EdgeGaussians - 3D Edge Mapping via Gaussian Splatting Supplementary Material

Kunal Chelani[1] Assia Benbihi[2] Torsten Sattler[2] Fredrik Kahl[1]
[1]Chalmers University of Technology
[2]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
chelani@chalmers.se

Appendix A provides implementation details about the training of the edge-specialized Gaussian Splatting. Appendix B shows qualitative results over the scenes from the Replica [7] dataset used by the authors of EMAP [4] and three scenes from the Tanks and Temples dataset [3]. Appendix C discusses some limitations and failure cases of our method, pointing to relevant future work.

A. Implementation details

Initialization. *Gaussian Position:* For scenes from the DTU [1], Replica [7] and Tanks and Temples [3] datasets, we use the SfM [6] points as initialization. Note that random point initialization also produces reasonable, but slightly worse results. For *ABC-NEF* [9], we initialize our method with Gaussians centered at 10000 points randomly sampled in a unit cube. This is because the dataset comprises texture-less objects for which Structure-from-Motion (SfM) [6] generates extremely sparse or no point reconstruction at all. *Gaussian Scale*: We use a constant initial value of 0.004 for all datasets. However, a point-dependent value based on the complexity of the neighboring region may be more robust. *Gaussian Opacity*: We use a constant initial value of 0.08 for all datasets. *Gaussian Orientation*: Random unit quaternions are used as initial values for all Gaussians.

Training. We train the model for 500 epochs. For the first 30 epochs, we only train the position parameters so that the scale and the orientation of the Gaussian do not compensate for its incorrect position during rendering. Thus the training constrains the Gaussian's position, *i.e.*, its mean, to lie on 3D edges. We cull the Gaussians based on opacity and duplicate the ones with high positional gradients at regular intervals as in the original work [2]. The learning rates of different parameters are as follows. Position: starting with $1e^{-3}$, scaled with a factor of 0.75 every 10 epochs, 5 times. Scale: $2e^{-4}$ constant. Opacity: $3e^{-2}$ constant. Orientation: $1e^{-3}$ constant.

We use $k = 4$ as the number of nearest neighbors for computing $\mathcal{L}_{\text{orient}}$ defined in Eq.(4) of the main paper. The weights of the loss function in Eq.(6) of the main paper are $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$ for object level scenes from *ABC-NEF* [9] and DTU [1], while for larger scenes we use smaller values of $\lambda_1 = 0.01$ and $\lambda_2 = 0.01$. The geometric regularization assumes that the Gaussians are already positioned close to the edges, therefore we start applying this regularization at epoch 300. Note that the computation of nearest neighbors, required for the geometric regularization is computationally intensive and we observe that it is sufficient to only apply this regularization once in every 10 steps of the training process.

Clustering During clustering, the alignment threshold is $\theta = 0.8$ on *ABC-NEF* [9], which have clean straight lines, and $\theta = 0.6$ on DTU [1] to account for the higher curvature of the shapes. During the parametric edge fitting, we fit a curve whenever the curve residual error is $\delta = 0.5$ lower than the line residual error. For objects from the DTU dataset [1], due to the prior knowledge that the objects have more curves than lines, a larger $\delta = 1$ is used.

For any further clarifications, please refer to the code released at https://github.com/kunalchelani/EdgeGaussians.

B. Additional Qualitative Results

Fig. 1 to 3 show the results for the scenes *room_0*, *room_1*, *room_2* of the Replica [7] dataset. The results show that our method produces edges with a higher completeness than EMAP [4]. Also, EMAP [4] predicts clusters of duplicate edges close to the ground-truth edge, which is not a desirable result as it makes the reconstruction less sharp. Overall, our method produces clean single edges that are more complete. However, in some cases, EMAP [4] produces geometrically accurate lines, which our method captures as incomplete curves. Fig. 2 shows one such example. Although this can be partially addressed by adjusting the parameter δ involved in the model selection when fitting a line or a curve, this could be seen as a current limitation of our method.

