AniGrad: Anisotropic Gradient-Adaptive Sampling for 3D Reconstruction From Monocular Video

Supplementary Material

11. Additional implementation details

Code structure. Our system is implemented as a fork of FineRecon [34], in order to re-use the general framework and backbone architectures. Our most important additions are to the TSDF output and surface extraction components for adaptive resolution (Sections 4.2, 5.1, and 5.2), with other significant changes to data pre-processing and loss functions (Sections 6.1 and 6.2).

Fast execution. Our final TSDF prediction architecture (Eqs. 1 and 3) and our gradient bound computation (Eq. 8) are implemented as vectorized PyTorch operations running on GPU, and thus have minimal performance impact. Our surface extraction is implemented from scratch using Cython [1], allowing us to compile it to a fast Python extension running on CPU.

Scene size. We determine the scene bounds by rendering the mesh to depth images from the perspective of each input camera, then back-projecting to a point cloud, and finally taking a bounding box around the resulting points. Effectively this sets the bounds according to the visible portion of the ground-truth mesh. This is important, particularly for ScanNet++ where certain large portions of the ground truth mesh are never observed by the input camera. The voxel grid typically spans 100-300 voxels horizontally, and 50-100 voxels vertically, depending on the camera path.

Image size. For ScanNet++ we use the iPhone images, not the DSLR images. We resize all input images to 320x240 pixels before passing to our 2D feature extractor. While higher image resolution might provide useful extra information, we found that in practice the additional memory consumption limits other parameters such as batch size, leading to worse overall performance.

Training. We use the Adam optimizer, with a learning rate of 1e-3 for 55,000 steps, followed by a learning rate of 1e-4 for 5,000 steps. Our batch size is 4, but we accumulate gradients over two mini-batches at a time for an effective batch size of 8. We train on volumetric scene crops of size $64 \times 64 \times 64$ voxels.

Basis functions. We found the network training stability to be sensitive to the activation function used in the basis function MLP (Eq. 2). We achieved our best results using the SoftPlus activation with parameter $\beta = 10$.

12. Architecture choices

In Table 3, sub-sampling refers to the time to make a TSDF prediction at all sample points, given the voxel features and the sample point locations. For our model, sub-sampling consists of executing the basis functions and performing the weighted sum across neighboring voxels, as shown in Eq. 3. For FineRecon it consists of sampling and fusing high-resolution image features and depth estimates (these are called point back-projection and depth guidance), and then running an MLP on the result. We opted to eliminate FineRecon's depth guidance and point back-projection branches for efficiency, as we found that they make the TSDF sampling operation much more expensive. This change initially caused a large drop in mesh quality (see Table 2, row e), but we were able to recover it using our improved mesh-based supervision (Section 6.2).

13. Gradient bound derivation

Here we provide the full derivation of the gradient bound ϕ used in the paper.

13.1. Definition of the bound

$\phi(s)$ represents an upper bound on the maximum absolute predicted TSDF gradient within a central cube (CC) called s, so by definition we start with,

$$\phi(s) \geq \max_{p \in s} |\nabla \hat{d}(p)|, \;\; \phi \in \mathbb{R}^3, \tag{19}$$

where $\hat{d}$ is the predicted TSDF and $p = [x, y, z$ is a query point. We continue with the derivation in the x dimension only, noting that it is very similar for y and z. Denoting the first element of ϕ as ϕ_x, we can restate the definition from Eq. 19 in the x dimension as,

$$\phi_x(s) \geq \max_{p \in s} |\frac{\partial}{\partial x} \hat{d}(p)|, \;\; \phi_x(s) \in \mathbb{R}. \tag{20}$$

13.2. Definition of the predicted TSDF

We re-iterate the definition of the predicted TSDF from Section 4.2,

$$\hat{d}(p) = \sum_{v \in N(p)} \hat{d}_v(p) \text{vol}(p - c_{7-v}). \tag{21}$$

