FineRecon: Depth-aware Feed-forward Network for Detailed 3D Reconstruction Supplementary Material

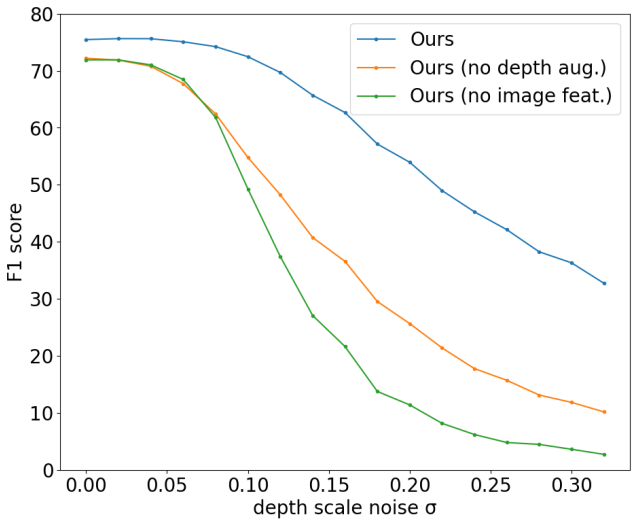

Figure 1. Our model is much more robust to depth error when trained with depth scale augmentation. In addition, as depth noise grows, image features are increasingly important to reconstruction quality.

1. Depth source and noise sensitivity

Fig. 1 explores the effect of test-time depth noise on reconstruction accuracy for three different trained models. We apply a random scaling factor to each predicted depth image, sampled from the normal distribution with standard deviation σ, observing a decay in F-score as σ increases. Interestingly, the falloff is earlier and steeper when our model is trained either without depth augmentation or without image features. This confirms our intuition that the depth scale augmentation increases robustness to depth error, and that the image features are important for recovering from depth error and disagreement.

Table 1 shows results for our model using different choices of the depth estimator M. The model is re-trained for each row. We observe that the choice of depth estimator has a considerable effect on performance. In the future, co-design of the depth estimator with the depth-guided reconstruction system may yield further improvements.

	3D metrics		Depth metrics	
Depth source	Cham ↓	F1 ↑	L1 ↓	$\delta_{1.05}$↑
Ground truth	3.00	91.8	4.15	94.6
SimpleRecon	5.18	75.5	6.91	86.6
DeepVideoMVS	5.86	72.0	8.67	82.4

Table 1. We train and test our model on ScanNet with three different depth estimators: Ground truth (structured-light depth sensor [2]), SimpleRecon [5], and DeepVideoMVS (pair network) [3].

2. Additional ablations

Table 2 shows additional experiments involving variations of the depth guidance strategy. In row (b) we train with no depth scale augmentation, showing significantly worse reconstruction metrics, and confirming that this augmentation is a critical component of our depth guidance contribution. In row (c) we use a small MLP θ_w to predict a scaling weight w for back-projecting image features:

$$w = \theta_w(\max(\min(\frac{z_v - \hat{d}_v}{12\text{cm}}, 1), -1)) \quad (1)$$

where z_v is the camera-to-voxel depth of the current voxel, and $\hat{d}_v$ is the predicted depth along the current camera ray. Surprisingly, this strategy performs worse than using a weight of $w = 1$ everywhere as in row (a). In row (d) we project each image feature only into the voxel at its predicted depth. This greatly impairs the reconstruction ability, and we infer that spreading the image features throughout space helps the 3D CNN to recognize and unify corresponding surfaces that are predicted at inconsistent depths.

3. Additional qualitative results

In Fig. 2 we replicate Fig. 5 from the main text with an alternate view, to further demonstrate the impact of varying the output resolution, highlighting the completeness, smoothness, and detail of our results.

