This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

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

Noah Stier **Alex Rich** **Pradeep Sen** **Tobias Höllerer**
{noahstier, anrich, psen, holl}@ucsb.edu
University of California, Santa Barbara

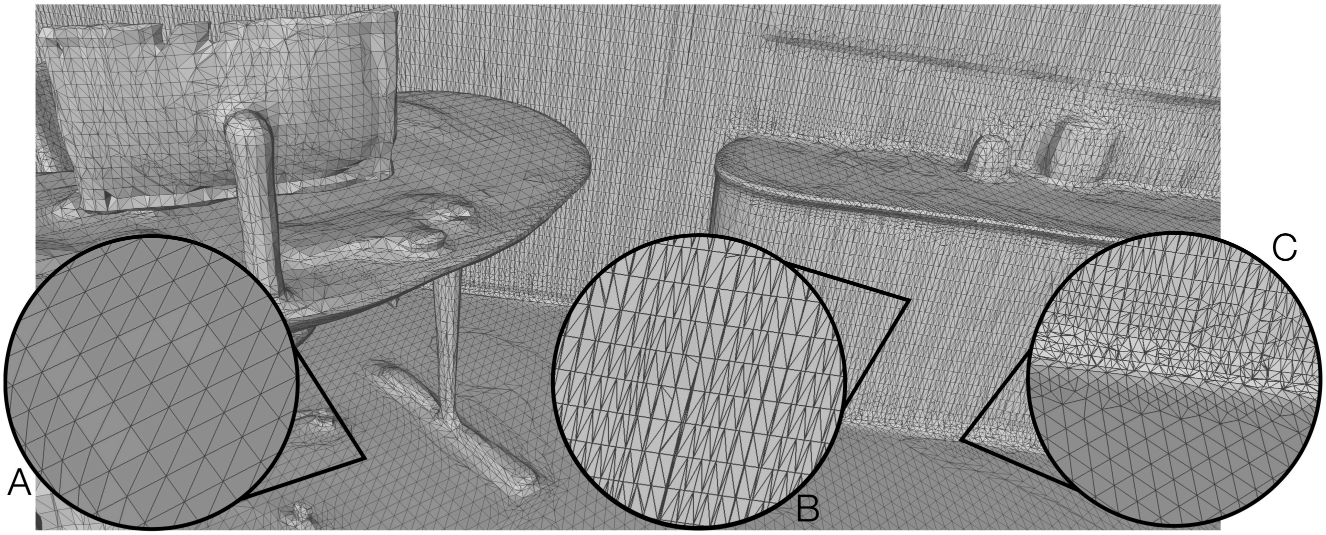

Figure 1. **Reconstruction with adaptive & anisotropic resolution.** Our 3D reconstruction system allocates computational resources effectively, using a variable sample pattern that adapts to the local surface complexity and orientation. **A**) The floor, which is flat and consistently oriented, is accurately reconstructed with a coarse resolution (4 cm). **B**) The cabinet is coarsely sampled in the vertical dimension because it exhibits no variation along that axis. Horizontally, however, it is sampled finely enough to accurately localize the surface. **C**) Object boundaries and other high-curvature structures are densely sampled to avoid aliasing, and to maintain high fidelity. The scene shown here is from ScanNet++ [39].

Abstract

Recent image-based 3D reconstruction methods have achieved excellent quality for indoor scenes using 3D convolutional neural networks. However, they rely on a high-resolution grid in order to achieve detailed output surfaces, which is quite costly in terms of compute time, and it results in large mesh sizes that are more expensive to store, transmit, and render. In this paper we propose a new solution to this problem, using adaptive sampling. By re-formulating the final layers of the network, we are able to analytically bound the local surface complexity, and set the local sample rate accordingly. Our method, AniGrad [1] *, achieves an order of magnitude reduction in both surface extraction latency and mesh size, while preserving mesh accuracy and detail.*

1. Introduction

3D reconstruction from images is a key capability for many important applications such as augmented reality, robotics, and 3D digital asset creation. In order to achieve high-quality reconstructions at interactive speeds, many recent methods [2, 5, 12, 18, 22, 32–35] focus on using feed-forward end-to-end neural networks. These models can generalize to new scenes at test time, and they do not require time-consuming direct optimization or volume rendering.

The common approach among these methods is to build a 3D feature volume by back-projecting deep image features using their camera parameters, and then use a 3D convolutional neural network (CNN) to estimate a 3D truncated signed distance function (TSDF) over the scene. This framework has achieved excellent results, but it is limited in terms of resolution due to the cubic cost of its voxel-based

[1] https://github.com/noahstier/AniGrad

representation. To address this, FineRecon [34] proposed to keep voxel size fixed at 4 cm, and to sub-sample on a $4 \times 4 \times 4$ grid of query points within each near-surface voxel, to achieve a higher resolution. This introduces a two-level resolution hierarchy: a coarse 4 cm voxel grid, and a fine 1 cm sub-sample grid. While that strategy does enable more detail, it requires long inference times ($>$ 30 s), which is prohibitive for online applications.

In this paper we propose a solution to this bottleneck, based on the observation that not all near-surface voxels need to be sub-sampled at such a high resolution. Rather, the ideal sub-sample rate is proportional to the degree of local surface variation: flat regions like walls and floors require a lower sampling density than thin or curved structures like the legs of a chair. Furthermore, the ideal sub-sample rate is often asymmetrical across dimensions: the TSDF near the floor varies highly along the gravitational axis, but it is nearly constant along any axis parallel to the floor surface. Therefore, we posit that the ideal sub-sample rate is both adaptive and anisotropic. Note: throughout this paper, we use a fixed 4 cm voxel size; all references to adaptive/local resolution and sample rates are referring to the sub-sample rate, illustrated in Fig 2.