This is a volume-weighted smoothing of the neighboring per-voxel predictions. The weights are equivalent to those used in trilinear interpolation,

$$\text{vol}([x, y, z]) = |xyz|. \tag{22}$$

Each voxel makes its own independent TSDF prediction as,

$$\hat{d}_v(p) = \sum_i \mathcal{F}(v)^{(i)} \beta_i (p - c_v), \tag{23}$$

where $\mathcal{F}(v)$ is the voxel feature, $\mathcal{F}(v)^{(i)}$ is its i^{th} channel, β_i is the i^{th} basis function, and c_v is the center of the voxel.

13.3. Bounding $\hat{d}$ in terms of d_v

We start by expanding Eq. 21 as follows,

$$\begin{aligned}\hat{d}(p) = (& \\ & \hat{d}_0(p)\text{vol}(p - c_7) \\ & + \hat{d}_1(p)\text{vol}(p - c_6) \\ & \dots \\ & + \hat{d}_7(p)\text{vol}(p - c_0) \\ &),\end{aligned} \tag{24}$$

where the relative positions of voxel centers c_0 through c_7 are illustrated in Fig. 3. Now, since we aim to bound $|\frac{\partial}{\partial x}\hat{d}(p)|$, we differentiate Eq. 24 using the chain rule and take the absolute value,

$$\begin{aligned}|\frac{\partial}{\partial x}\hat{d}(p)| = | & \\ & \text{vol}(p - c_7)\frac{\partial}{\partial x}\hat{d}_0(p) + \hat{d}_0(p)\frac{\partial}{\partial x}\text{vol}(p - c_7) \\ & + \text{vol}(p - c_6)\frac{\partial}{\partial x}\hat{d}_1(p) + \hat{d}_1(p)\frac{\partial}{\partial x}\text{vol}(p - c_6) \\ & \dots \\ & + \text{vol}(p - c_0)\frac{\partial}{\partial x}\hat{d}_7(p) + \hat{d}_7(p)\frac{\partial}{\partial x}\text{vol}(p - c_0) \\ & |.\end{aligned} \tag{25}$$

In order to bound Eq. 25, we first re-order its terms to group the $\frac{\partial}{\partial x}\hat{d}$ terms together and the $\hat{d}$ terms together,

$$\begin{aligned}|\frac{\partial}{\partial x}\hat{d}(p)| = | & \\ & \text{vol}(p - c_7)\frac{\partial}{\partial x}\hat{d}_0(p) + \dots + \text{vol}(p - c_0)\frac{\partial}{\partial x}\hat{d}_7(p) \\ & + \hat{d}_0(p) \cdot \frac{\partial}{\partial x}\text{vol}(p - c_7) + \dots + \hat{d}_7(p) \cdot \frac{\partial}{\partial x}\text{vol}(p - c_0) \\ & |.\end{aligned} \tag{26}$$

By the triangle inequality,

$$\begin{aligned}|\frac{\partial}{\partial x}\hat{d}(p)| \leq & \\ & |\text{vol}(p - c_7)\frac{\partial}{\partial x}\hat{d}_0(p) + \dots + \text{vol}(p - c_0)\frac{\partial}{\partial x}\hat{d}_7(p)| \\ & + |\hat{d}_0(p) \cdot \frac{\partial}{\partial x}\text{vol}(p - c_7) + \dots + \hat{d}_7(p) \cdot \frac{\partial}{\partial x}\text{vol}(p - c_0)|.\end{aligned} \tag{27}$$