C. Limitations and Failure Cases

Fine structures and geometric regularization. As briefly described in the main paper, the method is limited by the noise in the supervisory signal of the 2D edge maps. In many cases the fine structures in such edge maps [5, 8] are not discernible, leading to incorrectly positioned edge points. Geometric regularization applied to noisy edge points can lead the Gaussians' to form short local curves to satisfy the alignment with their nearest neighbors. Examples of such cases can be seen in Fig. 5.

Clustering and edge fitting. Further, the clustering algorithm exhibits limitations when applied to larger scenes with complex structures. Fig. 4 shows examples from the Tanks and Temples dataset [3] where the oriented edge points (red), *i.e.*, the 3D Gaussians, better cover the ground-truth 3D edges than the paramtric edges (black). One explanation is that the graph traversal based clustering removes several correct edge components close to the true scene structure while including several incorrect edges. Instead of relying only on local geometric heuristics, defining a prior on which parts of the scene are more likely to hold 3D edge could improve the method's robustness.

References

[1] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 1

[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), July 2023. 1

[3] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 1, 2, 6

[4] 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. 1, 3, 4, 5

[5] 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. 2, 7

[6] 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. 1

[7] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 3, 4, 5

[8] 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. 2, 7

[9] 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, June 2023. 1, 7

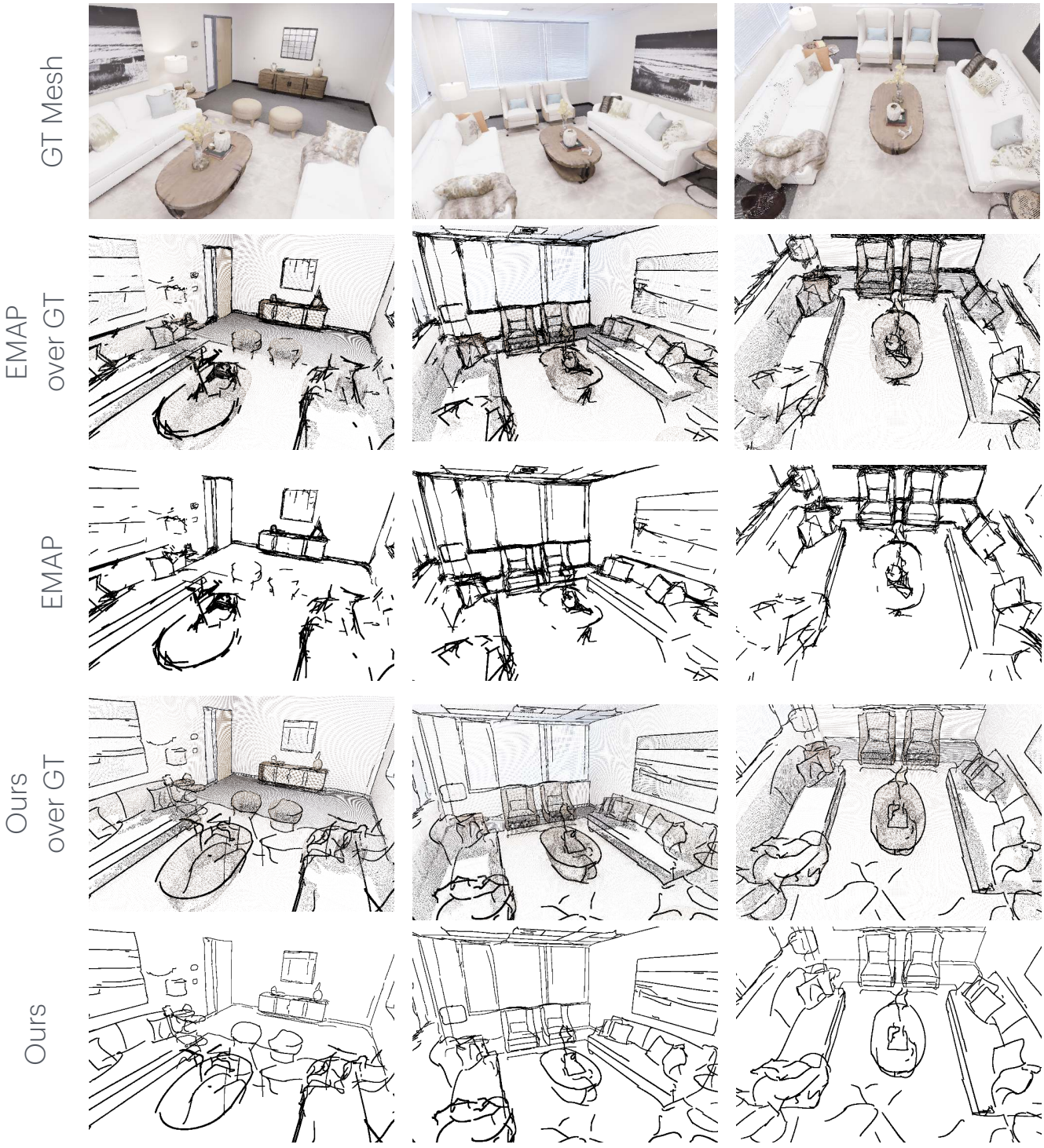

Figure 1. **Replica [7] *room_0*** : Qualitative result showing edges produced by our method and EMAP [4]. In general it can be observed that EMAP [4] has several duplicate / dense sets of edges close to ground-truth edges whereas our method produces clean single edges.

