This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

Incremental Dense Reconstruction from Monocular Video with Guided Sparse Feature Volume Fusion

Xingxing Zuo[1,2], Nan Yang[1], Nathaniel Merrill[3], Binbin Xu[4,5,†], Stefan Leutenegger[1,2,4,6]

***Abstract*—Incrementally recovering 3D dense structures from monocular videos is of paramount importance since it enables various robotics and AR applications. Feature volumes have recently been shown to enable efficient and accurate incremental dense reconstruction without the need to first estimate depth, but they are not able to achieve as high of a resolution as depth-based methods due to the large memory consumption of high-resolution feature volumes. This letter proposes a real-time feature volume-based dense reconstruction method that predicts TSDF (Truncated Signed Distance Function) values from a novel *sparsified* deep feature volume, which is able to achieve higher resolutions than previous feature volume-based methods, and is favorable in outdoor large-scale scenarios where the majority of voxels are empty. An uncertainty-aware multi-view stereo (MVS) network is leveraged to infer initial voxel locations of the physical surface in a sparse feature volume. Then for refining the recovered 3D geometry, deep features are attentively aggregated from multi-view images at potential surface locations, and temporally fused. Besides achieving higher resolutions than before, our method is shown to produce more complete reconstructions with finer detail in many cases. Extensive evaluations on both public and self-collected datasets demonstrate a very competitive real-time reconstruction result for our method compared to state-of-the-art reconstruction methods in both indoor and outdoor settings.**

***Index Terms*—Monocular Dense Mapping, Neural Implicit Representation, Feature Volume Fusion**

I. Introduction

DENSE reconstruction from video images with deep neural networks has attracted significant attention in recent years. Deep feature volume-based 3D scene reconstruction, regressing scene geometry directly from volumetric feature volume, has shown promising results [1–4], and has the potential to enable a wide range of robotic applications. The incremental variant [2] can even achieve real-time performance on a desktop with commercial-level GPU. Compared to predicting dense depth at multiple views and then fusing depths into a global 3D map, feature volume-based method back-project encoded 2D image features into 3D voxel grids, and directly regress truncated signed distance function (TSDF) from accumulated features across multiple image views, by using a neural network composed of 3D convolutional layers and multiple layer perceptron (MLP) layers. Operation and prediction directly on the view-independent 3D feature volume have the advantage of capturing smoothness and 3D shape prior of the surface in the scene.

However, there are several drawbacks for existing feature volume-based methods [1–4]. Firstly, allocating features into all visible voxels along the whole rays cast from image pixels is cumbersome and redundant, which not only creates unnecessary confusion for network inference but also incurs excessive memory consumption and heavy computational burden. The ideal solution is only allocating image features into the relevant regions in 3D space, i.e., around the physical surface, for reconstruction. In the case that surface location is not certainly known, features can be allocated to the potential region where the surface is likely to locate. Secondly, due to the memory and computation issue for the volumetric dense feature volume, existing methods [1–4] are not capable of high-resolution reconstruction. All of them have demonstrated to ability to perform 3D dense reconstruction with feature volume, using a voxel size of over 4cm at the finest level. It is acceptable but causes visible aliasing artifacts such as surfaces appearing overly smooth and lacking fine details. Besides, the memory consumption of feature volume grows cubically with the increased volumetric resolution, which hinders existing methods from scaling up to high-resolution and fine-detailed reconstruction.

To address the aforementioned issues, we propose a novel guided sparse feature volume fusion method for real-time incremental scene reconstruction. Feature volumes are constructed fragment by fragment, and temporally fused into a global one. This incremental paradigm frequently updates the reconstructed 3D map, which favors real-time applications. To maintain the sparsity of the feature volume for efficient reconstruction, we propose a method to selectively allocate features into only relevant voxels around the actual physical surface, which aims to avoid excessive memory and computation consumption. We firstly leverage an efficient MVS network to predict dense depth and depth uncertainty, which is used to select the sparse set of voxels to be aggregated for predicting the surface. A self-attention mechanism [5] is utilized for feature aggregations across multiple views, and then 3D sparse convolutions are performed on the feature volume, followed by Gated Recurrent Unit (GRU) [6] to temporally fuse the feature volume fragment into the global one. We also utilize traditional TSDF fusion [7] with the available depths from MVS to generate a rough TSDF map, which is used as an additional feature channel to guide the feature volume-based reconstruction. The contribution of this paper can be summarized as follows:

- We develop a real-time incremental reconstruction system for monocular video images based on novel sparse feature volume fusion.
- We propose to utilize MVS neural networks to predict

Manuscript received: December 17, 2022; Revised March 25, 2023; Accepted April 17, 2023.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

[1] School of Computation, Information and Technology, Technical University of Munich, Germany. Email: firstname.lastname@tum.de

[2] Munich Center for Machine Learning (MCML), Germany.

[3] University of Delaware, USA. Email: nmerrill@udel.edu

[4] Department of Computing, Imperial College London, United Kingdom.

[5] University of Toronto Robotics Institute, University of Toronto, Canada. Email: binbin.xu@utoronto.ca

[6] Munich Institute of Robotics and Machine Intelligence (MIRMI), Germany.

[†] Corresponding author.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

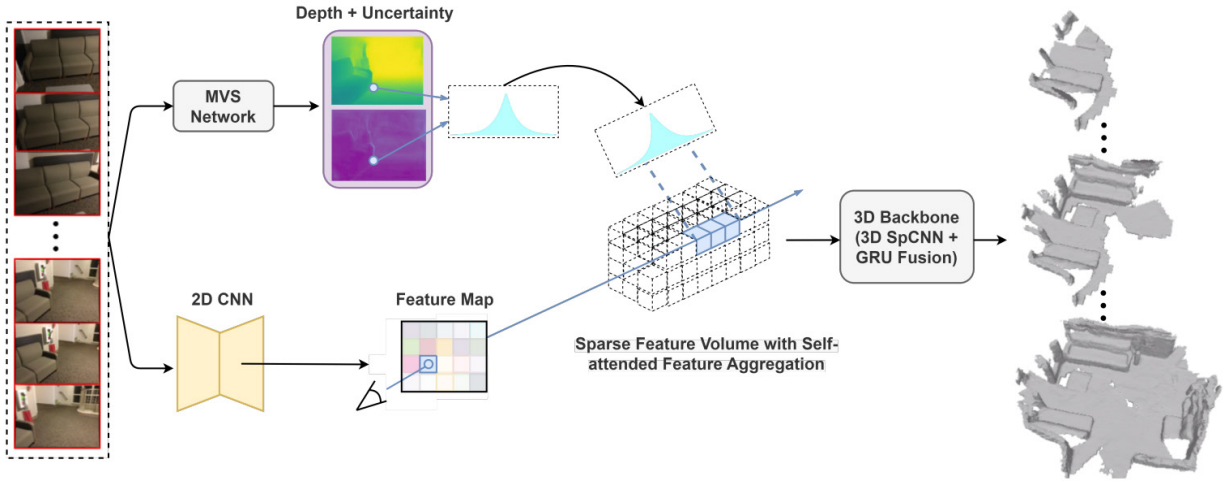

Fig. 1: The overview of our proposed method. We leverage an MVS neural network for depth and depth uncertainty predictions, which provide an initial guess of the physical surface location. Then feature volume-based reconstruction pipeline allocate and aggregate deep features around the physical surface, formatting sparse feature volume. TSDF values can be directly regressed from the feature volume which is incrementally updated. Note that the pipeline involves three levels of feature volumes, which are omitted here for simplicity.

initial depth and depth uncertainty maps for efficient feature allocation into 3D voxels, which maintains the natural sparsity of the problem and allows our method to recover more fine-grained details and to work effectively in outdoor large-scale scenarios.

- The proposed method is verified on various datasets, and demonstrated to have competitive reconstruction accuracy compared to state-of-the-art methods, and able to predict at a higher resolution than previous feature volume-based methods.

II. Related Work

Dense reconstruction from monocular videos has become increasingly accurate and robust with the emergence of deep learning. Our work is relevant to multi-view stereo (MVS) networks and feature volume-based scene reconstruction.

