TransformerFusion: Monocular RGB Scene Reconstruction using Transformers –Supplementary Material–

Aljaž Božič [1] **Pablo Palafox** [1] **Justus Thies** [1,2] **Angela Dai** [1] **Matthias Nießner** [1]

[1]Technical University of Munich
[2]Max Planck Institute for Intelligent Systems, Tübingen, Germany
`aljazbozic.github.io/transformerfusion`

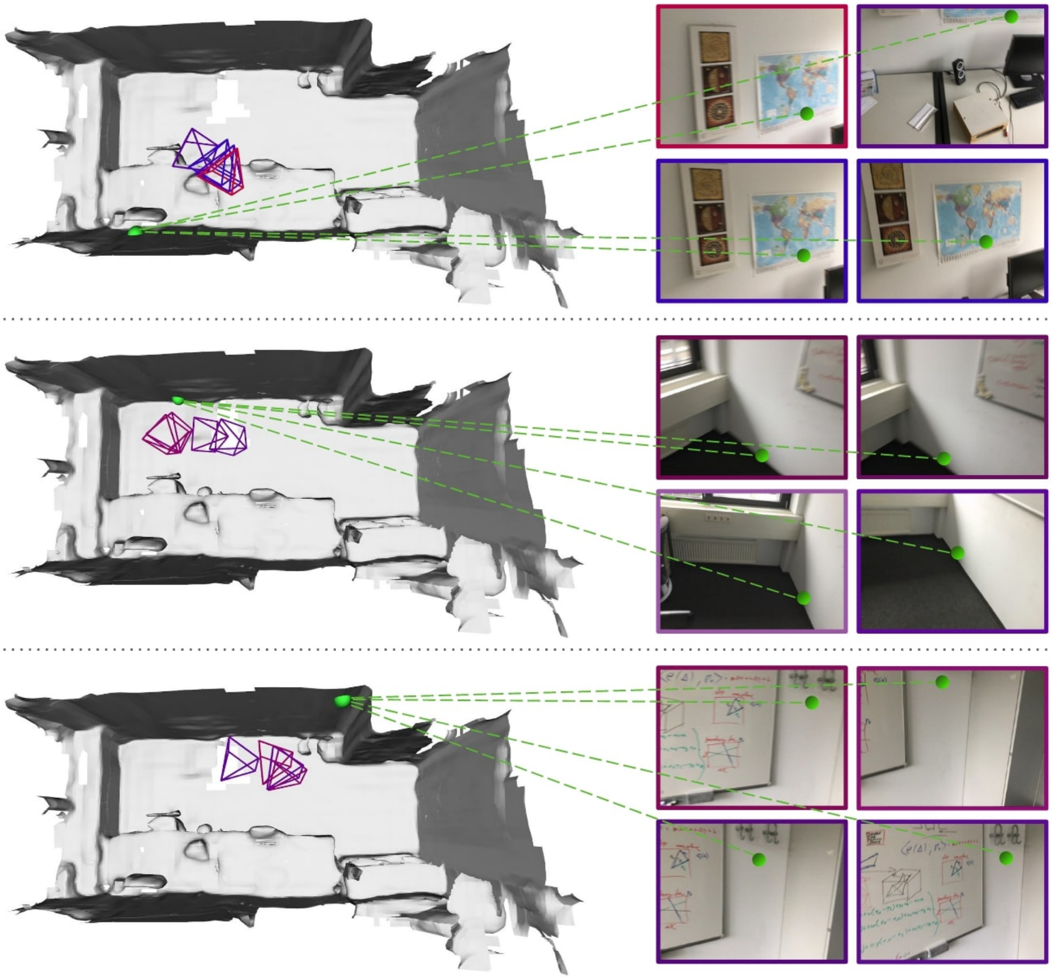

Figure 1: Visualization of selected camera views with self-supervised attention weights (lower weights are visualized as *blue* and higher as *red*) for specific 3D locations (highlighted as *green*) in the scene.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

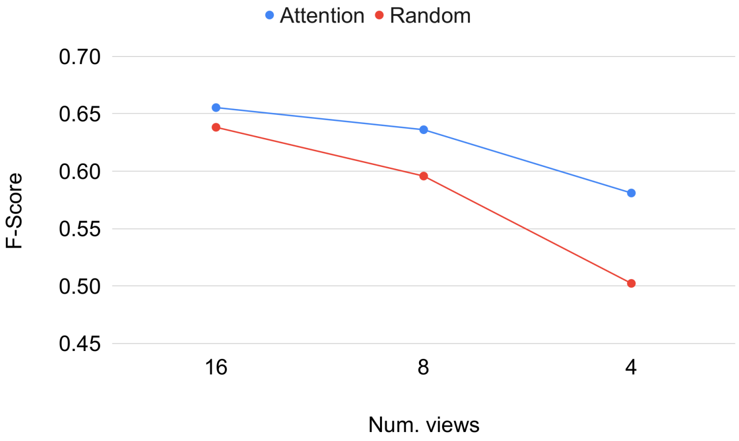

Figure 2: Comparison of our attention-based view selection scheme to a random selection of frames from the frame candidates based on the F-score.

