Cross-Dimensional Refined Learning for Real-Time 3D Visual Perception from Monocular Video

Anonymous ICCV submission

Paper ID 4

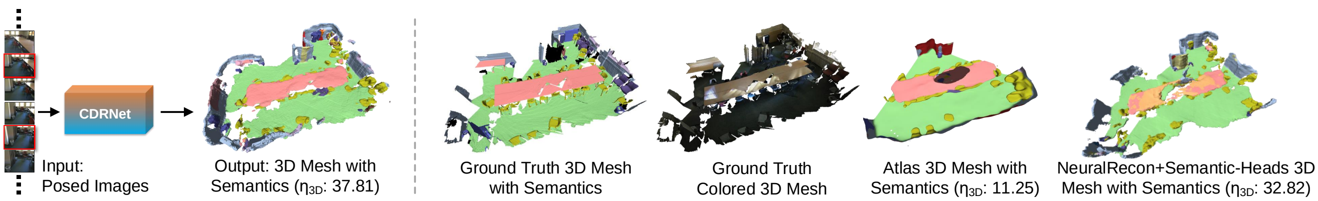


Figure 1. **Comparison between the proposed approach and baselines.** Our model is more accurate and coherent in real time, compared to two baseline methods with input from monocular video, Atlas [29] and NeuralRecon [41] + Semantic-Heads. Real-time 3D perception efficiency η_{3D} the higher the better. Color denotes different semantic segmentation labeling.

Abstract

We present a novel real-time capable learning method that jointly perceives a 3D scene's geometry structure and semantic labels. Recent approaches to real-time 3D scene reconstruction mostly adopt a volumetric scheme, where a Truncated Signed Distance Function (TSDF) is directly regressed. However, these volumetric approaches tend to focus on the global coherence of their reconstructions, which leads to a lack of local geometric detail. To overcome this issue, we propose to leverage the latent geometric prior knowledge in 2D image features by explicit depth prediction and anchored feature generation, to refine the occupancy learning in TSDF volume. Besides, we find that this cross-dimensional feature refinement methodology can also be adopted for the semantic segmentation task by utilizing semantic priors. Hence, we proposed an end-to-end cross-dimensional refinement neural network (CDRNet) to extract both 3D mesh and 3D semantic labeling in real time. The experiment results show that this method achieves a state-of-the-art 3D perception efficiency on multiple datasets, which indicates the great potential of our method for industrial applications.

1. Introduction

Recovering 3D geometry and semantics of objects or environment scenes prevails these days with the advent of the ubiquitous digitization. The digitization of the world where people live can not only help them better understand their environment scenes, but also enable robots to comprehend what they need to know about the world and therewith conducting assigned tasks. Generally, with surrounding environment measurements as input, 3D reconstruction and 3D semantic segmentation are two key 3D perception techniques [9, 42, 13] in the computer vision society, which enable a wide range of applications, including digital twins [19, 3], virtual/augmented reality (VR/AR) [41, 49], building information modeling [27, 44], and autonomous driving [4, 22].

Tremendous research efforts have been made for 3D perception techniques. Based on the sensor types, researches on 3D perception can be divided into two main streams, namely active range sensors that capture surface geometry information and RGB cameras that capture texture with perspective projection. Originated from KinectFusion [31], the commodity RGB-D range sensor is used to measure depth data first and then fuse it into Truncated Signed Distance Function (TSDF) volume for 3D reconstruction. Although the follow-up depth-based TSDF fusion methods [47, 48, 1, 50, 39] achieve detailed dense reconstruction result, they suffer from global incoherence due to the lack of sequential correlation, the tendency of noise disturbance due to redundant overlapped calculations, and the incapability of semantic deduction due to the lack of texture features.

On the other hand, as camera-equipped smartphones become readily available with built-in inertial measurement units, recent advances have emerged to explore 3D perception with RGB cameras on mobile devices. The problem of reconstructing 3D geometry with posed RGB images input only is referred to as multi-view stereo (MVS). Existing methods for MVS that are based on deep learning, tend to adopt a volumetric scheme by directly regressing the TSDF volume [29, 40, 7, 41] either as a whole or in fragments. However, these volumetric learning methods extract 3D geometric feature representation simply from the back projection of 2D image features, resulting in the mismatch to the 2D information priors for the predicted 3D reconstruction. Moreover, the intrinsic end-to-end learning manner and the lack of local details on the reconstructed mesh of these volumetric schemes result in an inferior semantic deduction based on its 3D reconstruction prediction.

What's worse, these learning-based methods tend to store their entire computational graphs in memory for aggregation and require prohibitive 3D convolution operations [29, 35, 40], which keeps them from being deployed on robots due to the real-time and low-latency requirements in SLAM. These limitations motivate our key idea to utilize 2D explicit predictions to further impose a light-weight feature refinement on the 3D features input in a sparse manner, while keeping the global coherence within the fragments. Unlike these preceding learning-based volumetric works, we conjecture that the utilization of 2D prior knowledge coming out of explicit predictions as a latent feature refinement plays a significant role in learning the feature representation in 3D perception. In addition, the feature refinement brought by 2D explicit prediction can be operated within the fragment input for keeping the computation redundancy and thus overhead low, while having the global coherence by correlating different fragments to extract the target 3D semantic mesh.

In this paper, we propose a novel framework, *CDRNet*, to accomplish both 3D meshing and 3D semantic labeling tasks in real-time. Our key contributions are as follows.

- We propose a novel, end-to-end trainable network architecture, which cross-dimensionally refines the 3D features with the prior knowledge extracted from the explicit estimations of depths and 2D semantics.
- The proposed cross-dimensional refinements yield more accurate and robust 3D reconstruction and semantic segmentation results. We highlight that the explicit estimations of both depths and 2D semantics serve as efficient yet effective prior knowledge for 3D perception learning.
- To achieve real-time 3D perception capability, our approach performs both geometric and semantic localized updates to the global map. We present a progressive 3D perception system that is capable of real-time interaction with input data streaming from cellphones with a monocular camera.