Finding the ideal local resolution is challenging, because it presents a chicken-and-egg problem: we need the local surface structure in order to determine the correct resolution, but we do not know the surface structure until we have reconstructed it.

We propose a solution to this problem. We re-formulate the final layers of the network to give us advance knowledge of the predicted geometry, before sampling any TSDF predictions. Specifically, we re-interpret each voxel's feature vector as a weight vector for a linear combination of local 3D basis functions. The advantage is that just by examining the predicted basis weights, we can gain insight into the local TSDF structure, before we begin the sub-sampling process. This insight enables us to compute the appropriate sub-sample rates. The strategy that we develop is *guaranteed* not to alias the predicted TSDF, beyond a user-selected minimum variation threshold. The key contributions of this paper are as follows.

1. We propose to express the predicted TSDF as a linear combination of learned basis functions, and we re-formulate the network to output this representation.
2. Using this representation, we introduce an algorithm for adaptive, anisotropic sampling that enables 3D reconstruction with an order of magnitude fewer query points, while guaranteeing a user-specified, acceptable degree of aliasing.
3. We demonstrate this strategy using the ScanNet dataset [8], showing that it results in an order of magnitude reduction in both surface extraction latency and mesh size, while maintaining state-of-the-art accuracy.

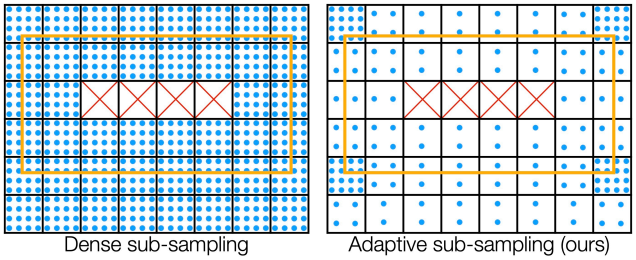

Figure 2. We capture fine details without reducing the voxel size (black). Instead, we sample our query points (blue) at a sub-voxel resolution. Unlike FineRecon [34] (left), our sub-sampling rate (right) adapts to the local predicted surface (yellow), enabling significant time savings. The red Xs indicate voxels that are pruned by occupancy filtering before sub-sampling, because they are too far from a surface.

2. Related Work

3D CNN-based TSDF reconstruction. Image-based 3D reconstruction is a classic problem, traditionally solved by multi-view stereo (MVS) [13, 30, 31], with recent MVS methods based on deep learning [3, 17, 26–28]. Here, we focus on the recent methods that are most related to ours, that use a 3D CNN to reconstruct a scene as a TSDF volume. This method was first used in the RGB-D domain [9, 10], and Atlas [22] proposed to handle the RGB-only case by densely back-projecting image features over the scene. Since then, numerous improvements have been introduced including sparse convolutions [35], multi-view fusion mechanisms [2, 12, 32], visibility and geometric constraints [14, 24, 40], self-supervision [15, 19], SLAM pose updates [33], depth guidance [16, 18, 34], and refinement via differentiable rendering [41, 43]. Our work builds on this research, focusing on adaptive resolution for efficiency.

Adaptive resolution in 3D reconstruction. Adaptive resolution is highly desirable in most 3D reconstruction scenarios, and many solutions have been proposed. Occupancy filtering is a common approach that discards free space to focus on surfaces [12, 18, 32, 35]. Related approaches use semantic segmentation or object detection to assign higher resolution to certain object types [4, 38, 42]. Iterative mesh subdivision is common in multi-view stereo [11, 20, 23]. Several methods set the local resolution based on viewing distance [36, 37], and others use active depth sensors to estimate local surface complexity [42]. In contrast, our adaptive resolution is fully automatic, RGB-only, agnostic to object type, and does not require any form of preliminary explicit 3D model. Ours is also the first, to our knowledge, to study adaptive resolution in the context of image-based TSDF reconstruction with 3D CNNs, following Atlas [22].

Sub-voxel resolution. With a voxel-based method, it is not tractable to reduce the voxel size indefinitely in pursuit of greater detail. Reconstructing sub-voxel detail has thus been a topic of recent interest. In this vein, FineRecon [34] is the most similar method to ours. It uses trilinear interpolation to sub-sample its 4 cm feature grid on a 1 cm grid of query points, achieving high resolution at the cost of high inference time. At the core of this paper, our re-formulation with local basis functions allows us to replace that operation with a new algorithm for adaptive sub-voxel resolution. Fig. 2 illustrates the resulting large reduction in query points.

3. Intuition and proposed approach

We first observe that the ideal local resolution is closely tied to the gradient of the predicted TSDF. For instance, if the maximum absolute gradient within some local volume is known to be $[0, 0, 0]$, then the predicted TSDF is locally constant, and a single query point is sufficient (a sample rate of 1 in all dimensions). If instead the gradient is higher in a particular dimension, additional query points are necessary to avoid aliasing.

The key question is, how can we know the maximum absolute gradient within a local volume? We could sample many query points and compute the per-point gradient, but this would still require many samples to find the maximum. Instead, our solution is to express the TSDF prediction as a weighted sum of local 3D basis functions (Section 4.2, Eq. 1), where our network predicts the basis weight vector per voxel. This re-formulation allows us to use a shortcut: we analyze the basis functions in a *one-time, offline* step to determine their individual maximum values and gradients. Then, at test time, we can use this information to quickly bound the maximum gradient of the weighted sum, as shown in Section 5.1.

Finally, we set the local sample rate proportionally to the gradient bound. This defines a non-uniform grid over the scene, and we extract a mesh using a non-uniform variant [29] of marching cubes [21] (see Section 5.2).

4. Model architecture

Our system takes as input a set of RGB images, along with the camera intrinsics and extrinsics, and reconstructs the scene by predicting the TSDF at a set of query points $p \in \mathbb{R}^3$.