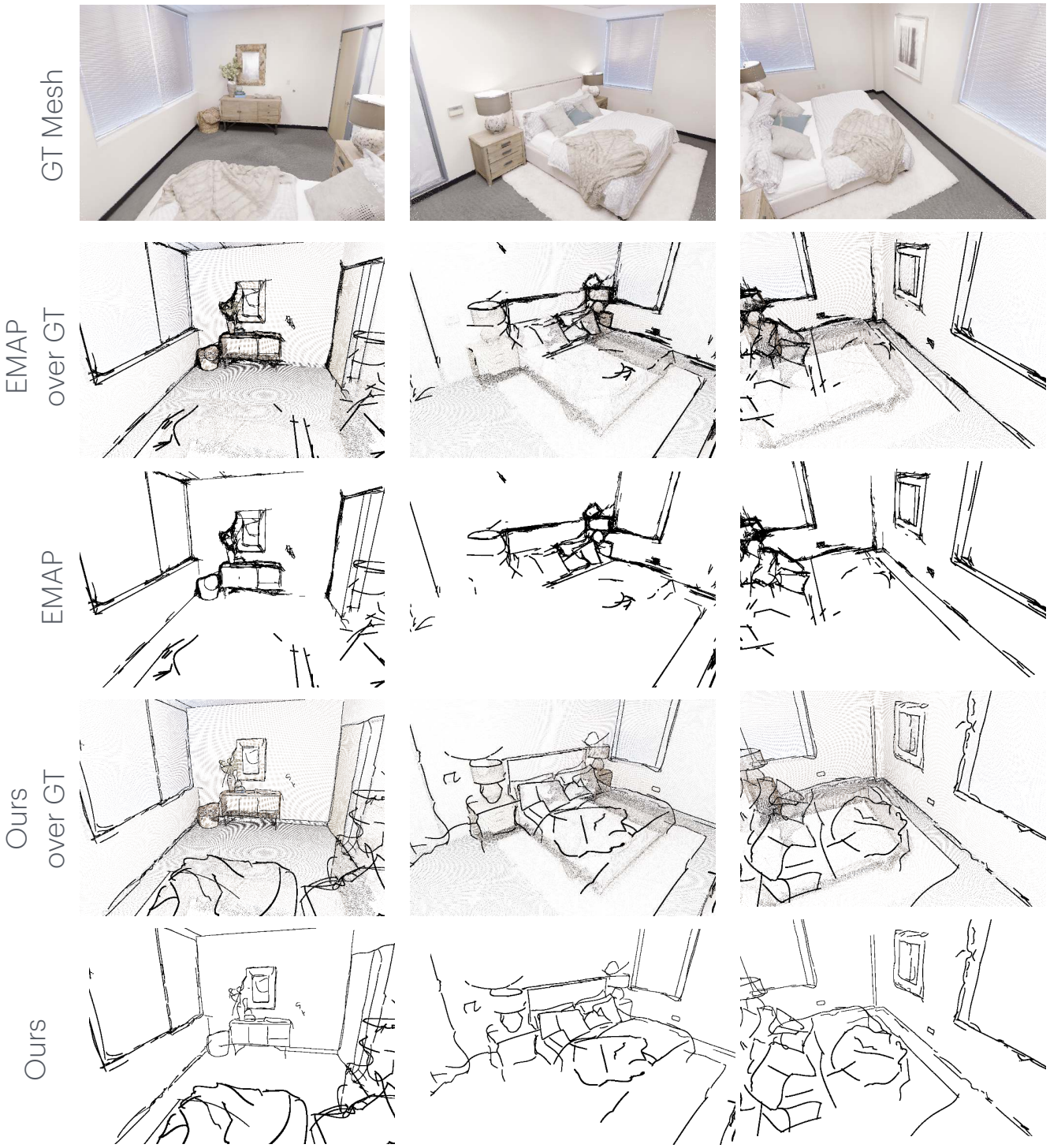

Figure 2. **Replica [7] *room_1*** : Qualitative result showing three different views of edges produced by our method and EMAP [4]. In general it can be observed that EMAP [4] has several duplicate / dense sets of edges close to ground-truth edges, while our method produces clean single edges. However, EMAP [4] produces more accurate lines for some geometric edges, for example, on the window pane.