2. Related Work

Real-Time 3D Perception. The prosperity of deep learning hardwares enables both inference and training at the edge [20, 14], thus it consolidates the foundation to deploy more and more learning-based 3D perception techniques in real time. KinectFusion [31] first brought in the concept of handling 3D reconstruction tasks in real time with commodity RGB-D sensors. Han et al. [13] presented a real-time 3D meshing and semantic labeling system similar to our work, however, depth measurements from RGB-D sensors are required as input in their work. Pham et al. [33] built up 3D meshes with voxel hashing, and then fuse the initial semantic labeling with super-voxel clustering and a high-order conditional random field (CRF) to improve labeling coherence. Menini et al. [28] extended RoutedFusion [47] by merging semantic estimation in its TSDF extraction scheme for each incoming depth-semantics pair. NeuralRecon [41] adopted sparse 3D convolutions and the gated recurrent unit (GRU) to achieve a real-time 3D reconstruction on cellphones, without the capability of semantic deduction. For depth estimation and semantic segmentation, there are also works achieving real-time processing capability [46, 30, 33].

Voxelized 3D Semantic Segmentation. The learning of semantic segmentation on the voxelized map started from [5], which extends TSDF fusion pipeline [31] with per-pixel labels. 3DMV [11] and MVPNet [18] further combined both depth and RGB modalities to train an end-to-end network with 3D semantics for voxels and point clouds, respectively. PanopticFusion [30] performed map regularization based on adopting a CRF on the predicted panoptic labels. Atlas [29] utilized its extracted 3D features and passed them to a set of semantic heads for voxel labeling, the pyramid features are proven to have strong semantics at all scales than the gradient pyramid in nature, as proven in [23]. BPNet [15] proposed to have a joint-2D-3D reasoning in an end-to-end learning manner. Two derivative works [28, 17] of RoutedFusion incorporated semantic priors into their depth fusion scheme and removed their routing module for less overhead. However, none of these works utilize the prior knowledge within the estimated 2D semantics as a 3D feature refinement.

Volumetric 3D Surface Reconstruction. Volumetric TSDF fusion became prevalent for 3D surface reconstruction starting from the seminal work KinectFusion [31] due to its high accuracy and low latency. A follow-up work, PSDF-Fusion [32] augmented TSDF with a random variable to improve its robustness to sensor noise. Starting from DeepSDF [32], the learned representations of TSDF using depth input dominates the current fad. These learning-based substitutes [47, 48, 3, 1, 50, 39, 52] to TSDF fusion achieve impressive 3D reconstruction quality compared to the base-

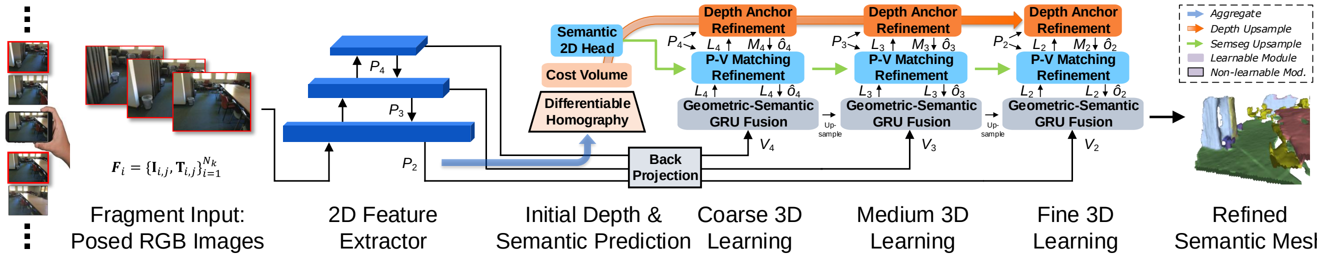


Figure 2. **Overview of CDRNet.** Posed RGB images from monocular videos are wrapped as fragment input for 2D feature extraction, which is used for both depth and 2D semantic predictions for cross-dimensional refinement purposes. To learn the foundational 3D geometry before conducting refinements, the extracted 2D features are back-projected into raw 3D features, $\mathcal{V}_s$, in different resolutions without any 2D priors involved. At each resolution, after being processed by the GRU, the output feature L_s in the local volume is further fed into Depth and Semantics refinement modules sequentially to have a 2D-prior-refined feature with better representations.

line method with the availability of RGB-D range sensors.

Given the fact that range sensors have relatively higher cost and energy consumption than RGB cameras, MonoFusion [34] is one of the first works to learn TSDF volume from RGB images by fusing the estimated depth into an implicit model. Atlas [29] started the trend of learning-based methods by a direct regression on TSDF volume. NeuralRecon [41] achieved a real-time 3D reconstruction learning capability by utilizing sparse 3D convolutions and recurrent networks with key frames as input. TransformerFusion [2] and VoRTX [40] introduced transformers [45] to improve the performance by more relevant inter-frame correlation. These learning-based methods prevail thanks to the availability of these general 2D feature extractors, such as FPN [23] and U-Net [37]. 2D information in RGB images can be effectively extracted and further utilized for constructing their 3D perception counterparts.

However, the learning of the explicit representations of 2D latent geometric features, such as depths and semantics, is **typically ignored** by all the prior arts. They only treat the 2D feature as an intermediate in the network and then conduct ray back-projection upon it, without considering the explicit representations for their 3D embodiment, which we found are significant prior knowledge for 3D perception. To extract depth as the explicit 2D representation, VolumeFusion [7] and SimpleRecon [38] performed local MVS and further fused it into TSDF volume with its customized network, while 3DVNet [35] performed sparse 3D convolutions on the feature-back-projected point cloud. Different from above, our method extracts the 2D representations from light-weight network modules, including a portion of MVSNet [51] for depth and a simple 2D MLP head for 2D semantics, to conduct the 3D feature refinements. Similar to [21, 24], we utilize the geometric and semantic prior information to improve the generalizability of our network by correlating the 2D representations and their 3D counterparts.

To the best of our knowledge, we present the very first learning-based method which uses posed RGB images input only to conduct 3D perception tasks in real time, including 3D meshing and semantic labeling.

3. Methods

Given a posed image sequence $\mathbf{I}$, our goal is to extract a 3D mesh model that can represent both **3D geometry** and **3D semantic labeling**, i.e., 3D meshing with vertices $\mathcal{K} \in \mathbb{R}^3$, surfaces $\mathcal{G} \in \mathbb{R}^3$, and its corresponding 3D semantic labeling $\mathcal{S} \in \mathbb{N}$. We achieve this goal by jointly predicting TSDF value $T \in [-1, 1]$ and semantic label $S \in \mathbb{N}$ for each voxel, and then extracting the mesh with the marching cubes [25]. Meanwhile, our proposed method aims at establishing a real-time capable deep learning model for these two 3D perception tasks. To quantitatively evaluate the efficiency of conducting these two tasks simultaneously, we define a 3D perception efficiency metric η_{3D} by involving frames per second (FPS) in runtime, as shown in Sec. 4.1.