4.1. Building the scene feature volume

We use a 2D-3D convolutional backbone to generate a 3D feature volume in scene space. Our implementation is based on FineRecon [34], and we summarize the main steps here.

1. A 2D CNN extracts a 2D feature map from each image.
2. The 2D features are densely back-projected into scene space to populate an initial 3D feature volume.
3. A 3D U-Net processes the 3D features to produce the final scene feature volume $\mathcal{F}$ and a predicted occupancy volume $\hat{O}$.

4.2. TSDF prediction with fixed basis functions

Our key architectural novelty is to re-interpret the voxel feature $\mathcal{F}(v)$ as a basis weight vector for a set of local basis functions β_i centered at voxel v. Each nearby voxel v makes an independent TSDF prediction $\hat{d}_v$ for query point $p = [x, y, z]$ as follows:

$$\hat{d}_v(p) = \sum_i \mathcal{F}(v)^{(i)} \beta_i(p - c_v). \qquad (1)$$

$\mathcal{F}(v)^{(i)}$ is the i^{th} element of the voxel feature vector $\mathcal{F}(v)$, and c_v is the center of voxel v. To represent the basis functions in a compact and easily-learnable way, we express each basis function β_i as the i^{th} output channel of a coordinate MLP θ,

$$\beta_i(p) = \theta(p)^{(i)}. \qquad (2)$$

To reduce voxel boundary artifacts, we aggregate the predictions from nearby voxels using a volume-weighted average to predict the final TSDF,

$$\hat{d}(p) = \sum_{v \in N(p)} \hat{d}_v(p)\text{vol}(p - c_v), \qquad (3)$$

where the neighborhood $N(p)$ is the set of voxels whose centers are found by rounding each of eight $p = [x, y, z]$ either up or down to the nearest whole voxel size. The weights $\text{vol}(p - c_v)$ can be interpreted as tri-linear interpolation weights,

$$\text{vol}([x, y, z]) = |V - x| \cdot |V - y| \cdot |V - z|, \qquad (4)$$

where V is the voxel size.

With this representation, the 3D CNN's output volume $\mathcal{F}$ defines a continuous TSDF field in $\mathbb{R}^3$, since $\mathcal{F}$ consists of weights for continuous basis functions. To extract an explicit mesh surface, we must first discretize this field by sampling it, evaluating $\hat{d}$ at a set of query points.

4.3. Occupancy filtering

When making our TSDF predictions, we simply ignore any voxels with occupancy probability $\hat{O} < 0.5$, effectively masking them out for the sake of surface extraction.

5. Surface Extraction

When we set our local sample rates, we do not set them for each voxel. We instead set them for each *central cube* (CC), as illustrated in red in Fig. 3a. This is because of the smoothing defined in Eq. 3, which means that the TSDF at a given point depends on the feature vectors of all eight neighboring voxels, and the TSDF is thus continuously differentiable within a CC. We adopt the CC as the basic spatial unit for adaptive sub-sampling; in other words, we take the CC to be the "local volume" referenced in Section 3.

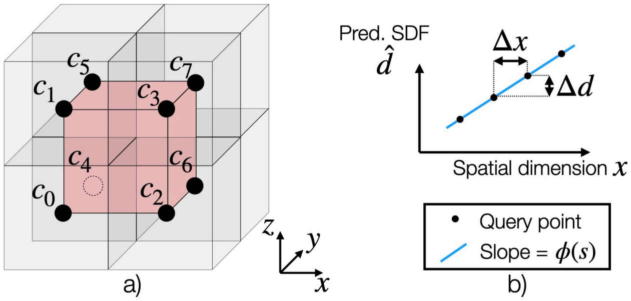

Figure 3. **a)** Our predicted TSDF is continuously differentiable within the central cube (CC, red). **b)** For each CC s we compute the maximum gradient bound $\phi(s)$, shown here in a 1D example as the slope of the blue line. By setting the spatial sampling rate to $\Delta x = \frac{\Delta d}{\phi_x(s)}$, we are **guaranteed** to capture every predicted TSDF change of at least Δd.

5.1. Adaptive sample rates

We set our local resolution proportionally to the maximum TSDF gradient within a CC. However, we do not know this maximum gradient a priori. We could estimate it with a Monte Carlo approach, but it would undermine our goal of reducing the required sample count. Instead, we analytically bound the gradient within the CC. For brevity, we present the process for one dimension only, developing an upper bound on $|\frac{\partial}{\partial x}\hat{d}(p)|$ that allows us to define the appropriate sample rate in the x dimension. We name this bound $\phi_x(s)$, for a CC that we call s. By definition,

$$\phi_x(s) \geq \max_{p \in s} |\frac{\partial}{\partial x}\hat{d}(p)|. \tag{5}$$

By differentiating, we find that $\phi_x(s)$ can be bounded in terms of each nearby voxel's maximum gradient and the maximum TSDF differences across voxels (full derivation in Supp.),

$$\begin{aligned}\phi_x(s) \leq \max_{v=0\ldots7}\{\max_p |\frac{\partial}{\partial x}\hat{d}_v(p)|\}+ \\ \max_{v=[0,1,4,5]}\{\max_p |\hat{d}_{v+2}(p) - \hat{d}_v(p)|\}.\end{aligned} \tag{6}$$

Note, however, the terms $\max_p |\frac{\partial}{\partial x}\hat{d}_v|$ and $\max_p |\hat{d}_v - \hat{d}_w|$. We do not know these per-voxel maxima a priori. Therefore, we further develop upper bounds on those terms. Here we only show this process for the first term, $\max_p |\frac{\partial}{\partial x}\hat{d}_v|$, leaving the second for the Supp. We can expand the absolute value into a max and a min,

$$\max_p |\frac{\partial}{\partial x}\hat{d}_v(p)| = \max\{|\max_p \frac{\partial}{\partial x}\hat{d}_v(p)|, |\min_p \frac{\partial}{\partial x}\hat{d}_v(p)|\}. \tag{7}$$