In Fig. 3 we show the reconstruction quality on the 7-scenes dataset [4], using our ScanNet-trained model with no fine-tuning. Generalizing to this new data is not trivial, because the characteristics of 7-Scenes differ from ScanNet in terms of image noise, typical camera trajectory, rolling

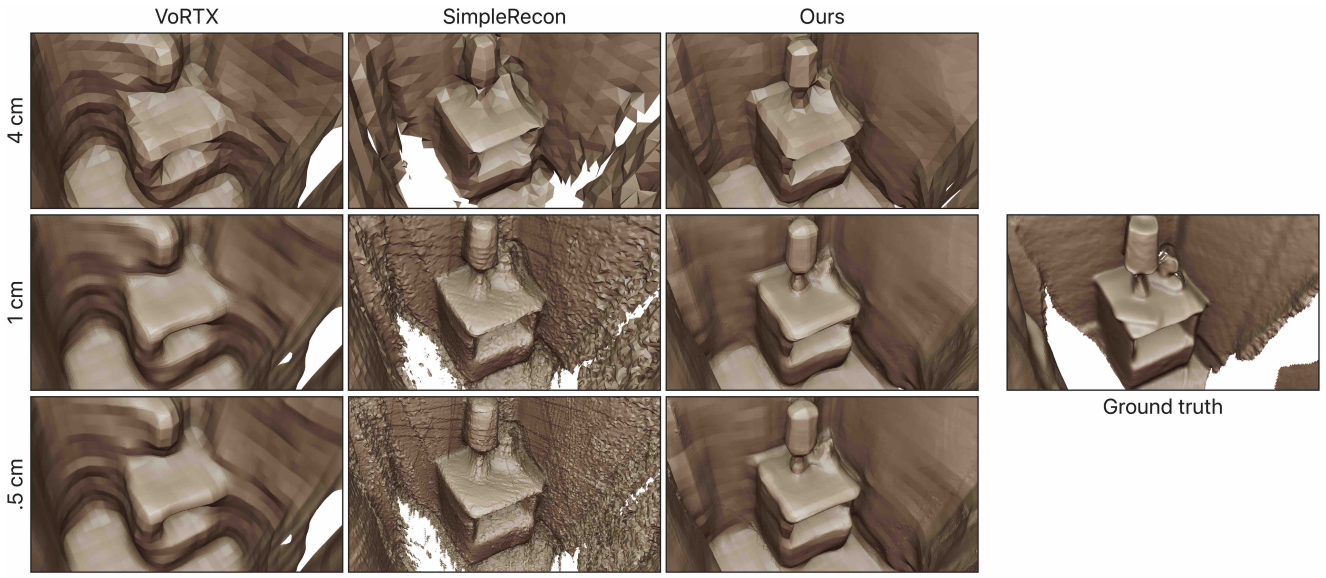

Figure 2. This additional close-up view shows the effect of different output resolutions for three models: VoRTX [6], SimpleRecon [5], and ours. Increasing the output resolution causes VoRTX to become overly-smooth, and it causes strong artifacts for SimpleRecon. In contrast, our method is able to smooth out curved surfaces while retaining sharp corners where appropriate, without adding high-frequency noise. Our best results occur at 1cm resolution, beyond which we see diminishing returns.

	3D metrics		Depth metrics	
	Cham ↓	F1 ↑	L1 ↓	$\delta_{1.05}$↑
(a) Ours	**5.19**	**75.4**	**7.08**	**86.4**
(b) w/o depth aug.	5.49	72.2	7.41	84.4
(c) w/ MLP weight	5.26	74.2	6.96	86.1
(d) w/ direct placement	5.62	71.2	8.06	83.6

Table 2. Our model performs significantly better when trained with depth augmentation (a) vs without (b). In (c) we use a small MLP to weight the back-projected image features based on distance to the predicted depth – most metrics become slightly worse. Row (d) shows the effect of back-projecting each image feature only into the voxel at its predicted depth estimate, reducing reconstruction quality.

shutter artifacts, and focal length. Nonetheless, the visual reconstruction quality is comparable to our ScanNet results, indicating reasonable robustness to these factors.