A. Depth-based Reconstruction with MVS Networks.

Given a set of monocular images with known poses and camera intrinsics, the goal of MVS networks is to infer the dense depth map of the reference frame using the provided multi-view information. Inspired by traditional plane-sweeping MVS methods [8], a rich body of MVS neural networks first construct a plane-sweeping cost volume using the image intensity values [9, 10] or deep 2D features generated from images [11–14]. Then the depth map of the reference image can be regressed from the cost volume through 2D convolutions [9, 10, 15] or 3D convolutions [11, 12].

Most of the methods that can be considered for real-time applications utilize 2D convolutions. DeepMVS [15] exploits patch matching for plane-sweep volume generation, and incorporates both intra-volume and inter-volume feature aggregation across an arbitrary number of input images. MVDepthNet [9] is an efficient network enabling real-time applications. It generates a cost volume directly from the warped image pixel values, and regresses the depth via a lightweight encoder-decoder network composed of 2D convolutions and skip connections. GPMVS [10] further extends MVDepthNet with the introduction of Gaussian Process (GP) prior to the bottleneck layer. The intuition is to leverage a pose kernel to measure the difference between camera poses, and to encourage similar poses to have similar latent variables in the bottleneck layer. With only a slight computation increment originating from the GP prior constraint, GPMVS can achieve much higher accuracy on dense depth regression over MVDepthNet. SimpleRecon [16] achieves high accuracy for depth prediction by including extra information in the cost volume via parallel MLP reduction of readily-available metadata – such as dot product of features across multiple views, back-projected rays from pixels, pose distances, and a validity mask – but incurs a higher computational cost than competing methods. In order to lift the 2D depth images into a 3D volumetric representation, traditional TSDF fusion [7, 17] is typically utilized by these methods.

MVSNet [11] constructs a 3D cost volume based on the variance of deep features, and further regularizes and regresses the depth map of the reference image via 3D convolutions – which enables higher accuracy than 2D convolutions but pays the cost of higher computation and memory consumption. To alleviate this issue, and allow depth prediction at higher resolutions, Gu et al. propose a cascade 3D cost volume [18] – narrowing the depth range of cost volume gradually from coarse to fine scales. PatchmatchNet [13] imitates the traditional PatchMatch method [19] by an end-to-end trainable architecture, which reduces the number of 3D convolutions needed in cost volume regularization and allows prediction at an even higher resolution than the cascade method [18]. While the accuracy and efficiency of the MVS methods utilizing 3D convolutions is ever increasing, they are still mostly amenable to offline processing.

B. Feature Volume-Based Scene Reconstruction.

SurfaceNet [20] is among the first works to directly predict surface probability from 3D voxelized colored volumes from two image views using a 3D convolutional network. For generating the 3D voxelized colored volume, all the pixels on images are projected into 3D through the known camera intrinsics and extrinsics. Two colored volumes are then concatenated along the color channel, and regularized by 3D convolutions. Atlas [1] extends this idea by replacing the

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

colored volume with more informative deep feature volumes, and further enables an arbitrary number of multi-view images. Constructed 3D volumetric deep feature volumes across multiple views go through average pooling before being fed into the 3D convolutions. TransformerFusion [1] and Vortx [4] exploit transformer-based attention mechanism for aggregating features from multiple image views instead of average pooling. TransformerFusion [1] also leverages the predicted attention weights to select the most relevant information for fusion. In order to alleviate the effects of occlusion in the aggregation of multi-view image features, Vortx [4] predicts the projective occupancy probabilities, which are used as weights to produce the aggregated feature in the volume.

Note that none of the aforementioned feature volume-based methods are aiming for real-time applications. TransformerFusion [1] processes every image frame one by one, and gradually selects a certain number of the most relevant feature measurements for every voxel grid based on attention weights. Besides, expensive dense 3D convolutions are utilized to deal with the feature volume. Those mentioned assignment choices make TransformerFusion far from real-time capable. Vortx [4] does not work in an incremental way, and it predicts the TSDF map from the final integrated feature volume with the aggregated features from certain selected image views in the whole video stream.

