multi-view input images. The quality of 2D edge detection will affect our final 3D edge reconstruction results. Therefore, we also conduct experiments

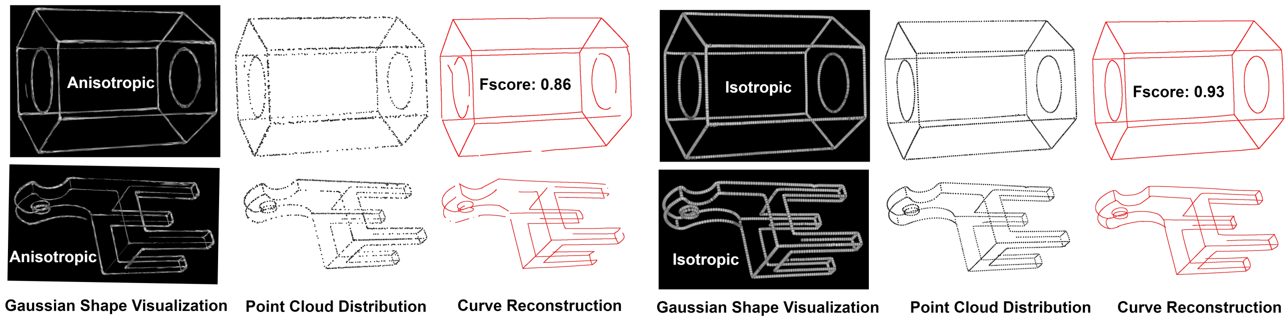

Figure 2. Comparisons between anisotropic Gaussians and Spherical Gaussians

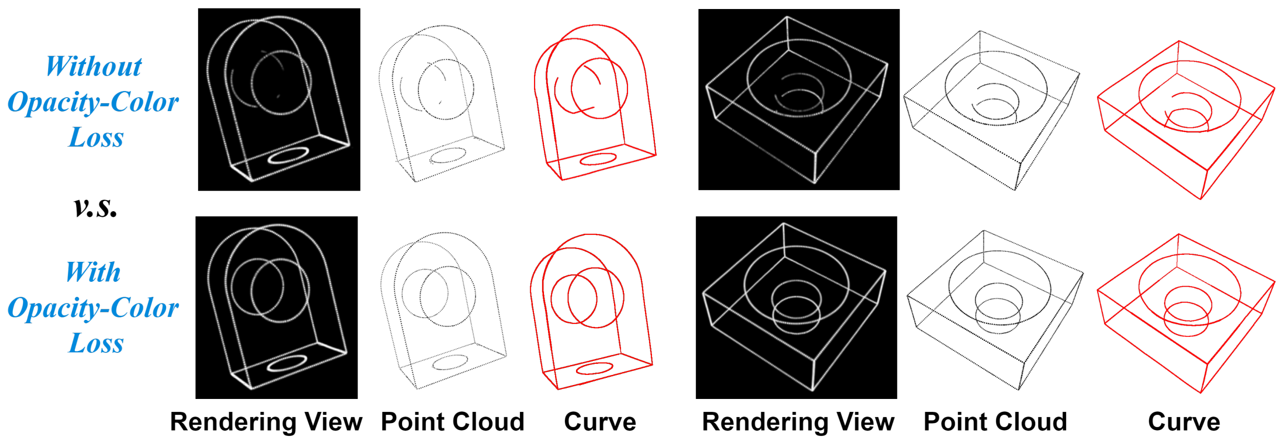

Figure 3. Visual ablation results on opacity-color loss

to evaluate our performance under different 2D edge detectors. We choose three state-of-the-art 2D edge detection methods—DexiNed [6], PiDiNet [7] and MuGE [11] to generate edgemaps on the same input images from ABC-NEF [10] dataset. Our method reconstructs similar results across these three 2D edge detectors, and the detailed quantitative comparisons are shown in Tab. 1. More advanced 2D edge detection methods (like MuGE) will also benefit our method.

For fair comparisons with NEF [10] and EMAP [5], we choose PiDiNet [7] as the main edge detector for experiments in our main paper.

Table 1. Quantitative comparisons on different 2D edge detectors. Methods includes DexiNed [6], PiDiNet [7] and MuGE [11].

Edge Detector	CD↓	Precision↑	Recall↑	F-score↑	IoU ↑
DexiNed [11]	0.0276	0.9686	0.8748	0.9159	0.8383
PiDiNet [7]	0.0280	0.9546	**0.9052**	0.9260	0.8615
MuGE [6]	**0.0250**	**0.9787**	0.8955	**0.9324**	**0.8643**

G. Parameters for Different Data Scale

Our basic parameter settings (e.g., $r_0 = 0.005$, $\delta_1 = 0.02$) are designed for unit cube spaces $[0, 1]^3$ based on the ABC-NEF[10] dataset. For larger scenes with a boundingbox N, we can either downscale the scene and adopt basic settings, or simply scale these parameters, e.g. $r'_0 = N \cdot r_0$. This modification works well across other datasets, as demonstrated by our results.