A Additional Results

Visualization of Frame Selection. In Fig. 1, we show a 3D reconstruction of a scene from the test set of the ScanNet dataset [1]. To visualize the view selection approach presented in the main paper that is based on the attention weights of the used transformer networks, we render the camera views that are selected for a specific 3D location (green point) with corresponding attention weights (color temperature corresponding to the weight). On the right, we show the corresponding input images. There are different colors (and, thus, attention weights) for similar views. We observed that the attention head in the transformer architecture tends to sparsify the views, assigning a high weight to a single view, and low weights for the rest, as other similar views tend to provide only redundant information. Such behavior is well-suited for the task of reconstruction, where multiple different observations are more useful, especially ones observed under different viewpoints and camera translations. This is achieved by using multiple attention heads (in our architecture we use 8 heads), each specializing for a certain view type, and each picking only a single view representative. This leads to elimination of very similar views that are redundant, and at the same time encourages high weights for different views that are more useful – additionally, this enables the attention weights to be very effective for online view selection. It is also a notable difference to existing works [3, 5], where all views are treated the same (by averaging over view features).

Ablation on Architecture Modules. As analyzed in the main document, the different algorithmic parts of our methods play an important role. Fig. 3 shows qualitative results for the ablation study. Specifically, one can clearly see the impact of the spatial refinement as well as the temporal feature fusion via our transformer architecture. For qualitative results of the setting without using transformers for feature fusion we use predicted weights for weighted averaging of features via an MLP.

Ablation on View Selection. In Fig. 2, we show a comparison of our view selection scheme to the baseline that takes random views from the view candidates (views that contain a specific point). Specifically, we vary the number of views that can be selected. As can be seen, the reconstruction quality gap between the baseline and our method increases with less views which is to be expected since the random selection is more likely to miss important views.

Ablation on Coarse-to-fine Features. To verify the intuition that high-level pixel features (at coarse image resolution) are well suited for coarse voxels, and similarly low-level pixel features (at high image resolution) for fine voxels, we additionally performed an ablation where features were switched – coarse features were used for fine voxels and fine features were used for coarse voxels. As can be seen in Tab. 1, this modification performs considerably worse.

Generalization to a Different Sensor. We additionally evaluated our approach on TUM RGB-D sequences [4], featuring office-like environments. These sequences were captured with a Kinect