Now we bound the two terms separately. For the first, observe that the sum of the volume weights is always equal to the total CC volume,

$$\sum_v \text{vol}(p - c_v) = V^3, \tag{28}$$

where V is the voxel size. Thus we can bound the first term from Eq. 27 by imagining that the maximum of any $\frac{\partial}{\partial x}\hat{d}_v$ occurs with maximum volume weight V^3,

$$\begin{aligned}&|\text{vol}(p - c_7)\frac{\partial}{\partial x}\hat{d}_0(p - c_0) + \dots + \text{vol}(p - c_0)\frac{\partial}{\partial x}\hat{d}_7(p - c_7)| \leq \\ &V^3 \max\{\max_p |\frac{\partial}{\partial x}\hat{d}_0(p - c_0)|, \dots, \max_p |\frac{\partial}{\partial x}\hat{d}_7(p - c_7)|\}.\end{aligned} \tag{29}$$

The second term from Eq. 27 can be bounded using the fact that as the volume grows to the left, it must shrink to the right, $\frac{\partial}{\partial x}vol(p - c_v) = -\frac{\partial}{\partial x}vol(p - c_{v+2})$ (see Fig. **??**). Thus,

$$\begin{aligned}&|\hat{d}_0(p)\frac{\partial}{\partial x}\text{vol}(p - c_7) + \dots + \hat{d}_7(p)\frac{\partial}{\partial x}\text{vol}(p - c_0)| = \\ &\quad |\frac{\partial}{\partial x}\text{vol}(p - c_7)(\hat{d}_2(p) - \hat{d}_0(p)) \\ &\quad + \frac{\partial}{\partial x}\text{vol}(p - c_6)(\hat{d}_3(p) - \hat{d}_1(p)) \\ &\quad + \frac{\partial}{\partial x}\text{vol}(p - c_3)(\hat{d}_6(p) - \hat{d}_4(p)) \\ &\quad + \frac{\partial}{\partial x}\text{vol}(p - c_2)(\hat{d}_7(p) - \hat{d}_5(p))|\end{aligned} \tag{30}$$

We can then observe that $\sum_{v\in[2,3,6,7]} \frac{\partial}{\partial x}\text{vol}(p - c_v) = -V^2$, so Eq. 30 is bounded as,

$$\begin{aligned}&|\hat{d}_0(p - c_0)\frac{\partial}{\partial x}\text{vol}(p - c_7) + \dots + \hat{d}_7(p - c_7)\frac{\partial}{\partial x}\text{vol}(p - c_0)| \leq \\ &\quad V^2 \max\{ \\ &\quad \max_p |(d_2(p) - d_0(p)|, \\ &\quad \max_p |(d_3(p) - d_1(p)|, \\ &\quad \max_p |(d_6(p) - d_4(p)|, \\ &\quad \max_p |(d_7(p) - d_5(p)|, \\ &\quad \}\end{aligned} \tag{31}$$

Combining Eq. 29 and Eq. 31, we arrive at Eq. 6 from the main paper, restated here,

$$\begin{aligned}\phi_x(s) \leq & V^3 \max_{v=0\dots7}\{\max_p |\frac{\partial}{\partial x}\hat{d}_v(p)|\} + \\ & V^2 \max_{v=[0,1,4,5]}\{\max_p |\hat{d}_{v+2}(p) - \hat{d}_v(p)|\}.\end{aligned} \tag{32}$$

13.4. Bounding d_v in terms of β

We still don't know the quantities $\max_p |\frac{\partial}{\partial x}\hat{d}_v(p)|$ and $\max_p |\hat{d}_{v+2}(p) - \hat{d}_v(p)|$ from Eq. 32 a priori. Thus we bound the first and second terms from Eq. 32 as follows.