In contrast to the above methods, we focus on real-time incremental reconstruction based on feature volumes. The closest work to our method is NeuralRecon [2], which is also a baseline of our method. It performs feature volume-based reconstruction fragment by fragment at the first phase, which appears similar in spirit to active sliding windows in traditional SLAM methods [21–24]. In order to get a globally consistent reconstruction, NeuralRecon further adopts GRU-Fusion at the second phase to fuse the fragment feature volumes over time, which can be regarded as an alternative to the conventional TSDF fusion [7, 17]. NeuralRecon does not have access to pick out the most relevant features from the whole video before the feature volume fusion and processing, thus it is supposed to have inferior performance than the full-batch methods [1, 4]. Distinct from all the existing feature volume-based methods that unproject and allocate features into the voxel grids along the whole rays in feature volume, we leverage MVS for rough dense depth predictions – which allows us to allocate features to sparse voxels around the physical surfaces only. The retained sparsity keeps the memory consumption low and further enables high-resolution feature volume for scene reconstruction.

III. Methodology

We take as input a fragment sequence of N keyframe images $\{\mathbf{I}_k\}_{k=0}^{N-1}$ along with their corresponding poses $\{\mathbf{T}_k\}_{k=0}^{N-1}$ and camera intrinsics $\{\mathbf{K}_k\}_{k=0}^{N-1}$, $N = 9$ is used in our work. Following [2, 14], we utilize a 2D feature extraction network composed of an MnasNet encoder [25] and feature pyramid network (FPN) [26] style decoder. We unproject extracted features into a 3D aggregated feature volume representation and directly regress the sparse TSDF values from the feature volume. The key insight in our method is the use of *depth priors* to construct a feature volume that is sparse from the very start – allowing our 3D network to focus on the surface from the very beginning without wasting effort in allocating and processing dense volumes. An overview of our system can be seen in Fig. 1.

A. MVS-Guided Sparse Feature Allocation

Unlike existing feature volume-based methods [1, 2, 4] that allocate dense feature volumes from unprojected features, we utilize depth priors (depth map and its uncertainty) to allocate feature volume locations only where the physical surface is likely located. To get the depth priors of every keyframe in the fragment for efficient feature allocation, we leverage GPMVS [10] due to its appealing efficiency and adequate accuracy. Due to the modularity of our design, other MVS methods like [14, 16] could also be applied to our method.

1) MVS-Based Depth and Uncertainty Prediction: GPMVS [10] predicts the inverse depth map $\hat{\mathbf{D}}_{L_i}^{-1}$ at four scales $i \in \{0, 1, 2, 3\}$, and applies supervision at the four levels by resizing the ground truth inverse depth $\mathbf{D}_{L_i}^{-1}$. Note that we use $\hat{\cdot}$ to denote the estimated/predicted variables and, for ease of notation, assume operations (inverse, exp, etc.) to be element-wise throughout this paper. We augment the GPMVS network architecture to enable the prediction of dense depth uncertainty $\hat{\mathbf{B}}$ which is parameterized by $\hat{\mathbf{B}} = \exp(\hat{\mathbf{B}}_{\log})$ to ensure a positive uncertainty value. To predict $\hat{\mathbf{B}}_{\log}$ we simply duplicate the last three layers of the decoder in GPMVS to create a shallow second decoder head at the highest resolution. For the highest GPMVS resolution L_0, following [27, 28], we apply the Laplacian maximum likelihood estimator (MLE) loss to enforce that the predicted uncertainty tightly bounds the true prediction error:

$$\mathcal{L}_{\text{mvs}}^{(0)} = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \frac{|\hat{D}_{L_0}^{-1}(\mathbf{u}) - D_{L_0}^{-1}(\mathbf{u})|}{\hat{B}(\mathbf{u})} + \log \hat{B}(\mathbf{u}) \quad (1)$$

where Ω is the set of pixels with valid ground truth. For the other three resolutions with no uncertainty prediction, we use the standard ℓ_1 loss. The losses $\mathcal{L}_{\text{mvs}}^{(i)}$ from all resolutions are added together with equal weights, and mean reduction is used across batches. For the remainder of this work, we will only use the (inverse) depth at the highest resolution, L_0, and omit the respective subscript for brevity.