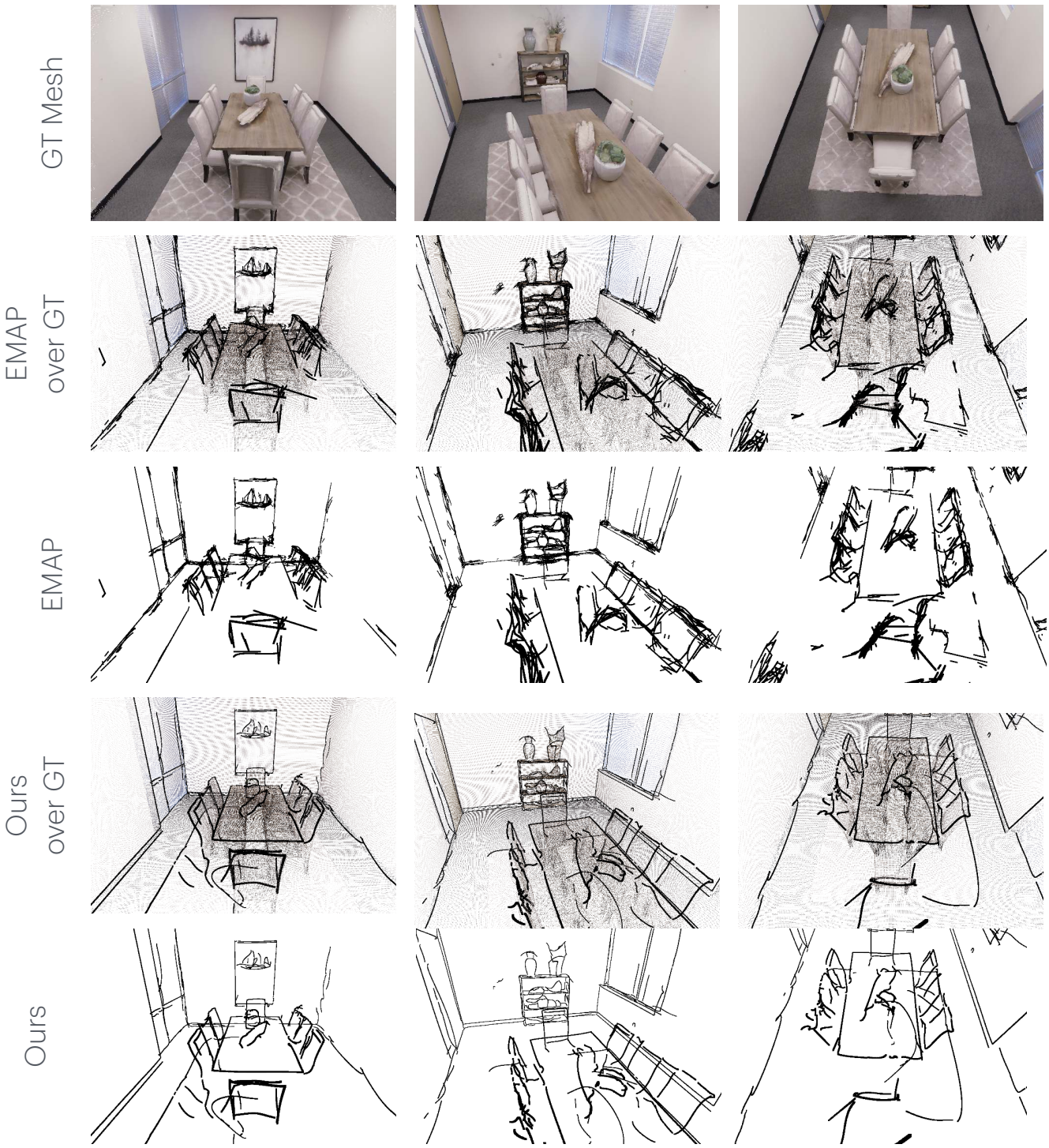

Figure 3. **Replica [7] *room_2*** : Qualitative result showing edges produced by our method and EMAP [4]. In general it can be observed that EMAP [4] has several duplicate / dense sets of edges close to ground-truth edges, while our method produces clean single edges.