The proposed network architecture is illustrated in Fig. 2. In Sec. 3.1, we introduce the joint fragment learning on depth, 2D semantic category, intermediate TSDF, and occupancy using key frames input, for the following cross-dimensional refinements of TSDF and 3D semantics. For each fragment, the geometric features are progressively extracted in a coarse-to-fine hierarchy using binomial inputs GRU to build the learned representations of 3D. Sec. 3.2 describes the cross-dimensional refinements for 3D features that refines 3D features with anchored features and semantic pixel-to-vertex correspondences enabled by the depth and 2D semantic predictions, which helps the learning of not only the TSDF value, but also the 3D semantic labeling in a sparsified manner. We also present the implementation details including loss design in Sec. 3.3. Specifications of the network are elaborated in the supplement.

3.1. Sparse Joint Fragment Learning in a Coarse-to-Fine Manner

Given the inherent nature of great sparsity in the ordinary real-world 3D scene, we utilize sparse 3D convolutions to efficiently extract the 3D feature from each input scene. However, the memory overhead of processing a 3D scene is still prohibitive, thus we fragment the whole 3D scene and progressively handle each of them, to further release the memory burden of holding up the huge 3D volume data.

Inspired by [29, 12, 41, 40, 35], we adopt a coarse-to-fine learning paradigm for the sparse 3D convolutions to effectively exploit the representation of 3D features in multiple scales. In each stage of the hierarchy, the raw features in the fragment bounding volume (FBV) is extracted from a GRU by correlating local features and global feature volume.

FBV Construction by Image Features. The input sequence of a monocular RGB video is first processed into frames. Following [46, 41], we select a set of key frames out of the input sequences $\mathbf{I}$ by querying on each frame's pose, namely the relative translation and optical center rotation with empirical thresholds, θ_{key} and t_{key}. Key frames are further wrapped into a fragment $\mathbf{F}_i = \{\mathbf{I}_{i,j}, \mathbf{T}_{i,j}\}_{j=1}^{N_k}$ as the input to the network, where i, j, and N_k denote the fragment index, the key frame index, and the number of key frames in the fragment, respectively.

Once the fragment $\mathbf{F}_i$ is constructed, it is processed by a 2D image backbone network to extract image features. In the decoder part of the backbone network, three different resolutions of feature maps are extracted from the pyramid sequentially as $\mathcal{P}_s \in \{P_2, P_3, P_4\}$, where the suffix notation of P denotes the scaling ratio level in $\log_2$ similar to [23]. The extracted feature $\mathcal{P}_s$ is then back-projected into a local 3D volume, according to the projection matrix of each frame in $\mathbf{F}_i$. We hereby define FBV as the current local volume $\mathcal{F}_{s,i} = \{T_{s,i}^{x\times y\times z}, S_{s,i}^{x\times y\times z}\}$ that is conditioned on the pyramid layers $\mathcal{P}_s$, where all the 3D voxels that are casted in the view frustums of current $\mathbf{F}_i$ are included.

Initial Depth and 2D Semantics Learning. With the fine feature P_2 as input, we build up differentiable homography fronto-parallel planes for the coarse-level depth prediction $\hat{D}_4$. Likewise, 2D semantics prediction $\hat{S}_4^{2D}$ is extracted with a pointwise convolutional decoder as the 2D semantic head using P_2. The resolution gap between the input and output feature map provides generalizability. The initial depth estimation is retrieved from the features using a lightweight multi-view stereo network via plane sweep [51]. For each source feature map x in P_2, we conduct the planar transformation $\mathbf{x}_j \sim \mathbf{H}_j(d) \cdot x$, where "$\sim$" denotes the projective equality and $\mathbf{H}_j(d)$ is the homography of the j^{th} key frame at depth d. For a given fragment input $\mathbf{F}_i$, the homography[1] is defined as:

$$\mathbf{H}_j(d) = d \cdot \mathbf{K}_j \cdot (\mathbf{T}_j \cdot \mathbf{T}_1^{-1}) \cdot \mathbf{K}_1^T \ , \qquad (1)$$

where $\mathbf{T} \in SE(3)$ denotes the transform matrix inversed from the pose. To measure the similarity after conducting homography warping, we calculate the variance cost of $\mathbf{x}_j$ and further process it with an encoder-decoder-based cost regularization network. The output logit from the regularization network is treated as the depth probability on each hypothesis plane and we conduct the same *soft argmin* in [51] to have the initial depth prediction.

[1]For brevity's sake, the transformation from homogeneous coordinates to Euclidean coordinates in the camera projection is omitted here.

Geometric and Semantic GRU Fusion. Meanwhile, as the 2D features are extracted in different resolutions, they are back-projected from each of the pyramid level in $\mathcal{P}_s$ into raw geometric 3D features $\mathcal{V}_s \in \{V_2, V_3, V_4\}$, which are further sparsified by sparse 3D convolutions. To improve the global coherence and temporal consistency of the reconstructed 3D mesh, following [41], we first correlate the sparse geometric feature $\mathcal{V}_s$ in the current $\mathcal{F}_{s,i}$ using GRU, with the local FBV hidden states $H_{s,i-1}$ whose information coming from all of the previous fragments $\mathcal{F}_{s,i'}, i' < i$ and coordinates are masked to be the same as $\mathcal{V}_s$. Such correlation outputs a temporal-coherent local feature $L_{s,i}$ for each stage s, which will be used to generate dense occupancy intermediate $o_{s,i}$, and passed to the 2D-to-3D cross-dimensional refinements. The global feature volume for the entire scene $G_{s,i}$ will be fused by $G_{s,i-1}$ and $L_{s,i}$ given the coordinates of $\mathcal{V}_s$ as masks, and update $H_{s,i}$. Unlike [41], we reuse the same parameters in GRU to process the back-projected and upsampled 3D semantic features to generalize better for the semantic prediction $\hat{S}$ in the current FBV. This is because inputting TSDF and semantic features sequentially into GRU enables its selective fusion across modalities, thus the feature extracted from the hidden state incorporates more semantic information, as pointed out in [36].