We leverage linear uncertainty propagation to convert the uncertainty of inverse depth to the uncertainty of depth:

$$\hat{\mathbf{C}} = \hat{\mathbf{D}}^2 \odot \hat{\mathbf{B}} \quad (2)$$

where $\odot$ is the element-wise product.

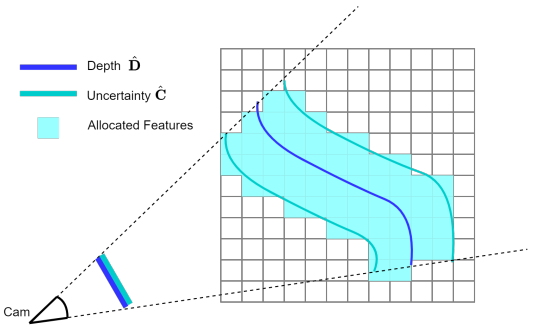

Fig. 2: 2D illustration of sparse feature allocation with the predicted depth map $\hat{\mathbf{D}}$ and accompanying uncertainty map $\hat{\mathbf{C}}$.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

2) Sparse Feature Allocation: Here we present the scheme to do sparse feature allocation based on casting the predicted depth and uncertainty (Sec. III-A1) into sparse voxels. Fig. 2 shows a visual representation of how the sparse feature volume is constructed. For each keyframe in a fragment, we utilize the predicted depth $\hat{\mathbf{D}}$ and uncertainty map $\hat{\mathbf{C}}$ to determine where we should allocate voxels in the sparse feature volume. Specifically, we only allocate voxels in the feature volume that lie along the ray of a pixel with positive predicted depth value and within the range of uncertainty bounds $[\hat{\mathbf{D}} - s\hat{\mathbf{C}}, \hat{\mathbf{D}} + s\hat{\mathbf{C}}]$. In our experiments, we use $s = 2$.

B. Multi-view Feature Aggregation via Self Attention

Deep image features at three resolutions are obtained by the feature extraction backbone, consisting of an efficient variant of MnasNet encoder [25] followed by a feature pyramid network (FPN) [26]. A 3D voxel location can be observed by multiple images from different viewpoints. We perform the feature aggregations in a content-aware way via multi-head self-attention [5] at three scales. We first project a 3D voxel location into image planes across different views by the known camera poses and camera intrinsics, and fetch features for this specific voxel from the extracted multiple-scale feature maps through differentiable bilinear interpolation. Since the feature aggregations procedure at different scales is the same, we exemplify it at one scale. The fetched features from N views, $\mathbf{F}_{\text{bp}} \in \mathbb{R}^{N \times C}$, with zero padding for the invisible views, and visibility binary mask $\mathbf{M} \in \mathbb{R}^{N}$ are the input to the self-attention based feature aggregation module, which outputs the content-aware features with viewpoint and data dependencies:

$$\mathbf{F}_{\text{attn}} = f_{\text{attn}}(\mathbf{F}_{\text{bp}}, \mathbf{M}_{\text{attn}}) \tag{3}$$

where the output feature $\mathbf{F}_{\text{attn}} \in \mathbb{R}^{N \times C}$ has the same feature channel with the input. Following [4], we realize the above self-attention aggregation module $f_{\text{attn}}(\cdot)$ by two transformer layers following the original transformer pipeline [5]. Each layer includes a multi-head attention mechanism (two heads in our implementation), as well as layer normalization, linear layers with ReLU activation, and residual connections. All the query, key, and value features originate from the same input features in the multi-head attention mechanism. In practice, we use two heads for the multi-head attention module. The aggregated features $\mathbf{F}_{\text{attn}}$ are simply averaged to generate a single feature vector $\mathbf{F}$ for the specific voxel.