Considering just the max term for illustration, and leaving the details for the Supp., we find,

$$\max_p \frac{\partial}{\partial x}\hat{d}_v(p) \leq \sum_i \mathcal{F}(v)^{(i)} \cdot \begin{cases} \max_p \frac{\partial}{\partial x}\beta_i(p) & \mathcal{F}(v)^{(i)} \geq 0 \\ \min_p \frac{\partial}{\partial x}\beta_i(p) & \mathcal{F}(v)^{(i)} < 0 \end{cases}. \tag{8}$$

The terms $\max_p \frac{\partial}{\partial x}\beta_i(p)$ and $\min_p \frac{\partial}{\partial x}\beta_i(p)$ can be computed in a one-time offline step, after training: we densely sample each basis over the domain of one voxel, and compute the gradient by finite differences. Eq. 8 is an important result: it says that we can bound the TSDF gradient within a voxel as a function of the voxel feature, and the maximum gradients of the individual bases, which are available for free at test time. Plugging this back into Eq. 7 and then Eq. 6, we can quickly compute ϕ_x for each CC.

We then compute the desired distance between sample points as

$$\Delta x = \frac{\Delta d}{\phi_x(s)}. \tag{9}$$

By spacing our samples at most Δx apart, we are **guaranteed** to capture every predicted TSDF change of at least Δd. This is because ϕ is a gradient bound, so the predicted TSDF variation cannot exceed it. Thus, the parameter Δd represents a maximum acceptable degree of aliasing. In practice, our final per-CC sample rates are clamped and rounded up to the nearest integer,

$$R_x = \min(\lceil \frac{1}{\Delta x} \rceil, R_{\max}), \tag{10}$$

where $R_{\max}$ is a cap that can be set to avoid diminishing returns at high resolutions.

5.2. Non-uniform marching cubes

We use dual marching cubes (DMC) [29] to extract the zero-isosurface. DMC was introduced as a way to run marching cubes [21] on octrees. Its main steps are 1) generate the *dual grid* by connecting sample points as shown by the dotted lines in Fig. 4, and 2) run marching cubes on the dual grid. Degenerate dual cells (appearing as dotted triangles in Fig. 4) are handled by duplicating the cell vertices to make them topological cubes.

To adapt DMC to our problem setting, we make two modifications. First, we skip two of DMC's preliminary stages (feature isolation and octree construction) that are designed to generate and optimize the sample point locations. This is because in our framework the sample point locations are already determined by our adaptive sampling strategy. Second, whereas DMC recurses through its multi-level octree using a fixed number of operations per octree cell, we instead build the dual grid iteratively, computing the centers and the connectivity of the dual cells as a function of the adaptive sample rates.

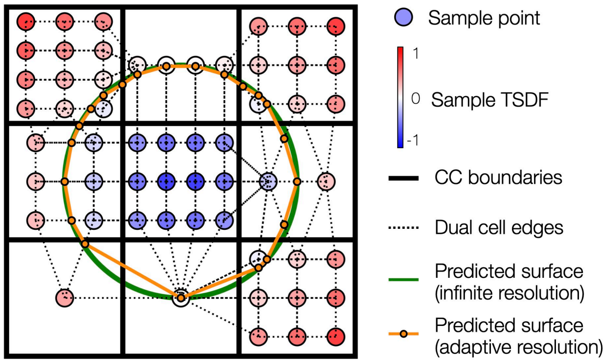
Sample point
1
0 Sample TSDF
-1
CC boundaries
Dual cell edges
Predicted surface (infinite resolution)
Predicted surface (adaptive resolution)

Figure 4. Each central cube (CC) has a sample rate in each dimension, chosen arbitrarily for this 2D illustration. Dual marching cubes (DMC) [29] generates the *dual grid* by connecting the sample points as shown by the dotted lines, and then it runs marching cubes [21] on the resulting graph, handling degenerate cells by duplicating their vertices to make them topological cubes. The green line shows the zero-isosurface of the underlying continuous TSDF, and the yellow line shows the surface extracted by adaptive sampling and DMC.

6. Training

To train our system we assume the existence of a ground truth surface mesh for each scene, and we train in a fully-supervised manner.

6.1. Ground truth generation

We sample two sets of training points from the ground truth mesh. First, P_{surf} is a set of points obtained by uniform random sampling over the mesh surface, and for each point $p \in P_{\text{surf}}$ we also sample the local mesh surface normal $\vec{S}(p)$. Second, P_{unif} is a set of points sampled on a 4 cm regular grid throughout the entire volume of the scene. We compute the ground truth TSDF $d(p), p \in P_{\text{unif}}$ by rendering the ground truth mesh to a set of depth maps, and running TSDF fusion [6]. We also define the ground truth occupancy O as

$$O(p) = |d(p)| < t, p \in P_{\text{unif}}, \tag{11}$$

where t is the truncation distance parameter of the TSDF fusion. Similar to Atlas [22] we then mark any entirely unobserved columns of points as unoccupied, to avoid producing artifacts outside the scene walls. However, we use a cumulative product test to make sure we only apply this label outside of the scene walls, and not in the center of the scene, which is frequently unobserved. This prevents the pillar artifacts that can be seen in previous work [22]. Finally, we filter by occupancy to further define a new subset of points that are near surfaces, $P_{\text{occ}} = \{p : p \in P_{\text{unif}}, O(p) = 1\}$.