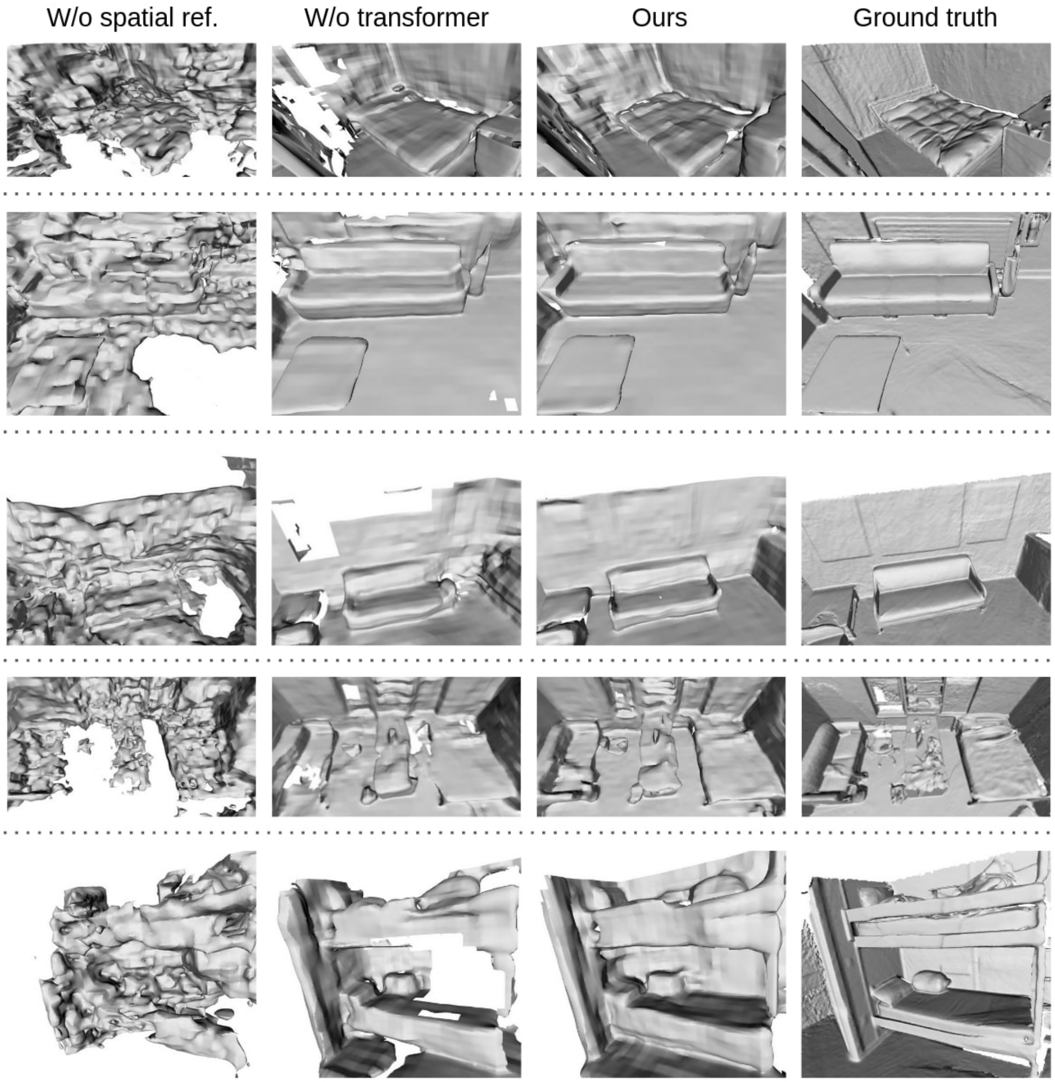

Figure 3: Qualitative comparison of ablations of our approach on test set of ScanNet dataset [1]; note that only RGB input is used by each method while the ground truth is reconstructed using the input depth.

sensor, while the ScanNet data that we used for training was recorded using a StructureIO sensor that uses an iPad RGB camera. Our approach achieves an F-score of 0.437, which is less than on the ScanNet test set but maintains accurate capture of scene structure. We believe the difference is mainly caused by a different RGB sensor used at test time. We note that in a practical scenario, one would fine-tune on the respective test device's camera and/or include recordings from multiple different sensors.

Table 1: Quantitative ablation on the use for pixel features: we compared our approach with modified method where coarse image features are used for fine voxels and fine image features are used for coarse voxels.

Method	**Acc ↓**	**Compl ↓**	**Chamfer ↓**	**Prec ↑**	**Recall ↑**	**F-score ↑**
Coarse/fine switch	6.92	9.55	8.24	0.620	0.477	0.536
Ours	$\mathbf{5.52}$	$\mathbf{8.27}$	$\mathbf{6.89}$	$\mathbf{0.728}$	$\mathbf{0.600}$	$\mathbf{0.655}$