4. 3D CNN architecture details

Figure 4 shows the details of our 3D CNN architecture. Our main motivating design principle is simplicity, since our contributions are independent of the particular 3D CNN architecture.

5. Metrics definitions

The definitions for 3D reconstruction metrics are shown in Table 3, and the definitions for 2D depth metrics are

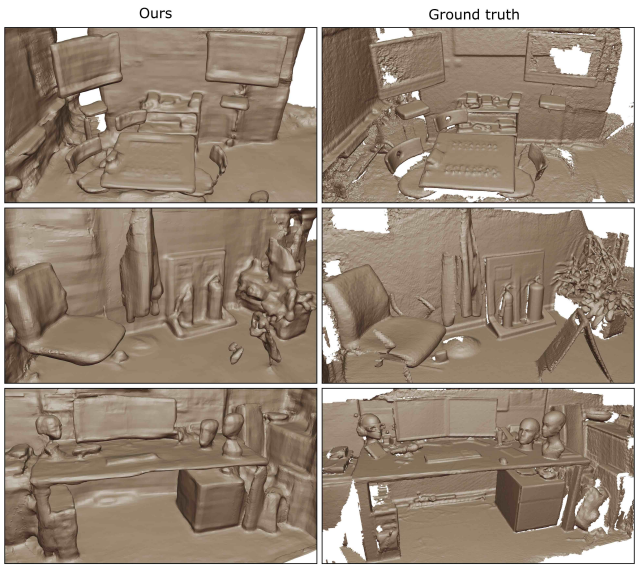

Figure 3. Qualitative results on the 7-scenes dataset [4] illustrate generalization to new data with no fine-tuning.

shown in Table 4. We compute 3D metrics using the protocol and code from TransformerFusion [1].

6. CNN feature visualizations

Figure 5 compares four selected feature maps from each 2D CNN feature extractor. The top row features from Ω^c are

Accuracy	$\frac{1}{\lvert P\rvert}\sum_{p\in P}\min_{p^*\in P^*}\lvert\lvert p-p^*\rvert\rvert_2$
Compl.	$\frac{1}{\lvert P^*\rvert}\sum_{p^*\in P^*}\min_{p\in P}\lvert\lvert p-p^*\rvert\rvert_2$
Chamf. dist.	$\frac{acc+comp}{2}$
Precision	$\frac{1}{\lvert P\rvert}\sum_{p\in P}\mathbb{1}(\min_{p^*\in P^*}\lvert\lvert p-p^*\rvert\rvert_2<5\text{cm})$
Recall	$\frac{1}{\lvert P^*\rvert}\sum_{p^*\in P^*}\mathbb{1}(\min_{p\in P}\lvert\lvert p-p^*\rvert\rvert_2<5\text{cm})$
F-score	$\frac{2}{prec^{-1}+rec^{-1}}$

Table 3. **3D reconstruction metrics.** P is a point cloud sampled on the predicted mesh. P^* is a point cloud consisting of the ground-truth mesh vertices. $\mathbb{1}$ is the indicator function.

L1	$\frac{1}{\lvert P_{DD^*}\rvert}\sum_{p\in\lvert P_{DD^*}\rvert}\lvert D(p)-D^*(p)\rvert$
AbsRel	$\frac{1}{\lvert P_{DD^*}\rvert}\sum_{p\in\lvert P_{DD^*}\rvert}\frac{\lvert D(p)-D^*(p)\rvert}{D^*(p)}$
SqRel	$\frac{1}{\lvert P_{DD^*}\rvert}\sum_{p\in\lvert P_{DD^*}\rvert}\frac{\lvert D(p)-D^*(p)\rvert^2}{D^*(p)}$
$\delta_{1.05}$	$\frac{1}{\lvert P_{DD^*}\rvert}\sum_{p\in\lvert P_{DD^*}\rvert}\mathbb{1}(\max(\frac{D(p)}{D^*(p)},\frac{D^*(p)}{D(p)})<1.05)$
$\delta_{1.25}$	$\frac{1}{\lvert P_{DD^*}\rvert}\sum_{p\in\lvert P_{DD^*}\rvert}\mathbb{1}(\max(\frac{D(p)}{D^*(p)},\frac{D^*(p)}{D(p)})<1.25)$
Compl.	$\frac{\lvert P_D\rvert}{\lvert P\rvert}$