For the sake of learning 3D features consistently between scales, we update $\mathcal{V}_s$ at each stage by fusing with the upsampled $L_{s+1,i}$. Inspired by the *meta data* mechanism proposed in [38], we further concatenate sparse features, with sparse TSDF, occupancy and semantics after masking with $o_{s,i}$, as the meta feature $L_{s+1,i}$ to be upsampled. We found the inclusion of semantic information in the hidden state of GRU helps build up a good starting point for the upcoming feature refinements, which is verified in the ablation.

3.2. 2D-to-3D Cross-Dimensional Refinements

The raw coherent features from GRUs lack detailed geometric descriptions, leading to unsatisfactory meshing and semantic labeling results. To overcome these issues, we propose to leverage the 2D feature that is latent after incorporating the learning of depth and semantic frame for the refinement purposes. We notice that with the learning of depth and 2D semantics, the 2D features now reside in the latent space which can generalize to more accurate 3D geometry and semantics via cross-dimensional refinements.

2D-to-3D Prior Knowledge. Consider a probabilistic prior in the latent space of the output coherent feature coming from GRU, which accounts for the prior knowledge that the pixel information in both depth predictions and 2D semantic predictions should produce high confidence matching with regard to their own 3D representations. The prior condi-

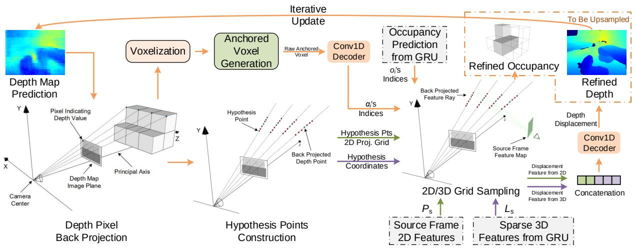


Figure 3. **Workflow of the depth anchor refinement module.** Anchored voxels are extracted from depth points and further serve as a geometric prior for the occupancy refinement.

tioned 3D feature for both perception tasks is defined as:

$$X_{prior} = f(L_{s,i}) = f\big(H_{s,i}(\mathcal{V}_s, H_{s,i-1} \mid \mathcal{F}_{s,i})\big) \quad , \tag{2}$$

where $f(\cdot)$ is the 2D-to-3D feature refinement process for either 3D meshing or 3D semantic labeling, whose input is $L_{s,i}$ extracted from $\mathcal{V}_s$ and $H_{s,i-1}$ given $\mathcal{F}_{s,i}$. We borrow the notation of $H_{s,i}$ to be a constructor function $H_{s,i}(\cdot)$ indicating GRU. For each voxel in $\mathcal{F}_{s,i}$, both TSDF and semantic labeling predictions can be formulated as:

$$\hat{I}_{s,i} = \epsilon h\big(H_{s,i}(\mathcal{V}_s, H_{s,i-1} \mid \mathcal{F}_{s,i})\big) + (1-\epsilon)X_{prior} \quad , \tag{3}$$

where $\hat{I}_{s,i} \in \mathcal{F}_{s,i}$ is the refined prediction; ϵ is a random variable for the respective prior, which is jointly learned by the feature refinement modules representing the 2D-to-3D priors and the GRU network trained with maximum likelihood estimation losses; $h(\cdot)$ is the prediction head. The proof of Eq. (3) can be found in the supplement.

The key insight is that the voxels back-projected from either depth prediction or semantic label prediction of the input images has strong evidence on its 3D counterparts. We hereby define anchored voxels α_i, as those voxels in $\mathcal{F}_{s,i}$ that are incorporating all the back-projected depth points, given the fact that the 3D reconstruction task is essentially an inverse problem. We propose two progressive feature refinement modules to learn the high confidence of the refined features in latent space such that a more accurate $\hat{I}_{s,i}$ can be extracted with the help of 2D-to-3D prior knowledge.

Depth-Anchored Occupancy Refinement. Unlike the volumetric methods [29, 41] that directly regress on the TSDF volume, we propose a novel module in each stage s that can explicitly refine the initial depth, predict depths in resolutions, and further create the 3D anchored features with the depth prediction, as shown in Fig. 3. The anchored feature is generated by 3D sparse convolutions with an anchored voxel on the occupancy intermediate o_i [2].

Intuitively, the anchored voxel will have higher confidence of achieving a valid o_i and $T_{s,i}$ close to zero. We imposed the anchored feature on occupancy feature to reinforce the occupancy information brought by the depth prior.

[2] The universal stage suffix s is hereinafter omitted for brevity.

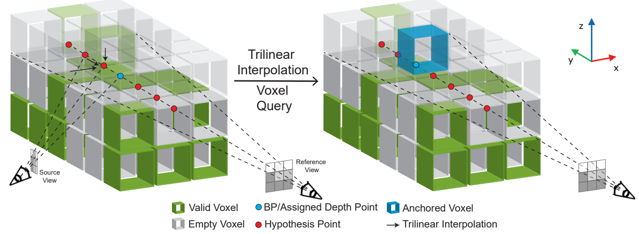


Figure 4. **Anchored voxel generation for occupancy refinement.** An example of occupancy refinement happening on the middle row of a $3 \times 6 \times 3$ FBV is shown with geometrically valid voxel highlighted in green. The initial depth prediction is back-projected into FBV and displaced by trilinear interpolation on all depth points, in the range of 6 additional hypothesis points for each depth point. The voxels on the top are set as half transparent for clarity.

Inspired by [6, 35], we conduct PointFlow algorithm for each stage in the coarse-to-fine structure $\mathcal{V}_s$ to determine the depth displacement on the initial depth prediction such that finer depth prediction can be achieved. Different from the PointFlow algorithm used in [35], we utilize the back-projected depth points from all N_k views in the fragment to query an anchored voxel, which can be further aggregated with o_i. Fig. 4 illustrates how these hypothesis points are selected and turned into depth displacement prediction, such that the anchored voxel can be generated. The anchored voxel index in the 3D volume is sparsified as a mask to update the occupancy prediction as $\hat{o}_i$ in the following:

$$\hat{o}_i = o_i \cap \alpha_i \quad . \tag{4}$$

The enhanced occupancy prediction $\hat{o}_i$ is used to condition the TSDF volume at the current stage to generate the refined $\hat{T}_i$, which is further sparsified with a light-weight pointwise convolution and upsampled to concatenate with $L_{s,i}$.