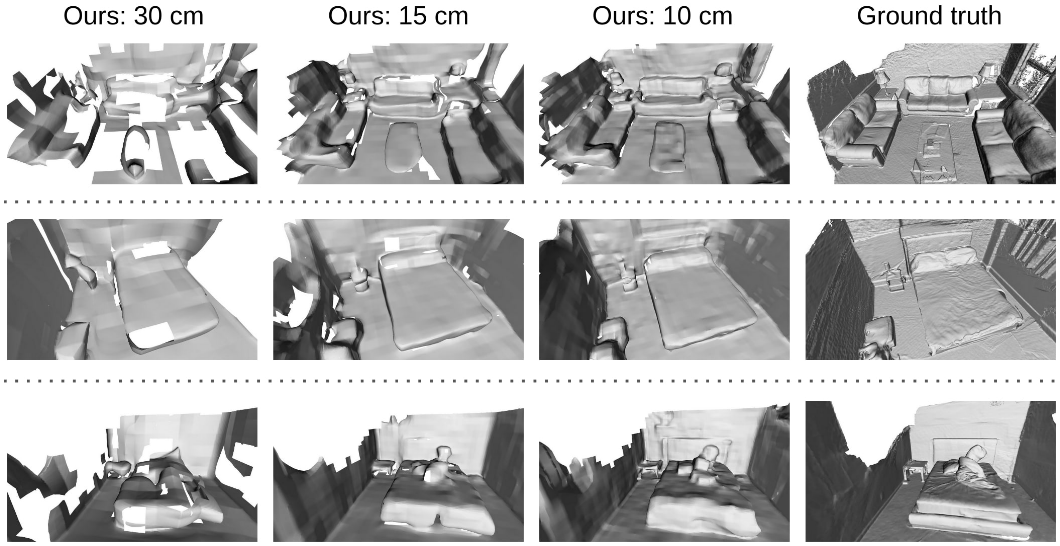

Figure 4: Qualitative comparison of ablations on feature voxel size: we replaced the original voxel size of 10 cm at the fine grid level with 30 cm and 15 cm.

Additional Qualitative Results. In Fig. 8 further examples are shown that demonstrate the reconstruction capability of our approach. We show a top down, as well as a view from inside the different reconstructed room scenes.

B Reproducibility

Table 2: Runtime analysis of our per-frame feature fusion.

Task	**Duration**
Image loading / feature extraction	21.50ms
Coarse feature fusion	11.81ms
Near-surface mask prediction	24.18ms
Fine feature fusion	73.03ms
Total	130.52ms

Table 3: Runtime analysis of the per-chunk mesh extraction.

Task	**Duration**
Coarse feature refinement	28.56ms
Fine feature refinement	140.79ms
MLP (occupancy prediction)	25.21ms
Marching cubes [2]	48.74ms
Total	243.29ms

Runtime Analysis. In this section we provide further details about the runtime of our approach. We benchmarked our approach using an Intel Xeon 6242R Processor and an Nvidia RTX 3090 GPU. For every new frame coarse-to-fine image features need to be extracted, and fused into global coarse and fine feature volumes. In Tab. 2, we report execution times of the different feature fusion steps. Coarse features are fused into the entire camera frustum, containing all coarse voxels that fall into valid depth range $[0.3\text{m}, 5\text{m}]$. Fine feature on the other hand are fused only in near-surface areas, as predicted by coarse filtering. The execution times are averaged over a representative video sequence of the ScanNet dataset. Note that surface reconstruction doesn't need to be extracted for every frame. It can either be done at the end, when all image features are already fused into the feature volume, or incrementally every couple of frames, on a per-chunk basis, if interactive feedback is desired. In Tab. 3, we report execution times for a chunk of size $1.5 \times 1.5 \times 1.5$ m. Both, coarse and fine features are spatially refined using a 3D CNN and surface occupancy is computed using the occupancy MLP at a voxel resolution of 2 cm, but only for near-surface voxels, as predicted by coarse and fine near-surface masks. Finally, the mesh is extracted using Marching cubes [2]. Note that our implementation uses high-level PyTorch routines, as well as CPU code (e.g., for Marching Cubes) and, thus, the implementation is not optimized for runtime. A more optimized implementation can be achieved via customized CUDA code. Another interesting avenue towards higher frame rates is the use of sparse 3D convolutions instead of dense 3D convolutions. Feature fusion timings are reported

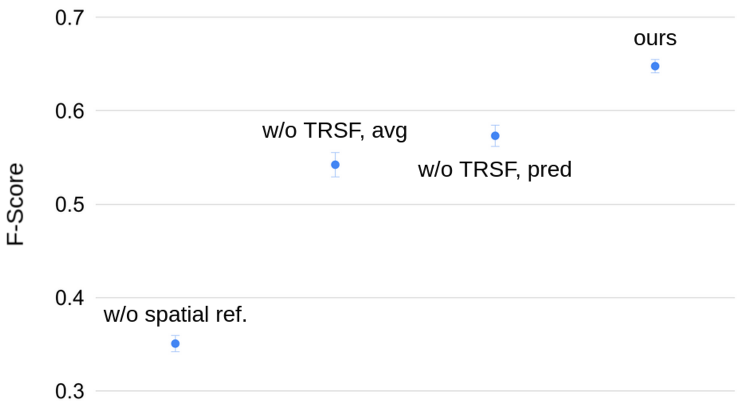

Figure 5: The F-score mean and standard deviation of multiple experiments of our approach and ablations.

for our default reconstruction setting, when we store $K = 16$ views for every feature grid voxel. The feature fusion execution can be further accelerated by using less views. The frames per second (FPS) increase from 7.66 FPS for 16 views to 10.17 FPS for 8 and to 12.28 FPS for 4 views.