Table 4. **2D depth metrics.** P is the set of all pixels. P_D is the subset of pixels with a valid predicted depth, and P_{D^*} is the subset of pixels with a valid ground-truth depth. For convenience we define the intersection $P_{DD^*} = \lvert P_D \cap P_{D^*}\rvert$. $D(p)$ is the predicted depth at pixel position p, and $D^*(p)$ is the ground-truth depth at pixel position p.

used as inputs to the 3D CNN to compute the coarse (4cm) scene structure, and the bottom row features from Ω^f are used in the point back-projection branch to recover fine local details. The Ω^c outputs are characterized by a speckle pattern, whereas the Ω^f outputs are smoother and preserve more of the input image texture. Further experiments investigating these differences and their origins may yield useful insights for the design of future systems.

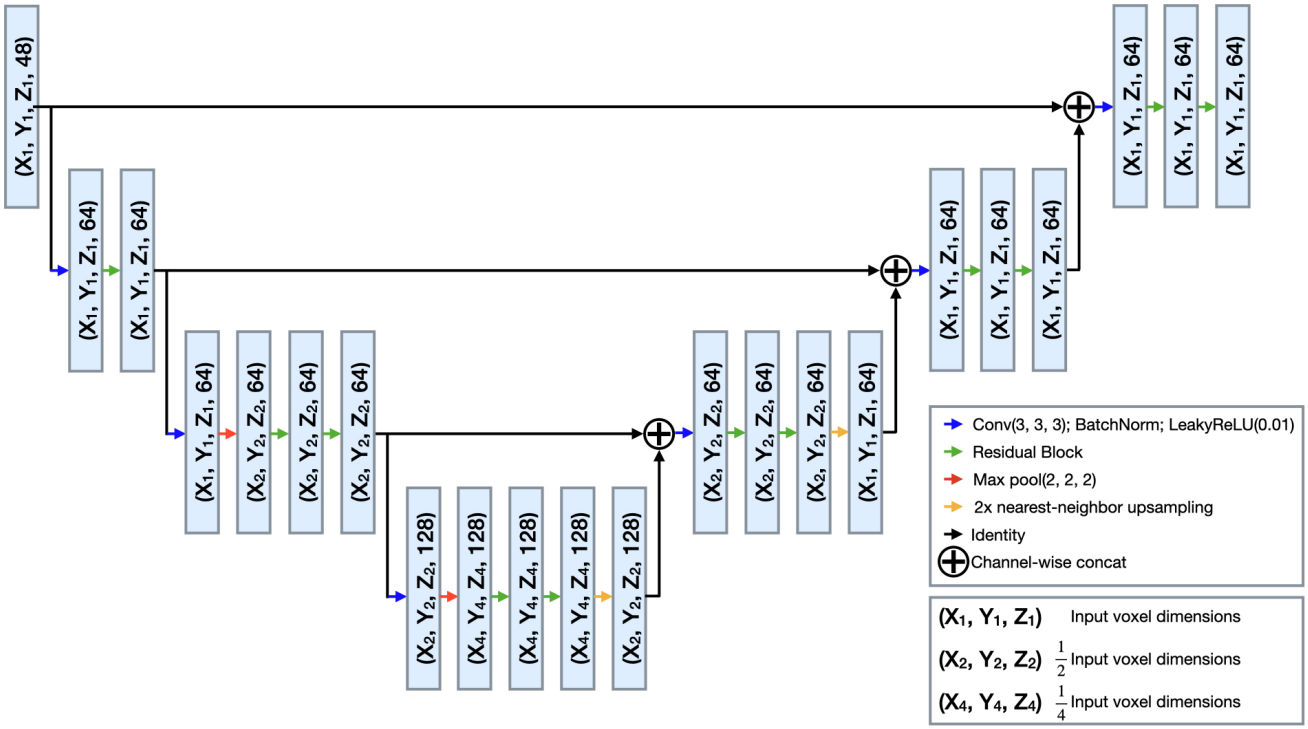

Figure 4. The architecture of our 3D CNN Ψ. Light-blue blocks represent 3D feature maps where the first three dimensions are spatial and the last is the feature dimension. Residual blocks have the form $R(x) = \sigma(x + (B_2 \circ C_2 \circ \sigma \circ B_1 \circ C_1)(x))$ where $\circ$ represents function composition, C_1 and C_2 are $3 \times 3 \times 3$ convolutional layers, B_1 and B_2 are batch normalization layers, and σ is a leaky ReLU [7]: $\sigma(x) = \max(x, 0.01x)$.

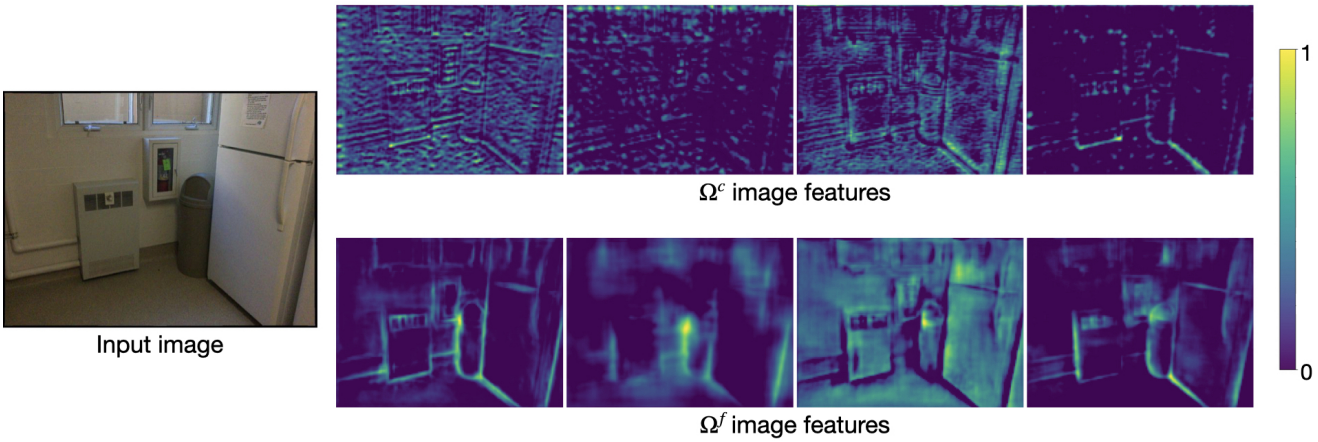

Figure 5. 2D image features extracted by our model, normalized to $[0, 1]$ for visualization. The features extracted by the two CNNs Ω^c and Ω^f are visually distinct, and further examination of these differences may lead to useful insights in the future.

References

[1] Aljaž Božič, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 2

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 1

[3] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15324–15333, 2021. 1

[4] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179. IEEE, 2013. 1, 2

[5] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 2

[6] Noah Stier, Alexander Rich, Pradeep Sen, and Tobias Höllerer. Vortx: Volumetric 3d reconstruction with transformers for voxelwise view selection and fusion. In *2021 International Conference on 3D Vision (3DV)*, pages 320–330. IEEE, 2021. 2

[7] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 4