Pixel-to-Vertex Matching Semantic Refinement. In addition to the depth anchor refinement, we propose a semantic cross-dimensional refinement which utilizes the semantic prior that lies in the 2D semantic prediction to have a refined 3D voxel semantic prediction, implemented as follows. First, the 2D feature extractor pyramid will learn the 2D semantic prior information that is useful for 3D voxel semantic labeling learning by incorporating the learning of 2D frame semantic labeling. Second, the sparse 3D feature $L_{s,i}$ will be passed to pointwise 3D convolution layers and come up with the initial 3D voxel semantic labeling predictions in respective scales. Third, to conduct the semantic feature refinement, we observed that there is a sole 3D voxel counterpart in $\mathcal{F}_{s,i}$ for each pixel on a 2D semantic prediction of $\mathbf{I}_{i,j}$, since the surface edges are encoded as vertices. We define these vertices as the one-on-one matching correspondences to their camera-projected pixels, which is recorded in a matching matrix for masking the 2D features $\mathcal{P}_s$.

The upper part of Fig. 5 illustrates the design of the matching matrix that is used to correlate the pixel-vertex pairs for each frame $\mathbf{I}_{i,j}$ across all vertices in $\mathcal{F}_{s,i}$. We construct the matching matrix $\mathbf{M} = \{\overrightarrow{m}_{idx}\}_{idx=1}^{N}$ for each semantic labeling frame, where N is the number of the vertices in the volume $\mathcal{F}_{s,i}$. Each column of the matching matrix $\mathbf{M}$ is defined as:

$$\mathbf{M}(idx) = \overrightarrow{m}_{idx} = \begin{bmatrix} u_{idx} \\ v_{idx} \\ \text{mask} \end{bmatrix} . \tag{5}$$

For each column, each pixel-vertex pair recorded in the matching matrix, i.e., the idx^{th} vertex in the 3D volume on the right-hand side of the upper part and its correspondence pixel on the left-hand side is recorded. The last entry of the pixel-vertex pair represents a mask which will be recorded as valid when the 2D correspondence for $\mathbf{M}$ is in the current view frustum of the frame.

After the matching matrix $\mathbf{M}$ is constructed, it will be used for masking each of the feature map $\mathcal{P}_s$ with the $\log_2$ scale of s to create a refined feature, whose voxel number is the same as the number of sparse 3D features, as shown in the lower part of Fig. 5. Meanwhile, the coordinates of the sparse 3D features $L_{s,i}$ will be mapped as the coordinate of the refined feature. By doing so, the underlying semantic information from the $\mathcal{P}_s$ can be incorporated by $L_{s,i}$, such that better 3D semantic prediction can be achieved. Then we use the sparse pointwise convolution to extract its underlined feature from 2D semantics, and concatenate it with $L_{s,i}$ to create $L_{s-1,i}$ with semantic information for the refinement in the next finer stage, so as to ensure the 2D semantic priors to have reliable refinement on the sparse coherent features.

3.3. Implementation Details

Our model is implemented in PyTorch, trained and tested on an NVIDIA RTX3090 graphics card. We empirically set the optimizer as Adam without weight decay [26], with an initial learning rate as 0.001, which goes through 3 halves throughout the training. The first momentum and second momentum are set to 0.9 and 0.999, respectively. For key frame selection, following [46, 41], we set thresholds θ_{key}, t_{key} and fragment input number N_k as 15 degrees, 0.1 meters, and 9, respectively. A fraction of FPN [23] is adopted as the 2D backbone with its classifier as MNasNet [43]. MinkowskiEngine [8] is utilized as the sparse 3D tensor library. More details are introduced in the supplement.

Loss Design. Our model is trained in an end-to-end fashion. Since our target is to learn the 3D geometry and semantic segmentation of the surrounding scene given posed images input, we regress the TSDF value with the mean absolute error (MAE) loss, classify the occupancy value with the binary cross-entropy (BCE) loss and the semantic label-

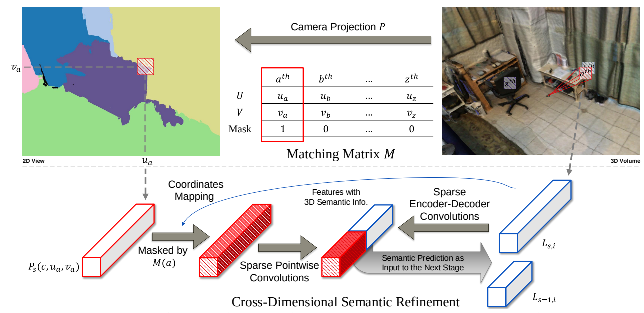


Figure 5. **Workflow of the pixel-to-vertex matching feature refinement.** *Upper:* Matching matrix $\mathbf{M}$ for pixel-to-vertex correspondence is constructed with camera projection. The red-shaded boxes in the 3D volume denote an example of valid correspondence pairs of the 2D semantic prediction $\overrightarrow{m}_a$ and its surrounding 3D scene. The green and purple boxes denote the occluded vertex and out-of-view vertex that is not imaged in the 2D semantic prediction, which correspond to $\overrightarrow{m}_b$ and $\overrightarrow{m}_z$, respectively; *Lower:* The 2D features are further masked by $\mathbf{M}(a)$ with the mapped coordinates from the sparse 3D features of the scene that are valid for the current view.

ing with cross-entropy (CE) loss as:

$$\begin{aligned}\mathcal{L}_{3D} = &\sum_{s=2}^{4} \alpha_s \mathcal{L}_{\text{MAE}}(T_s, \hat{T}_s) + \lambda \alpha_s \mathcal{L}_{\text{BCE}}(O_s, \hat{O}_s) \\ &+ \beta_s \mathcal{L}_{\text{CE}}(S_s, \hat{S}_s) \ ,\end{aligned} \tag{6}$$

where T, S, and O denote TSDF value, semantic labeling, and occupancy predictions. α_s, β_s, and λ are the weighting coefficients in different stages for TSDF volume, semantic volume and positive weight for BCE loss, respectively. By doing so, the learning process stays most sensitive and relevant to the supervisory signals in the coarse stage, and less fluctuating as the 3D features become finer with the upsampling, after log-transforming the predicted and ground-truth TSDF value following [29].