For the first term, we can expand the absolute value,

$$\max_p |\frac{\partial}{\partial x}\hat{d}_v(p)| = \max\{|\max_p \frac{\partial}{\partial x}\hat{d}_v(p)|, |\min_p \frac{\partial}{\partial x}\hat{d}_v(p)|\}. \tag{33}$$

The max and the min can then be bounded by recalling Eq. 23, but rounding each β_i up or down to its maximum or minimum over the voxel depending on the sign of the weight,

$$\max_p \frac{\partial}{\partial x}\hat{d}_v(p) \leq \sum_i \mathcal{F}(v)^{(i)} \tau_{v,i}(p) \tag{34}$$

$$\min_p \frac{\partial}{\partial x}\hat{d}_v(p) \leq \sum_i \mathcal{F}(v)^{(i)} \rho_{v,i}(p) \tag{35}$$

$$\tau_{v,i}(p) = \begin{cases} \max_p \frac{\partial}{\partial x}\beta_i(p) & \mathcal{F}(v)^{(i)} \geq 0 \\ \min_p \frac{\partial}{\partial x}\beta_i(p) & \mathcal{F}(v)^{(i)} < 0 \end{cases} \tag{36}$$

$$\rho_{v,i}(p) = \begin{cases} \max_p \frac{\partial}{\partial x}\beta_i(p) & \mathcal{F}(v)^{(i)} < 0 \\ \min_p \frac{\partial}{\partial x}\beta_i(p) & \mathcal{F}(v)^{(i)} \geq 0 \end{cases} \tag{37}$$

For the second term from Eq. 32, we can similarly expand the absolute value,

$$\max_p |d_2(p) - d_0(p)| = \max\{ |\max_p d_2(p) - \min_p d_0(p)|, |\max_p d_0(p) - \min_p d_2(p)|\}, \tag{38}$$

and each max and min can be bounded as,

$$\max_p d_v(p) \leq \sum_i \mathcal{F}(v)^{(i)} \mu_{v,i}(p) \tag{39}$$

$$\min_p d_v(p) \leq \sum_i \mathcal{F}(v)^{(i)} \omega_{v,i}(p) \tag{40}$$

$$\mu_{v,i}(p) = \begin{cases} \max_p \beta_i(p) & \mathcal{F}(v)^{(i)} \geq 0 \\ \min_p \beta_i(p) & \mathcal{F}(v)^{(i)} < 0 \end{cases} \tag{41}$$

$$\omega_{v,i}(p) = \begin{cases} \max_p \beta_i(p) & \mathcal{F}(v)^{(i)} < 0 \\ \min_p \beta_i(p) & \mathcal{F}(v)^{(i)} \geq 0 \end{cases} \tag{42}$$

13.5. Bounding β

To estimate $\max_p \beta_i(p)$ and $\min_p \beta_i(p)$, we densely sample each β_i over the domain of one voxel, and then we take the minimum and maximum. In practice we use a sample rate of $15 \times 15 \times 15$, for a resolution of approximately 2.7 mm. To estimate $\max_p \frac{\partial}{\partial x}\beta_i(p)$ and $\min_p \frac{\partial}{\partial x}\beta_i(p)$, we use finite differences on the previously sampled points.

13.6. Summary

In this section we have derived a local gradient bound $\phi(s)$ that can be computed in terms of the minimum and maximum values and gradients of the bases, which are estimated offline and are therefore available for free at test time, as well as the voxel features $\mathcal{F}(v)$.

The equations throughout this section are readily implemented as massively parallel GPU operations, and on average it takes about 170 ms to compute ϕ over the whole scene, for ScanNet [8].

14. ScanNet++ results

In Fig. 7 and Table 4, we show our results on the ScanNet++ dataset [39]. These results are obtained using the same architecture and hyperparameters as our ScanNet model, showing that our model can easily be adapted to new data.

	F-Score ↑	Chamf. (cm) ↓	Mesh size (MB) ↓	Per-frame (ms) ↓	SEL (s) ↓
Ours	82.7	3.83	7.78	7.0	1.34

Table 4. Reconstruction metrics for our method on ScanNet++ [39]. The accuracy metrics, F-score and Chamfer distance, are quite strong relative to the results on ScanNet [8] (e.g. F-score of 82.7 vs. 74). We presume this is due to the higher-quality ground truth in ScanNet++, which uses stationary laser scanners instead of structured-light depth cameras, leading to more accurate training and evaluation.