6.2. Loss function

Unlike previous works that rely solely on direct TSDF supervision, we find that supervising the TSDF gradient leads to better results. Inspired by Neural Poisson [7], we begin with a squared-error gradient loss,

$$\mathcal{L}_{\text{grad}}(p) = \|\nabla\hat{d}(p) - \vec{S}(p)\|_2^2, \;\; p \in P_{\text{surf}}, \tag{12}$$

where $\nabla\hat{d}$ is the analytical gradient of our TSDF prediction and $\vec{S}$ is the ground truth mesh surface normal. We find that to get the best quality, an additional constraint is necessary to enforce 0 TSDF magnitude at the surface,

$$\mathcal{L}_{\text{surf}}(p) = |\hat{d}(p)|, \;\; p \in P_{\text{surf}}. \tag{13}$$

We then apply a weaker form of direct TSDF supervision based on the smooth L_1 loss [25],

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & |x| \geq 1, \end{cases} \tag{14}$$

and we only apply this loss within occupied regions. Our TSDF loss is thus

$$\mathcal{L}_{\text{dist}}(p) = \text{smooth}_{L_1}(\hat{d}(p) - d(p)), \;\; p \in P_{\text{occ}} \tag{15}$$

The occupancy predictions are trained on points uniformly sampled throughout the scene,

$$\mathcal{L}_{\text{occ}}(p) = \text{BCE}(\hat{O}(p), O(p)), \;\; p \in P_{\text{unif}}. \tag{16}$$

We finally add the basis gradient loss,

$$\mathcal{L}_{\text{basis}}(p) = |\mathcal{F}(v)^{(i)}\nabla\beta_i(p)|, \;\; p \in P_{\text{unif}} \cup P_{\text{surf}}, v \in N(p). \tag{17}$$

Without this loss, we notice that the network may reconstruct low-gradient regions, such as empty space, by summing two bases with high gradient in opposite directions. This causes us to estimate a high gradient bound and thus over-sample. This loss encourages the network to learn a minimal-gradient representation, which allows us to reduce our sample rates. Our combined training loss is then,

$$\mathcal{L} = \lambda_g\mathcal{L}_{\text{grad}} + \lambda_s\mathcal{L}_{\text{surf}} + \lambda_d\mathcal{L}_{\text{dist}} + \lambda_o\mathcal{L}_{\text{occ}} + \lambda_b\mathcal{L}_{\text{basis}}. \tag{18}$$

7. Implementation details

Like FineRecon [34], we find that there are diminishing returns at resolutions finer than 1 cm. We therefore set the parameter $\Delta d = 1$ cm, aiming to resolve structures at least 1 cm thick. Similarly, we set $R_{\max} = 4$, thus clamping our finest resolution to 1 cm. We set $\lambda_s = \lambda_d = \lambda_o = \lambda_b = 1$ and $\lambda_g = 3$. Training takes about 30 hours with two NVIDIA 4090 GPUs, and testing for all methods is done on one NVIDIA 4090.