To conduct cross-dimensional refinements, we regress the depth estimation with MAE loss and classify the 2D semantic segmentation with CE loss:

$$\begin{aligned}\mathcal{L}_{2D} = &\mathcal{L}_{\text{MAE}}(d_{init}, \hat{D}_{init}) + \mathcal{L}_{\text{CE}}(S_2^{2D}, \hat{S}_2^{2D}) \\ &+ \sum_{s=2}^{4} \gamma_s \mathcal{L}_{\text{MAE}}(D_s, \hat{D}_s) \ ,\end{aligned} \tag{7}$$

where D and γ_s denote depth and the weighting coefficient for depth estimation in different stages. We further wrap the losses into an overall loss $\mathcal{L} = \mathcal{L}_{3D} + \mu \mathcal{L}_{2D}$, where μ is the coefficient to balance the joint learning of 2D and 3D.

4. Experiments

4.1. Datasets and Metrics

We conduct the experiments on two indoor scene datasets, ScanNet (v2) [10] and SceneNN [16]. The model

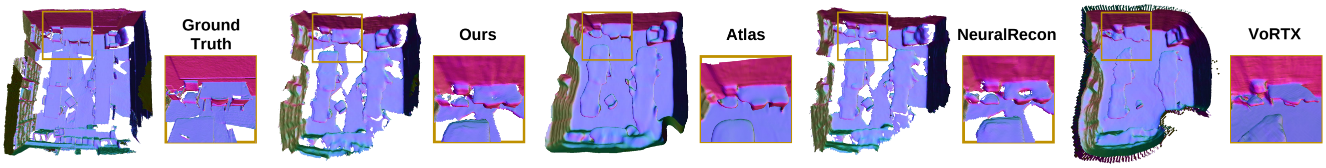


Figure 6. **Qualitative 3D reconstruction results on ScanNet.** Our method is capable of reconstructing consistent and detailed geometry which is neither overly smooth as the one from Atlas [29] nor eroded with holes as from NeuralRecon [41].

Method	Acc. ↓	Comp. ↓	Prec. ↑	Recall ↑	F-Score ↑
Atlas [29]	0.124	0.074	0.382	**0.711**	0.499
NeuralRecon [41]	0.073	0.106	0.450	0.609	0.516
3DVNet [35]	**0.051**	0.075	**0.715**	0.625	0.665
SimpleRecon [38]	0.061	**0.055**	0.686	0.658	**0.671**
VoRTX [40]	0.891	0.092	0.618	0.589	0.623
Ours	0.068	0.062	0.609	0.616	0.612

Table 1. **Quantitative 3D reconstruction results on ScanNet.** Our method is superior to two main baselines, Atlas and NeuralRecon, and as competitive as other SOTAs on 3D reconstruction.

Method	FPS ↑	KFPS ↑	FLOPF ↓	mIoU ↑	η_{3D} ↑
3DMV [11]	7.04	N/A	65.06G	44.2	N/A
BPNet [15]	4.46	N/A	141.06G	74.9	N/A
Atlas [29]	66.3	N/A	267.04G	34.0	11.25
NeuralRecon [41] + Semantics-Heads	**228**	**30.9**	**42.38G**	27.9	32.82
VoRTX [40] + Semantic-Heads	119	13.5	150.23G	13.2	9.79
Ours	158	21.4	90.62G	**39.1**	**37.81**

Table 2. **Quantitative 3D voxel semantic segmentation and overall 3D perception results on ScanNet.** *Upper:* Two representative state-of-the-art methods for semantic segmentation whose input requires either depth or 3D mesh, respectively. No key-frame selection and F-score are involved due to their input modality; *Lower:* RGB-input-only volumetric methods. Key-frame FPS (KFPS) is measured with the same selection scheme across all methods. FLOPF is measured with PyTorch operation counter across operations of neural network's learnable modules.

is trained on the ScanNet train set, tested and reported on the ScanNet test set and further verified on SceneNN data set. To quantify the 3D reconstruction and 3D semantic segmentation capability of our method, we use the standard metrics following [29, 41]. Completeness Distance (Comp.), Accuracy Distance (Acc.), Precision, Recall, and F-score, are used for 3D reconstruction, while mean Intersection over Union (mIoU) is used for 3D semantic segmentation.

To evaluate how much robustness a model can achieve while targeting solving 3D perception tasks in real time, we define the 3D perception efficiency metric η_{3D} = FPS × mIoU × F-score, since F-score is regarded as the most suitable 3D metric for evaluating 3D reconstruction quality by considering Precision and Recall at the same time [29, 41, 38]. It is noteworthy that for fairness across methods, FPS for processing speed is measured in the inference across all captured frames in a given video sequence rather than key frames only, since the input is the same for different methods regardless of their key frame selection scheme.

4.2. Evaluation Results and Discussion

3D Perception. To evaluate the 3D perception capability, we mainly compare our methods against state-of-the-art works in two categories: volumetric 3D reconstruction and voxelized 3D semantic segmentation methods.

For 3D reconstruction capability, we compare our proposed method with the canonical volumetric methods [29, 41] and several state-of-the-art 3D reconstruction methods with posed images input [35, 38]. Fig. 6 demonstrates the superiority of our method in terms of 3D reconstruction by showing the 3D meshing results in normal mapping. Table 1 shows that our method outperforms two main baseline methods in terms of 3D meshing accuracy. We further compare both state-of-the-art depth estimation methods and volumetric methods in depth metrics in the supplement to justify from the depth extraction perspective.

For 3D semantic segmentation quality, we compare Atlas, NeuralRecon with semantic heads, and VoRTX with semantic heads with our methods in Table 2. We augment three stages of MLP heads on top of the flattened 3D features to predict the semantic segmentation for both baselines. One of the SOTA baselines, SimpleRecon is intrinsically unable to follow this modification for semantics due to the lack of 3D feature extraction. Table 2 shows that our method outperforms these two baselines. Besides mIoU for semantic segmentation, we include FPS and η_{3D} for 3D perception efficiency in the comparison. We also include two state-of-the-art 3D semantic segmentation methods, 3DMV [11] and BP-Net [15]. It shows that our method can achieve mIoU results nearly comparable to 3DMV, but with only RGB images as input. Overall, our method achieves the best 3D semantic segmentation performance and highest 3D perception efficiency among all the volumetric methods. Fig. 7 and Fig. 8 illustrate the 3D semantic labeling results. We found that the semantic information generation on VoRTX is unsatisfying, mostly caused by its bias on geometric features brought by the projective occupancy mentioned in [40].