Reproducibility of Experiments. To ensure the reproducibility of our experiments, we ran our approach and ablations 3 times. The resulting F-score mean and standard deviation for different experiments is shown is Fig. 5, with standard deviations visualized as error bars.

Network Architectures. In Fig. 6, we depict the architectures of the neural networks used in our approach. For both, the coarse and fine layer, we use independent feature fusion and feature refinement networks. The building blocks used in these networks are detailed in Fig. 7.

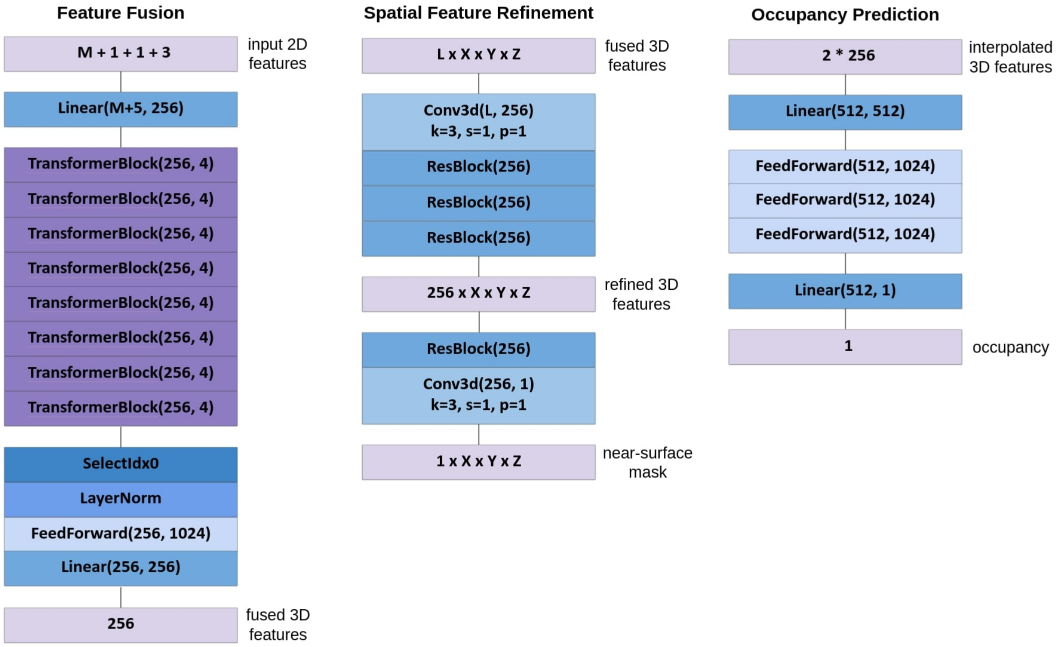

Figure 6: Overview of the used neural networks. Note that we are using a feature fusion and feature refinement network per level (coarse and fine).

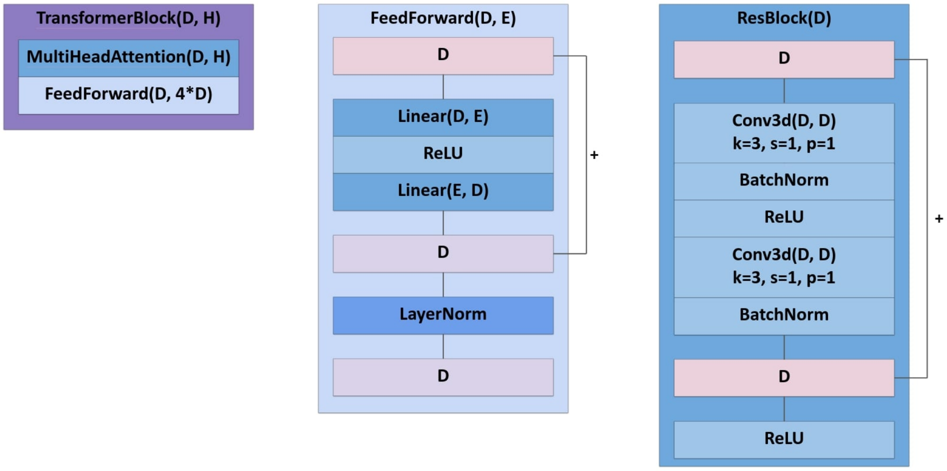

Figure 7: Low-level network details of the building blocks of our pipeline (see Fig. 6).