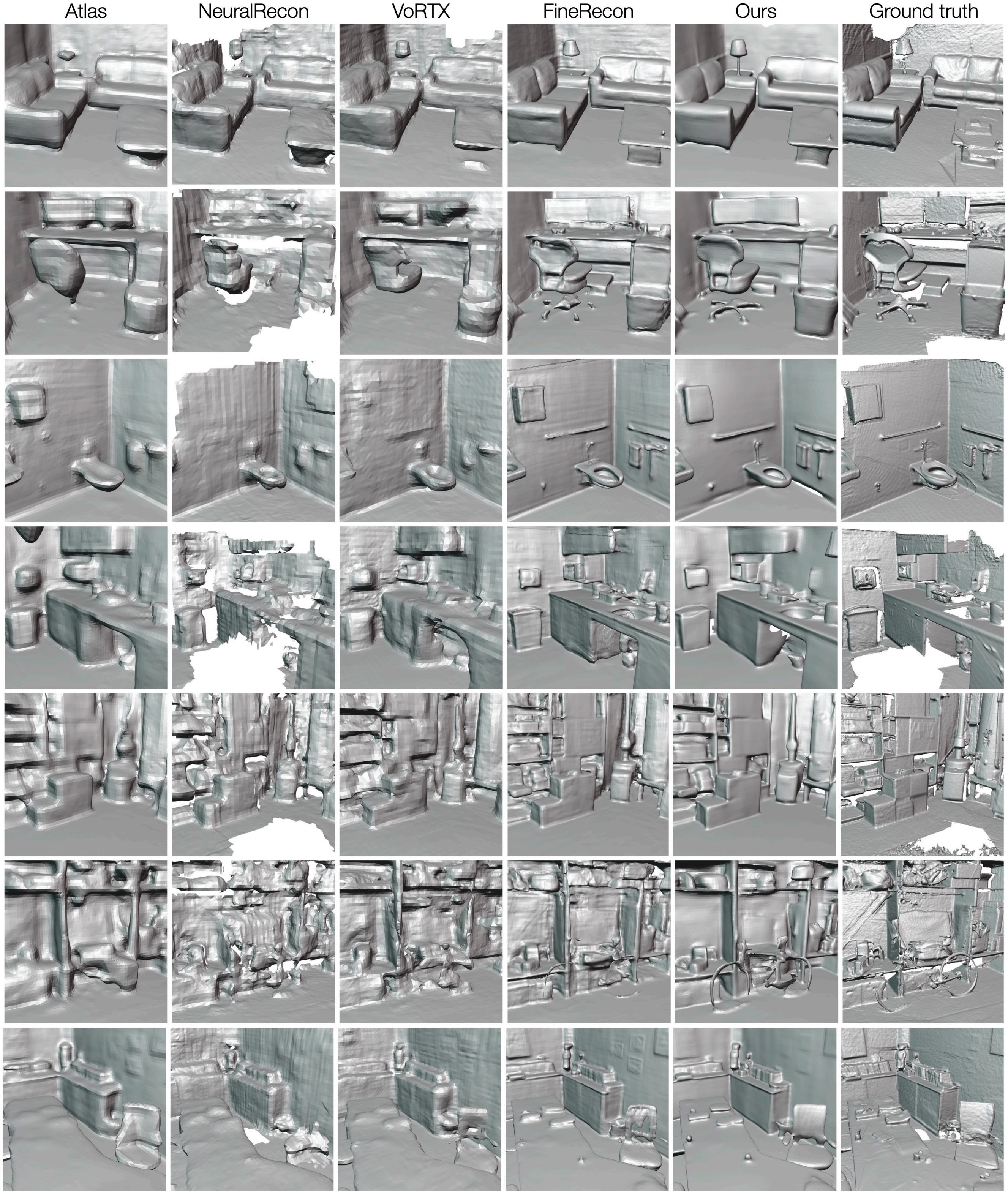

Figure 5. Qualitative comparison on the ScanNet dataset [8]. In terms of accuracy, our results are on par with the state of the art (FineRecon [34]). However, as shown in Table 1, **our method is an order of magnitude faster** to extract surfaces. Our surfaces also tend to be quite smooth, without the bumpy textures produced by the other methods (Atlas [22], NeuralRecon [35], VoRTX [32], and FineRecon [34]).

Method	Resolution	F-Score ↑	Chamf. (cm) ↓	Mesh size (MB) ↓	Per-frame (ms) ↓	SEL (s) ↓
Atlas	4cm (uniform)	65.4	6.13	2.58	29.4	0.93
NeuralRecon	4cm (uniform)	60.4	7.75	3.59	8.4	0.20
VoRTX	4cm (uniform)	69.5	5.59	2.13	7.3	2.04
FineRecon	1cm (uniform)	74.1	4.39	54.85	37.8	24.81
Ours	1cm (adaptive)	74.0	5.29	5.47	10.2	1.11

Table 1. Quantitative results on ScanNet. Our model reduces output mesh size and surface extraction latency (SEL) each by an order of magnitude relative to the state of the art (FineRecon [34]), with virtually no loss in F-score.

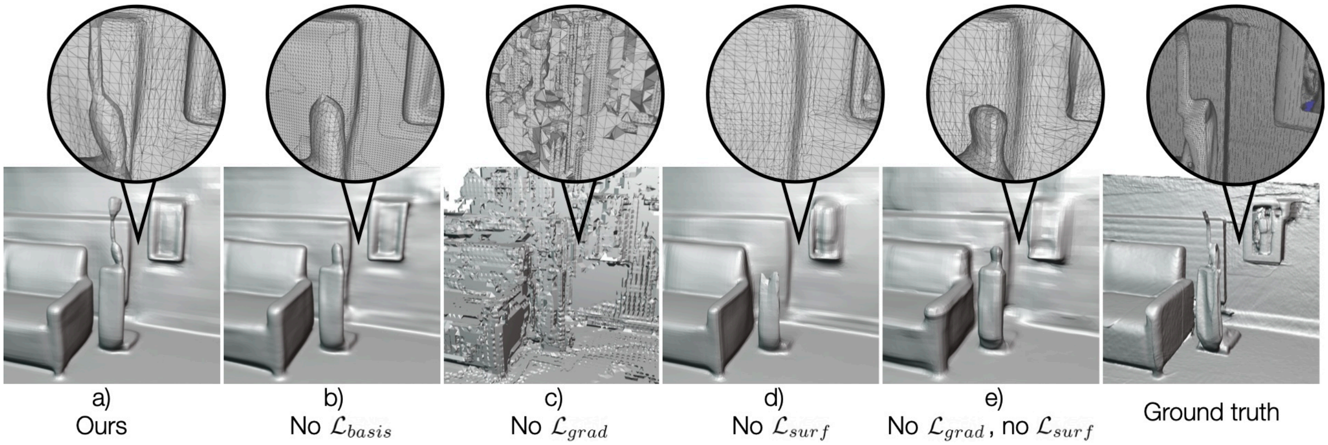

Figure 6. Visual ablation study. Panels a-e correspond to Table 2 row a-e. All of these loss components are critical to achieving good surface quality with a low SEL.

	$\mathcal{L}_{basis}$	$\mathcal{L}_{grad}$	$\mathcal{L}_{surf}$	F-score ↑	Chamf. ↓ (cm)	SEL ↓ (s)
a)	✓	✓	✓	74.0	5.29	1.11
b)		✓	✓	74.0	5.16	2.8
c)	✓		✓	73.1	6.33	2.4
d)	✓	✓		70.1	5.68	0.94
e)	✓			71.4	5.62	1.0

Table 2. Ablation study on ScanNet. We ablate L_{basis}, L_{grad}, and L_{surf}, because they are new additions relative to prior works in this area. We observe that L_{basis} and L_{grad} are important for surface extraction latency (SEL). L_{grad} and L_{surf} are both important for mesh quality, as shown here and in Figure 6.

8. Experiments

8.1. Reconstructing ScanNet

We evaluate our method on the ScanNet dataset [8] of 1,613 indoor scans. We use the official train/val/test split, and we report all results on the test set (100 scenes).

Baselines. We compare our method to several others based on 3D CNNs. Atlas [22] is the first such system to our knowledge. NeuralRecon [35] and VoRTX [32] use sparse convolutions to manage computational cost, while FineRecon [34] uses depth guidance from SimpleRecon [28] to achieve very high quality, with longer compute times.

Metrics. We report F-Score and Chamfer distance using the method from TransformerFusion [2]. F-score measures coarse mesh accuracy at a 5 cm threshold, whereas Chamfer distance factors in shorter-range error (see Supp. for full definitions). We measure the per-frame time to extract features from each image and fuse them into the feature volume, and we measure the surface extraction latency (SEL). SEL is the time to obtain a surface mesh from the feature volume, and it includes the 3D CNN, any post-processing, and polygonization with marching cubes.

Qualitative results. Fig. 5 shows that our meshes are similar to FineRecon [34] overall, but they tend to be smoother, eliminating FineRecon's bumpy texture. Thus we have reduced the inference time while improving the visual quality.