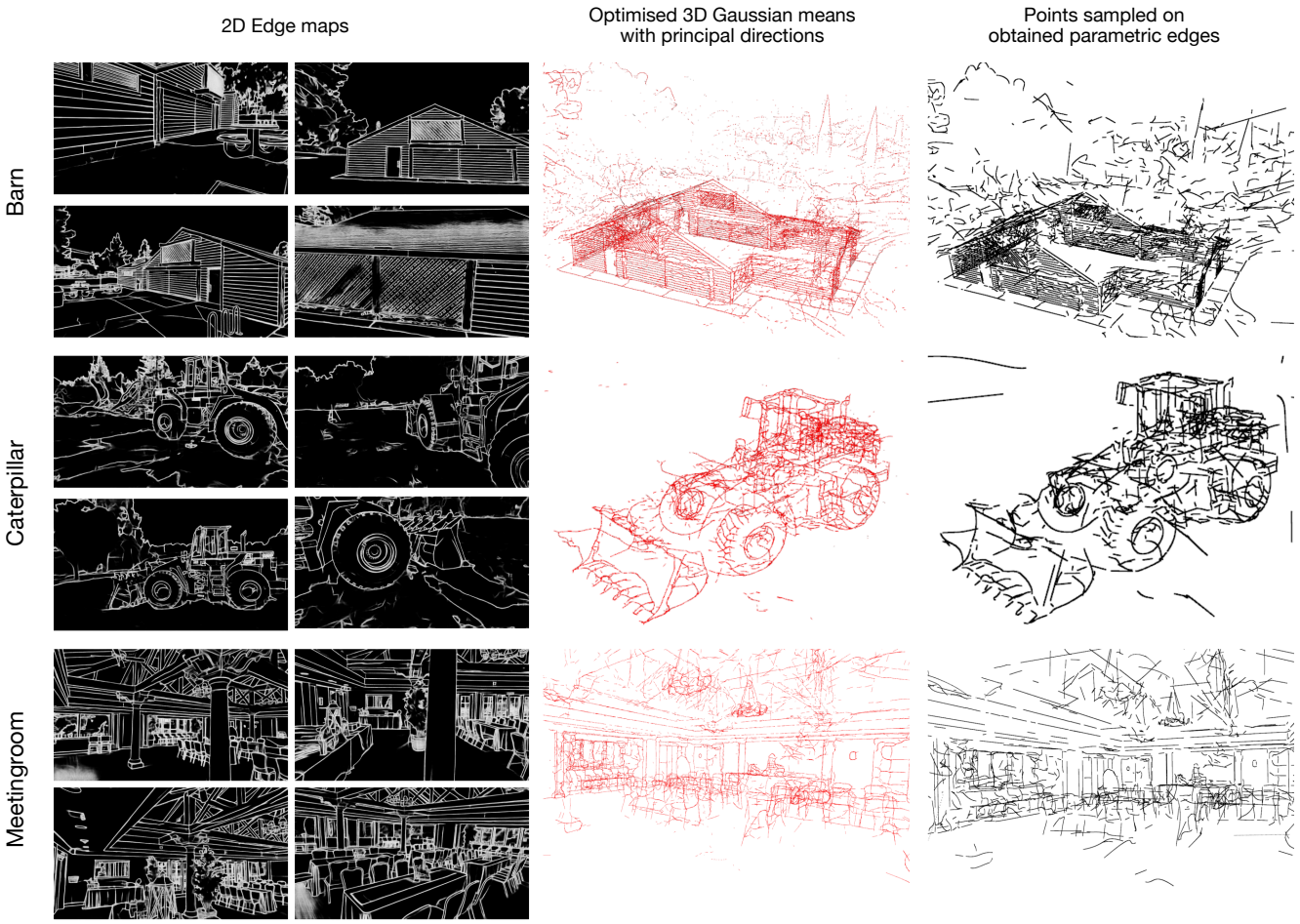

Figure 4. **Tanks and Temples** [3]: Qualitative result showing edges produced by our method on three scenes from the tanks and temples dataset. Supervisory signal (Left), edge points represented as a small line segment centered at the mean of the optimized 3D Gaussians and oriented towards their principal directions (Middle) and the points sampled on the parametric 3D edges estimated (Right). Note that the estimated Gaussians faithfully represent the scene but the clustering and edge fitting process have room for improvement as many correct edges are missed and spurious ones are created in this process.