H. More Comparisons

More visual comparisons on ABC [4] and Modelnet [8] models are shown in Fig. 4.

We also test our method on the more challenging real-world outdoor scenes from Tanks&Temples [3] dataset. All previous methods have failed except EMAP [5]. The visual results are illustrated in Fig. 5. Our method performs significantly better than EMAP [5]. EMAP takes about 12 hours for training a single scene in Tanks&Temples [3], while our approach finishes within 3 minutes.

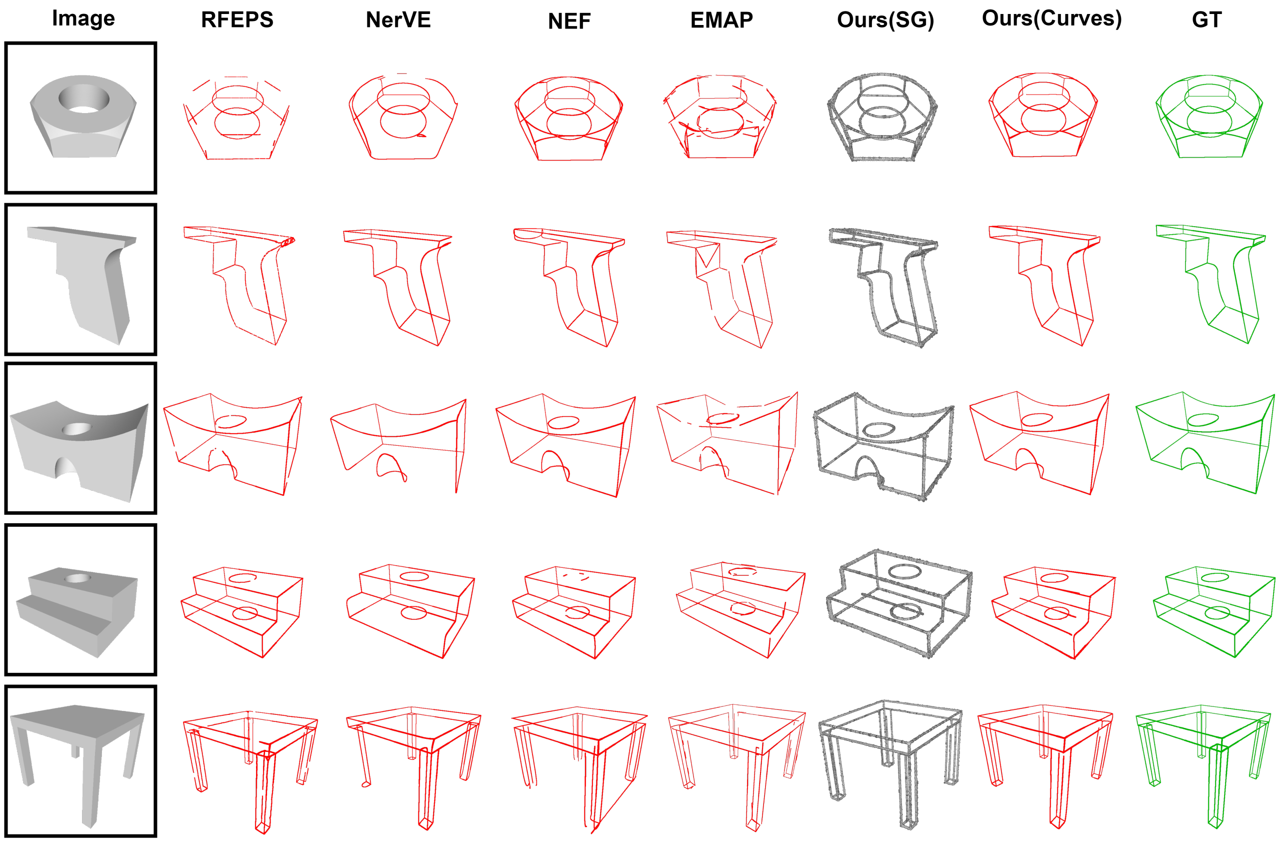

Figure 4. More qualitative comparisons results on curve reconstruction. From left to right, we present the rendered image, the results of RFEPS [9], NerVE [12], NEF [10], EMAP [5], our Spheical Gaussians, our final curves and the ground truth edges. Models come from ABC-NEF [10] dataset and ModelNet [8] dataset.