Quantitative results. Table 1 confirms that our coarse accuracy, as measured by F-score, is on par with the state of the art. Our Chamfer distance is second-best; we attribute this to our choice not to use depth guidance, resulting in faster per-frame time than FineRecon, at the cost of missing some details. Table 3 shows a breakdown of SEL and its components. Ours incurs an average cost of 170ms to compute the adaptive sample rates, but this saves over 21s overall relative to FineRecon.

Ablation study. In Table 2 and Fig. 6 we study the effects of various components of the supervision. In b, we show that disabling $\mathcal{L}_{basis}$ leads to oversampling and a factor of three increase to SEL. In c, we see that when $\mathcal{L}_{grad}$ is disabled, $\mathcal{L}_{surf}$ squashes all near-surface TSDF predictions to zero, causing major surface artifacts. In d and e, we observe that disabling $\mathcal{L}_{surf}$, or $\mathcal{L}_{surf}$ and $\mathcal{L}_{grad}$ together, significantly degrades the surface quality.

	Transformer	3D CNN	Sample rate computation	Sub-sampling	Marching cubes / DMC	Total SEL	# samples
Atlas	-	0.74	-	-	0.19	0.93	22.1M
NeuralRecon	-	0.11	-	-	0.09	0.20	2.4M
VoRTX	1.77	0.20	-	-	0.07	2.04	0.5M
FineRecon	-	0.14	-	21.68	2.58	24.81	25.9M
Ours (unif.)	-	0.12	-	3.27	2.02	4.90	25.9M
Ours	-	0.12	0.17	0.35	0.48	1.11	3.3M

Table 3. Breakdown of the time costs for surface extraction latency (SEL), in seconds. Ours (unif.) has the same architecture and weights as Ours, but uses a dense 1 cm sampling strategy like FineRecon. Comparing Ours (unif.) to FineRecon shows that our sub-sampling is 6.6× faster due to our architectural changes (additional discussion in Supp.). Comparing Ours (unif.) to Ours shows a further 9.4× reduction in sub-sampling time due to our adaptive sampling, leading to a 22.3× overall SEL reduction relative to FineRecon. Note, Atlas uses a large fixed size scene volume leading to high sample count. NeuralRecon and VoRTX reduce this using sparsity (occupancy filtering). FineRecon and Ours (unif.) also leverage sparsity, but have high sample count due to dense sub-sampling.

8.2. Reconstructing ScanNet++

We have trained and tested our method on the newer ScanNet++ dataset [39], and we find that it generalizes well with no parameter changes needed. Fig. 1 shows an example. Previous works have not been thoroughly evaluated on this new dataset so it is difficult to provide fair baselines, but we include our metrics and qualitative results in the Supp.

9. Discussion

Mesh quality. We produce high-quality meshes, without resorting to expensive depth guidance. We are able to do this because our supervision is entirely based on the ground truth mesh, instead of using noisy depth maps, thus giving our model access to a higher-quality training signal.

Analytical gradient bounds. A core element of our work is the derivation of a fast and effective local gradient bound. It is also possible to derive such a bound for previous models [2, 34]; however, we found the resulting bounds were not tight enough to be useful, causing extreme over-sampling. Our basis functions are thus critical to our approach.

9.1. Limitations and future work

Our resolution is adaptive within each voxel, thus our resolution is never coarser than the voxel size. This presents an opportunity for further gains, particularly for large flat surfaces: in the future, perhaps we can reach across voxels to achieve a sample rate even coarser than the voxel size.

As shown in Fig. 1B and 1C, our method can result in thin, sliver-shaped triangles, which are not optimal in terms of mesh size and complexity. Our method still greatly reduces mesh size overall, but addressing these slivers may yield further improvement, possibly by regularizing the sample rates or by adding a re-meshing step.

Regarding representational capacity, we use a fixed number of basis functions (64). While the space of possible output geometries is quite large, there may be opportunities for improvement by allowing the network to steer the bases via rotation or other transforms. Furthermore, our sample rates are always defined along the coordinate axes. As a result, we expect the greatest efficiency when the scene surfaces are predominantly axis-aligned. A pre-alignment step might help maximize these gains, and future work may explore more flexible sampling patterns.

While the ideal sample distance defined in Eq. 9 provides a guarantee against aliasing, we note that this refers to aliasing of the model's predicted continuous TSDF field. It is of course possible to alias the true scene TSDF. In addition, with Eq. 10 we cap the maximum sample rate; thus, the guarantee is technically not valid in those capped regions, but we see very little evidence of aliasing in practice.

10. Conclusion

We have presented AniGrad, a 3D reconstruction system for monocular video that automatically adapts its resolution to the local 3D surface structure. To make this possible, we have introduced a new representation of the predicted TSDF as a linear combination of learned, local 3D basis functions. This representation enables us to efficiently compute bounds on the local TSDF gradient, and we set the local resolution proportionally to this bound. As a result, we can extract a mesh from our feature volume with a latency reduction of over 20×, without sacrificing accuracy. This method makes 3D reconstruction more accessible for compute-limited devices, which otherwise may not meet the necessary speed and throughput requirements. This is essential for applications such as mobile augmented reality and robotics that need to repeatedly extract surface meshes with low latency. Our method makes a step in this direction.

Acknowledgements

Support for this work was provided in part by ONR grant N00014-23-1-2118 as well as NSF grant IIS-2211784. We would also like to thank Ziyue Feng for many insightful discussions during the early stages of this project.