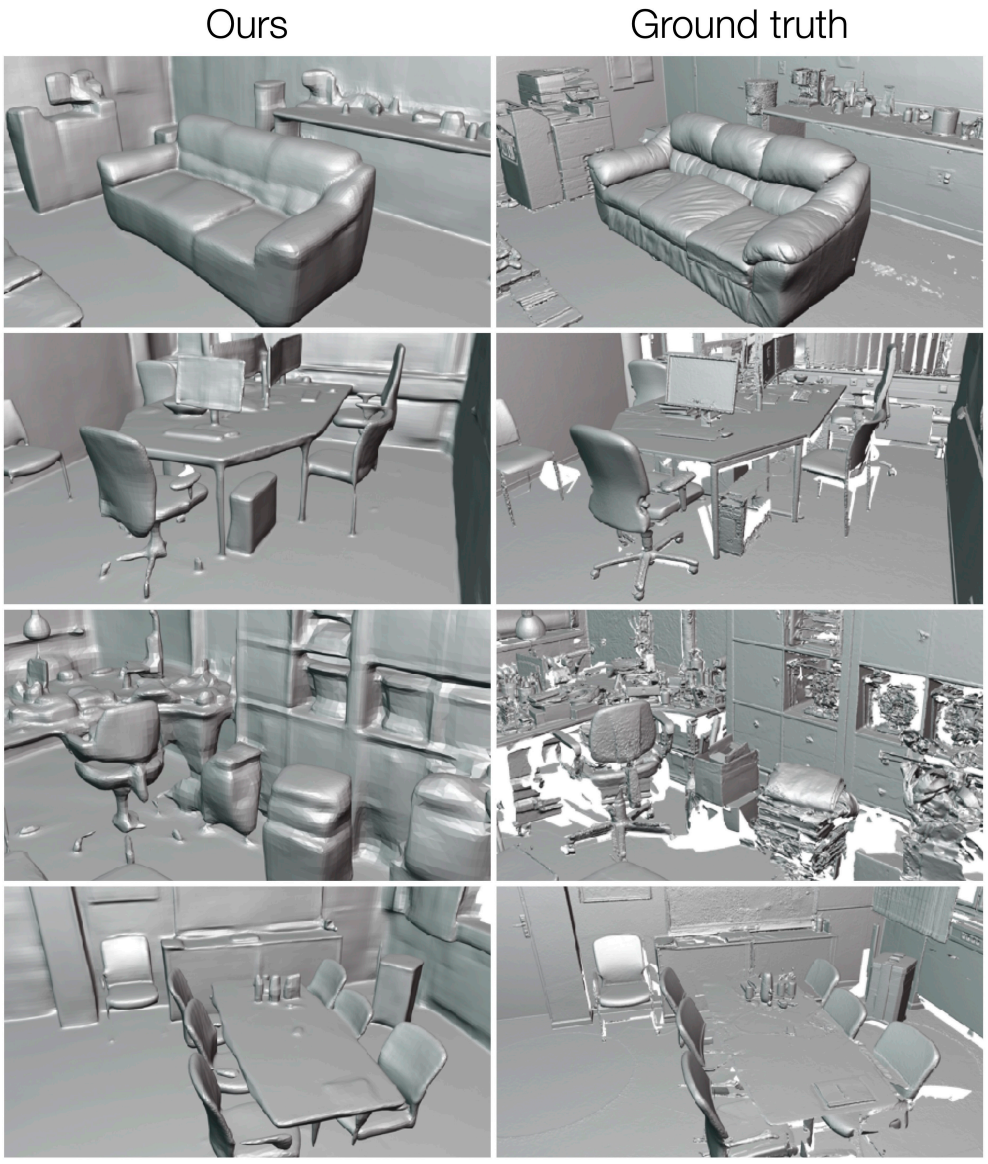

Figure 7. Qualitative results on the ScanNet++ dataset [39].