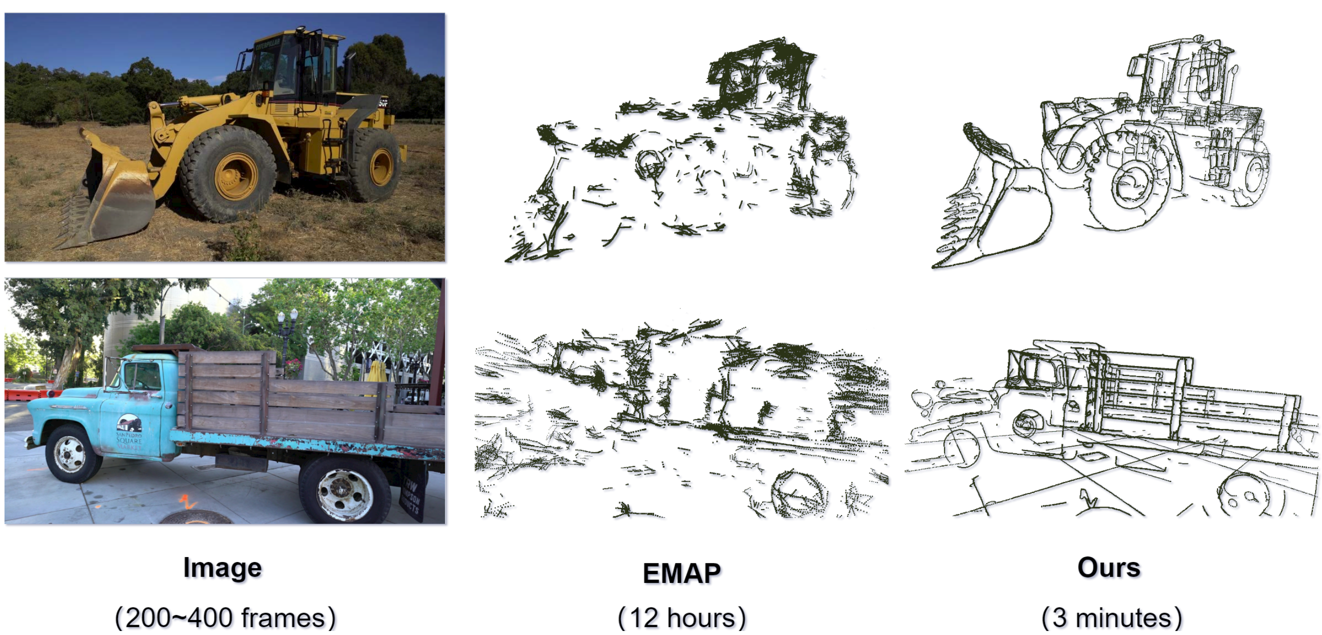

Figure 5. Visual results of edge reconstruction on the challenging real-world outdoor scenes from **Tanks&Temples [3] dataset**.

I. Limitations

In order to encourage further research in this area, we will discuss some limitations of our method and also suggest potential directions for future exploration.

Camera calibration. Although our method is robust to minor errors in camera poses, when it comes to reconstructing real scenes, the inaccurate estimation of camera poses will still bring drawbacks on generated Spherical Gaussians as well as the following reconstruction. It would be worthwhile to explore pose-free methods (as suggested in [1]) for training Gaussians to make the reconstruction process more robust and convenient.

Textured object. 3D edges are exactly located in areas where the normal changes abruptly, while 2D edges encompass a broader range of edge types, such as shadow and surface texture. Objects with intricate textures may introduce noise on 2D edge maps, which can in turn affect the distribution of Spherical Gaussians and the accuracy of curve reconstruction. The noise can be reduced by identifying which edge pixels are caused by texture discontinuity in 2D level and by locating object surfaces to distinguish Spherical Gaussians that represents textures in 3D level.

Inner edges. Since our method relies solely on 2D supervision, it is unable to produce Spherical Gaussians for hidden edges within the object that are not visible, thus limiting the reconstruction of inner curves. To overcome this limitation, additional 3D cues, such as point clouds, meshes, or shape priors, can be integrated. As an explicit representation, Spherical Gaussians have the potential to tackle multimodal tasks effectively.

References

[1] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. *arXiv preprint arXiv:2312.07504*, 2023. 5

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2

[3] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 3, 4

[4] 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. 3

[5] Lei Li, Songyou Peng, Zehao Yu, Shaohui Liu, Rémi Pautrat, Xiaochuan Yin, and Marc Pollefeys. 3d neural edge reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3, 4

[6] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1923–1932, 2020. 3

[7] 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. 3

[8] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 3, 4

[9] Rui Xu, Zixiong Wang, Zhiyang Dou, Chen Zong, Shiqing Xin, Mingyan Jiang, Tao Ju, and Changhe Tu. Rfeps: Reconstructing feature-line equipped polygonal surface. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. 4

[10] Yunfan Ye, Renjiao Yi, Zhirui Gao, Chenyang Zhu, Zhiping Cai, and Kai Xu. Nef: Neural edge fields for 3d parametric curve reconstruction from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8486–8495, 2023. 3, 4

[11] Caixia Zhou, Yaping Huang, Mengyang Pu, Qingji Guan, Ruoxi Deng, and Haibin Ling. Muge: Multiple granularity edge detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25952–25962, 2024. 3

[12] Xiangyu Zhu, Dong Du, Weikai Chen, Zhiyou Zhao, Yinyu Nie, and Xiaoguang Han. Nerve: Neural volumetric edges for parametric curve extraction from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13601–13610, 2023. 4