References

[1] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2): 31–39, 2010. 1

[2] 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. 1, 2, 7, 8

[3] Di Chang, Aljaž Božič, Tong Zhang, Qingsong Yan, Yingcong Chen, Sabine Süsstrunk, and Matthias Nießner. RC-MVSNet: Unsupervised multi-view stereo with neural rendering. In *European conference on computer vision*, pages 665–680. Springer, 2022. 2

[4] Chao Chen, Ruoyu Wang, Yuliang Guo, Cheng Zhao, Xinyu Huang, Chen Feng, and Liu Ren. AdaOcc: Adaptive-resolution occupancy prediction. *arXiv preprint arXiv:2408.13454*, 2024. 2

[5] Jaesung Choe, Sunghoon Im, Francois Rameau, Minjun Kang, and In So Kweon. VolumeFusion: Deep depth fusion for 3D scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16086–16095, 2021. 1

[6] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 5

[7] Angela Dai and Matthias Nießner. Neural poisson: Indicator functions for neural fields. *arXiv preprint arXiv:2211.14249*, 2022. 5

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias 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, 6, 7, 3, 4

[9] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. 2

[10] Angela Dai, Christian Diller, and Matthias Nießner. SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2020. 2

[11] Amaël Delaunoy and Emmanuel Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3D reconstruction problems dealing with visibility. *International journal of computer vision*, 95(2):100–123, 2011. 2

[12] Ziyue Feng, Leon Yang, Pengsheng Guo, and Bing Li. CVRecon: Rethinking 3D geometric feature learning for neural reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1, 2

[13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 2

[14] Huiyu Gao, Wei Mao, and Miaomiao Liu. VisFusion: Visibility-aware online 3D scene reconstruction from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17317–17326, 2023. 2

[15] Pengsheng Guo, Miguel Angel Bautista, Alex Colburn, Liang Yang, Daniel Ulbricht, Joshua M Susskind, and Qi Shan. Fast and explicit neural view synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3791–3800, 2022. 2

[16] Ziyang Hong and C Patrick Yue. Cross-dimensional refined learning for real-time 3D visual perception from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2169–2178, 2023. 2

[17] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2

[18] Jihong Ju, Ching Wei Tseng, Oleksandr Bailo, Georgi Dikov, and Mohsen Ghafoorian. DG-Recon: Depth-guided neural 3D scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18184–18194, 2023. 1, 2

[19] Runfa Li, Upal Mahbub, Vasudev Bhaskaran, and Truong Nguyen. MonoSelfRecon: Purely self-supervised explicit generalizable 3D reconstruction of indoor scenes from monocular RGB views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 656–666, 2024. 2

[20] Shiwei Li, Sing Yu Siu, Tian Fang, and Long Quan. Efficient multi-view surface refinement with adaptive resolution control. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 349–364. Springer, 2016. 2

[21] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 3, 4, 5

[22] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3D scene reconstruction from posed images. In *European conference on computer vision*, pages 414–431. Springer, 2020. 1, 2, 5, 6, 7

[23] Jan Neumann and Yiannis Aloimonos. Spatio-temporal stereo using multi-resolution subdivision surfaces. *International Journal of Computer Vision*, 47:181–193, 2002. 2

[24] Yu-Kun Qiu, Guo-Hao Xu, and Wei-Shi Zheng. Inner-outer aware reconstruction model for monocular 3D scene reconstruction. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 5

[26] Alexander Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer. 3DVNet: Multi-view depth prediction and volumetric refinement. In *2021 International Conference on 3D Vision (3DV)*, pages 700–709. IEEE, 2021. 2

[27] Alex Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer. Smoothness, synthesis, and sampling: Re-thinking unsupervised multi-view stereo with DIV loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[28] 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. 2, 7

[29] Scott Schaefer and Joe Warren. Dual marching cubes: Primal contouring of dual grids. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pages 70–76. IEEE, 2004. 3, 4, 5

[30] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016. 2

[31] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 519–528. IEEE, 2006. 2

[32] 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. 1, 2, 6, 7

[33] Noah Stier, Baptiste Angles, Liang Yang, Yajie Yan, Alex Colburn, and Ming Chuang. LivePose: Online 3D reconstruction from monocular video with dynamic camera poses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7921–7930, 2023. 2

[34] Noah Stier, Anurag Ranjan, Alex Colburn, Yajie Yan, Liang Yang, Fangchang Ma, and Baptiste Angles. FineRecon: Depth-aware feed-forward network for detailed 3D reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18423–18432, 2023. 2, 3, 5, 6, 7, 8, 1

[35] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021. 1, 2, 6, 7

[36] Emanuele Vespa, Nils Funk, Paul HJ Kelly, and Stefan Leutenegger. Adaptive-resolution octree-based volumetric SLAM. In *2019 International Conference on 3D Vision (3DV)*, pages 654–662. IEEE, 2019. 2

[37] Hoang-Hiep Vu, Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):889–901, 2011. 2

[38] Weidong Wang, Yu Hu, Wei Xi, Danping Zou, and Wenxian Yu. Efficient semantic-aware tsdf mapping with adaptive resolutions. In *2023 3rd International Conference on Robotics, Automation and Artificial Intelligence (RAAI)*, pages 39–45. IEEE, 2023. 2

[39] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 1, 8, 3, 4

[40] Ruihong Yin, Sezer Karaoglu, and Theo Gevers. Geometry-guided feature learning and fusion for indoor scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3652–3661, 2023. 2

[41] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. NeRFusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5449–5458, 2022. 2

[42] Jianhao Zheng, Daniel Barath, Marc Pollefeys, and Iro Armeni. Map-adapt: real-time quality-adaptive semantic 3d maps. In *European Conference on Computer Vision*, pages 220–237. Springer, 2025. 2

[43] Zi-Xin Zou, Shi-Sheng Huang, Yan-Pei Cao, Tai-Jiang Mu, Ying Shan, Hongbo Fu, and Song-Hai Zhang. GP-Recon: Online monocular neural 3D reconstruction with geometric prior. *IEEE Transactions on Visualization & Computer Graphics*, (01):1–16, 2024. 2