Efficiency. Since our main goal is to achieve real-time processing performance while solving 3D perception tasks, we compare the computational efficiency of our model against other RGB-input-only volumetric methods in Table 2. The 3D perception efficiency metric η_{3D} for several 3D semantic segmentation works are shown there. We employ FPS, which is commonly used to measure efficiency for 2D-input 3D perception methods [29, 41, 40], as a metric to bring out and emphasize the nature of real-time system. We also include the floating-point operations per frame (FLOPF) to compare the learnable parameters' operations across differ-

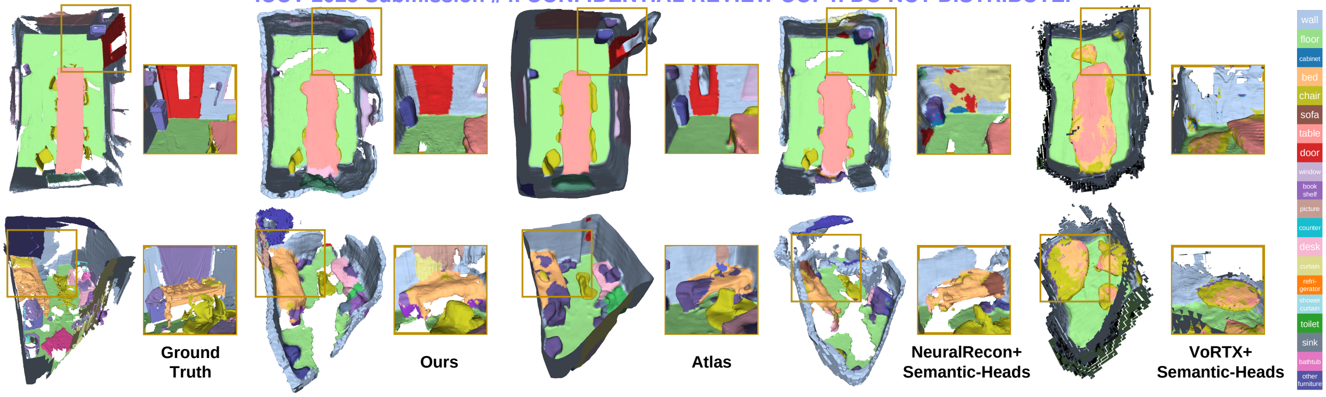
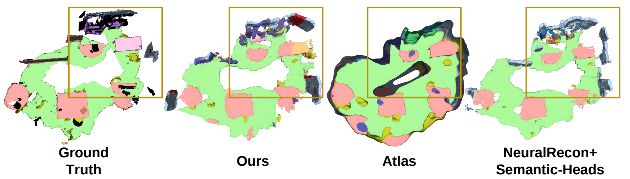


Figure 7. **Qualitative 3D semantic segmentation results on ScanNet.** Our method consistently outperforms baseline models and sometimes even surpasses the ground-truth labeling, e.g., in the bottom row, the photo-printed curtain above the bed is correctly recognized as "curtain" and "picture", whereas the ground truth mistakes it as "other furniture".



Method	Acc. ↓	Comp. ↓	FPS ↑	F-Score ↑	mIoU ↑	η_{3D} ↑
Atlas [29]	0.074	0.164	54.7	0.499	31.4	8.57
NeuralRecon [41] + Semantic-Heads	0.138	0.216	**178**	0.510	15.9	14.43
Ours	**0.068**	**0.143**	121	**0.611**	**36.7**	**27.09**

Figure 8. **Qualitative and quantitative 3D pereception results on SceneNN dataset.** Our method is proven to be generalized to SceneNN without pretraining on the SceneNN train set.

ent methods. The superiority in η_{3D} of our method manifests that it has better deployment potential for real-life 3D perception applications. From the human user's and robotic SLAM's points of view, our method greatly surpasses the threshold of being real-time, 90.17 FPS, as elaborated in the supplement. It shows that our method is more suitable for real-time industrial scenarios with input data from low-cost portable devices compared to baseline methods.

4.3. Ablation Study

To analyze the effectiveness of cross-dimensional refinement, we present 3D perception efficiency η_{3D} and its components of with different modifications in Table 3. In other experiments above, we adopt (e) as our method.

Binomial GRU Fusion. In (a), we remove the back-projected semantics input to GRU in the pipeline. Compared with (e), both F-score and mIoU of the removal degrade since no hidden semantic information from last FBV is fused with GRU anymore. Although FPS increases due to less computations, the efficiency η_{3D} is worse.

Depth Refinement. In (c), we remove the depth anchored refinement in the pipeline. The loss in F-score and mIoU manifests that the geometric feature without depth anchored refinement becomes inferior, which means depth anchored refinement can improve 3D reconstruction performance.

Semantic Refinement. We validate the semantic refinement in the pipeline by removing this module and, as shown in (d). The mIoU drops due to the insufficient learning information from semantics heads only. This result demonstrates the effectiveness of our semantic refinement scheme based on pixel-to-vertex matching for improving 3D semantic segmentation performance. We also experiment with no refinements but depth and 2D semantics learning setup in (b), which gives the highest FPS but not satisfying 3D perception performance.

	GRU Input	Depth		Semantics		F-Score↑	mIoU↑	FPS ↑	η_{3D} ↑
		DE	AR	SE	PVR				
a	Geo.	✓	✓	✓	✓	0.477	31.7	190	28.73
b	Geo.+ Sem.	✓		✓		0.479	27.1	**232**	30.12
c	Geo.+ Sem.	✓		✓	✓	0.482	34.5	169	28.10
d	Geo.+ Sem.	✓	✓	✓		0.556	26.8	226	33.68
e	Geo.+ Sem.	✓	✓	✓	✓	**0.612**	**39.1**	158	**37.81**

Table 3. **Ablation study.** We assess our method by removing each of the proposed feature fusion techniques on ScanNet. DE, AR, SE, and PVR denote depth estimation, anchored refinement, 2D semantics estimation, and point-to-vertex refinement, respectively.