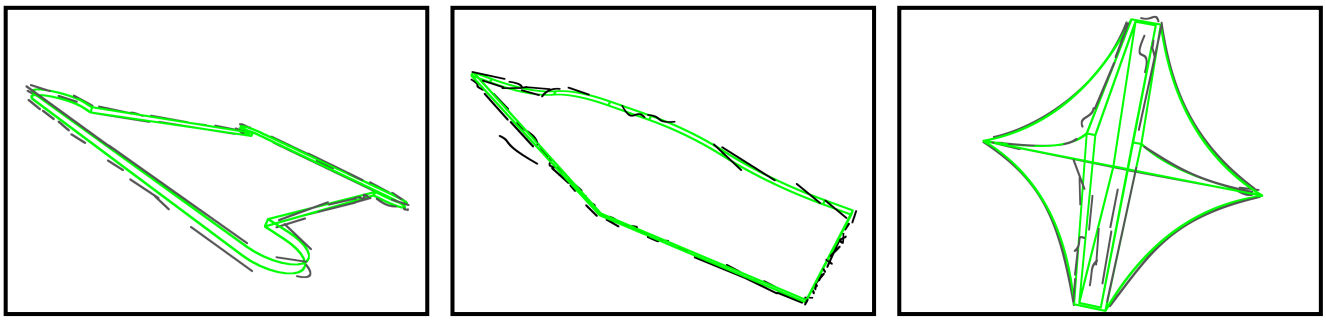

Figure 5. **Failure cases** : Scans *00009685*, *00002412* and *00003884* (left to right) on the *ABC-NEF* [9]. The edges predicted by our method are shown in black and the ground-truth ones in green. These examples are challenging because they show extremely thin structures: the projection on two distinct parallel and close 3D edges can get projected into a single edge in several views of the supervisory 2D edge maps [5, 8]. Another example where the proposed method is incomplete (right) is when the object has 3D edges inside the structure that are not detected by the 2D edge detectors. Then, there is no supervisory signal for those 3D edges.