C Data

To train and evaluate our method, we use the ScanNet dataset [1], which is available under a non-commercial academic license[1]. ScanNet collects data of static indoor environments, and the ScanNet authors report that consent was obtained from the people whose private spaces were scanned. The ScanNet scenes and locations have been anonymized.

---

[1] `http://kaldir.vc.in.tum.de/scannet/ScanNet_TOS.pdf`

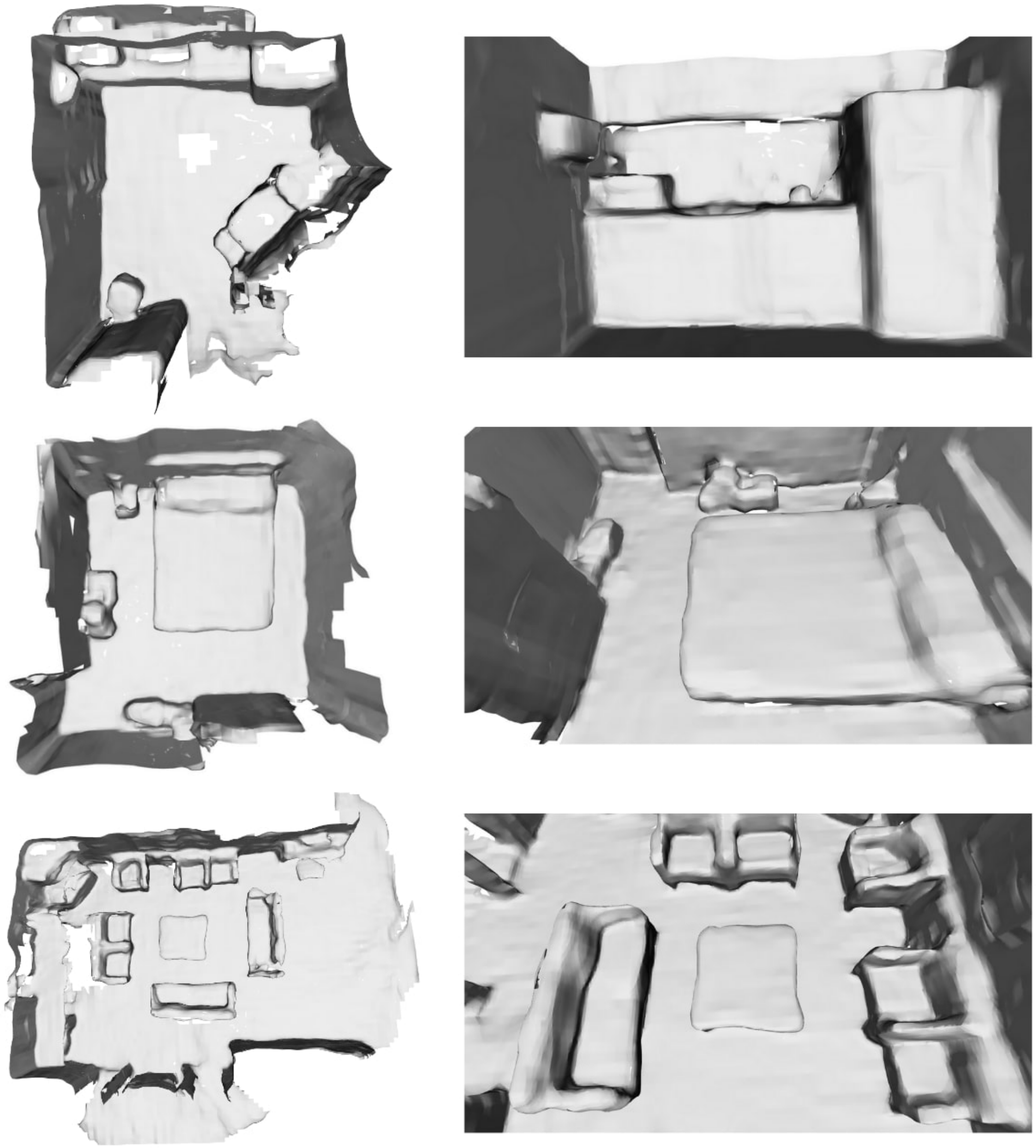

Figure 8: Qualitative results of representative scenes from the test-set of the ScanNet dataset [1].

References

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[2] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[3] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, 2020. URL `https://arxiv.org/abs/2003.10432`.

[4] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[5] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. *CVPR*, 2021.