5. Conclusion

In this paper, we proposed a lightweight volumetric method, *CDRNet*, that leverages the 2D latent information about depths and semantics as the feature refinement to handle 3D reconstruction and semantic segmentation tasks effectively. We demonstrated that our method has real-time 3D perception capabilities, and justified the significance of utilizing 2D prior knowledge when solving 3D perception tasks. Experiments on multiple datasets justify the 3D perception performance improvement of our method compared to prior arts. From the application point of view, the scalability of *CDRNet* supports the notion that 2D priors should not be disregarded in 3D perception tasks and opens up new avenues for achieving real-time 3D perception using input data from readily accessible portable devices such as smartphones and tablets.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *CVPR*, pages 6290–6301, 2022. 1, 2

[2] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. In *CVPR*, volume 34, pages 1403–1414, 2021. 3

[3] Aljaz Bozic, Pablo Palafox, Michael Zollhofer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *CVPR*, pages 1450–1459, 2021. 1, 2

[4] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, pages 3991–4001, 2022. 1

[5] Tommaso Cavallari and Luigi Di Stefano. Semanticfusion: Joint labeling, tracking and mapping. In *ECCV*, pages 648–664. Springer, 2016. 2

[6] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *ICCV*, pages 1538–1547, 2019. 5

[7] Jaesung Choe, Sunghoon Im, Francois Rameau, Minjun Kang, and In So Kweon. Volumefusion: Deep depth fusion for 3d scene reconstruction. In *ICCV*, pages 16086–16095, 2021. 2, 3

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 6

[9] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. In *NeurIPS*, volume 34, pages 8282–8293, 2021. 1

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 6

[11] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *NeurIPS*, pages 452–468, 2018. 2, 7

[12] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *ICCV*, pages 2495–2504, 2020. 4

[13] Lei Han, Tian Zheng, Yinheng Zhu, Lan Xu, and Lu Fang. Live semantic 3d perception for immersive augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):2012–2022, 2020. 1, 2

[14] Ziyang Hong and C. Patrick Yue. Efficient-grad: Efficient training deep convolutional neural networks on edge devices with gradient optimizations. In *ACM Transactions on Embedded Computing Systems (TECS)*, volume 21, pages 1–24. ACM New York, NY, 2022. 2

[15] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *CVPR*, pages 14373–14382, 2021. 2, 7

[16] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101. IEEE, 2016. 6

[17] Shi-Sheng Huang, Haoxiang Chen, Jiahui Huang, Hongbo Fu, and Shi-Min Hu. Real-time globally consistent 3d reconstruction with semantic priors. *IEEE Transactions on Visualization & Computer Graphics*, 01:1–1, 2021. 2

[18] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. In *ICCVW*, pages 0–0, 2019. 2

[19] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *CVPR*, pages 5616–5626, 2022. 1

[20] Yann LeCun. 1.1 deep learning hardware: Past, present, and future. In *2019 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 12–19. IEEE, 2019. 2

[21] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C van Gemert. Zero-shot day-night domain adaptation with a physics prior. In *ICCV*, pages 4399–4409, 2021. 3

[22] Chenguang Li, Jia Shi, Ya Wang, and Guangliang Cheng. Reconstruct from top view: A 3d lane detection approach based on geometry structure prior. In *CVPRW*, pages 4370–4379, 2022. 1

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2, 3, 4, 6

[24] Yancong Lin, Ruben Wiersma, Silvia L Pintea, Klaus Hildebrandt, Elmar Eisemann, and Jan C van Gemert. Deep vanishing point detection: Geometric priors make dataset variations vanish. In *CVPR*, pages 6103–6113, 2022. 3

[25] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 21(4):163–169, 1987. 3

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 6

[27] Jisan Mahmud, True Price, Akash Bapat, and Jan-Michael Frahm. Boundary-aware 3d building reconstruction from a single overhead image. In *CVPR*, pages 441–451, 2020. 1

[28] Davide Menini, Suryansh Kumar, Martin R Oswald, Erik Sandström, Cristian Sminchisescu, and Luc Van Gool. A real-time online learning framework for joint 3d reconstruction and semantic segmentation of indoor scenes. In *IEEE Robotics and Automation Letters*, volume 7, pages 1332–1339. IEEE, 2021. 2

[29] 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 *ECCV*, pages 414–431. Springer, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[30] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4205–4212. IEEE, 2019. 2

[31] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and

tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136. IEEE, 2011. 1, 2

[32] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 2

[33] Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Real-time progressive 3d semantic segmentation for indoor scenes. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1089–1098. IEEE, 2019. 2

[34] Vivek Pradeep, Christoph Rhemann, Shahram Izadi, Christopher Zach, Michael Bleyer, and Steven Bathiche. Monofusion: Real-time 3d reconstruction of small scenes with a single web camera. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 83–88. IEEE, 2013. 3

[35] 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, 3, 4, 5, 7

[36] Luis Roldao, Raoul De Charette, and Anne Verroust-Blondet. 3d semantic scene completion: A survey. *IJCV*, 130(8):1978–2005, 2022. 4

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 3

[38] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *ECCV*, 2022. 3, 4, 7

[39] Christiane Sommer, Lu Sang, David Schubert, and Daniel Cremers. Gradient-sdf: A semi-implicit surface representation for 3d reconstruction. In *CVPR*, pages 6280–6289, 2022. 1, 2

[40] 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, 3, 4, 7

[41] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *CVPR*, pages 15598–15607, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 1

[43] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, pages 2820–2828, 2019. 6

[44] Carlos A Vanegas, Daniel G Aliaga, and Bedrich Benes. Building reconstruction using manhattan-world grammars. In *CVPR*, pages 358–365, 2010. 1

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017. 3

[46] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International Conference on 3D Vision (3DV)*, pages 248–257. IEEE, 2018. 2, 4, 6

[47] Silvan Weder, Johannes Schonberger, Marc Pollefeys, and Martin R Oswald. Routedfusion: Learning real-time depth map fusion. In *CVPR*, pages 4887–4897, 2020. 1, 2

[48] Silvan Weder, Johannes L Schonberger, Marc Pollefeys, and Martin R Oswald. Neuralfusion: Online depth fusion in latent space. In *CVPR*, pages 3162–3172, 2021. 1, 2

[49] Yiming Xie, Matheus Gadelha, Fengting Yang, Xiaowei Zhou, and Huaizu Jiang. Planarrecon: Real-time 3d plane detection and reconstruction from posed monocular videos. In *CVPR*, pages 6219–6228, 2022. 1

[50] Yabin Xu, Liangliang Nan, Laishui Zhou, Jun Wang, and Charlie CL Wang. Hrbf-fusion: Accurate 3d reconstruction from rgb-d data using on-the-fly implicits. In *ACM TOG*, volume 41, pages 1–19. ACM New York, NY, 2022. 1, 2

[51] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783, 2018. 3, 4

[52] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022. 2