C. Fragment Reconstruction from Sparse Feature Volume

After aggregating features from different views, we obtain a single feature vector within each non-empty voxel. We then directly regress the TSDF value of the voxel from this feature vector. With the predicted dense depth maps of keyframes in the fragment, it is handy to perform conventional TSDF-fusion [7] and get the TSDF values and weights for every voxel inside the chunk. The TSDF value and weight are concatenated with the averaged image feature for subsequent 3D sparse convolutions [29]. The final TSDF values of the chunk can be directly predicted from the feature volume by an MLP layer.

D. Fragment to Global Fusion

We follow NeuralRecon [2] to fuse the fragment feature volume into a global feature volume incrementally via GRU fusion [6]. For a feature vector $\mathbf{F}_t$ originating from current fragment in the feature volume at time instant t, we fuse it with the historical feature $\mathbf{H}_{t-1}$ at the same voxel location by GRU fusion. We observe that the volume resulting from traditional TSDF fusion with the predicted MVS depth is an additional useful feature for the network predicting the TSDF volume. Thus, the fused TSDF values and weights, $\mathbf{S}_t$ and $\mathbf{S}_{Wt}$, from fusing the MVS depths are concatenated with features in order to guide the fusion process. After the concatenation with increased feature dimensions, we leverage single-layer MLPs for feature dimension reduction.

$$\mathbf{H}'_{t-1} = \text{MLP}_H\left([\mathbf{H}_{t-1}, \mathbf{S}_t, \mathbf{S}_{Wt}]\right) \tag{4a}$$

$$\mathbf{F}'_{t} = \text{MLP}_F\left([\mathbf{F}_{t}, \mathbf{S}_t, \mathbf{S}_{Wt}]\right) \tag{4b}$$

$$\mathbf{z}_t = \text{sigmoid}\left(\text{SpConv}\left(\left[\mathbf{H}'_{t-1}, \mathbf{F}'_t\right]\right)\right) \tag{4c}$$

$$\mathbf{r}_t = \text{sigmoid}\left(\text{SpConv}\left(\left[\mathbf{H}'_{t-1}, \mathbf{F}'_t\right]\right)\right) \tag{4d}$$

$$\breve{\mathbf{H}}_t = \tanh\left(\text{SpConv}\left(\left[\mathbf{r}_t \odot \mathbf{H}'_{t-1}, \mathbf{F}'_t\right]\right)\right) \tag{4e}$$

$$\mathbf{H}_t = (\mathbf{I} - \mathbf{z}_t) \odot \mathbf{H}'_{t-1} + \mathbf{z}_t \odot \breve{\mathbf{H}}_t \tag{4f}$$

where $\mathbf{z}_t$ is the update gate vector, $\mathbf{r}_t$ the reset gate vector, $[\cdot, \cdot]$ the concatenation operator. SpConv denotes the sparse point-voxel convolution operation [29]. With the above GRU fusion, we can temporally fuse features and keep updating the visible feature volumes at three scales.

E. Implementation Details

We maintain three levels of feature volumes and regress the TSDF values $\mathbf{S}_{L_i}$ from them in a coarse to fine manner. In order to further sparsify the feature volume for the proceeding scales, we also predict occupancy values $\mathbf{O}_{L_x}$ from feature volumes at all scales with simple MLP layers. If the occupancy prediction of a voxel at a coarser scale is lower than a threshold (0.5), that voxel is redeemed as empty and will not be involved in feature allocation and prediction at finer scales [2]. Overall, the training loss for regressing TSDF and occupancy at a single resolution i is:

$$\mathcal{L}^{(i)}_{\text{recon}} = \frac{1}{|\Lambda|} \sum_{\mathbf{x} \in \Lambda} \Big(\lambda_1 |\text{logt}(\hat{S}_{L_i}(\mathbf{x})) - \text{logt}(S_{L_i}(\mathbf{x}))| + \lambda_2 \text{BCE}(\hat{O}_{L_i}(\mathbf{x}), O_{L_i}(\mathbf{x})) \Big) \tag{5}$$

where Λ is the set of voxels with valid ground truth, and $\text{logt}(\mathbf{S}_{L_i}) = \text{sign}(\mathbf{S}_{L_i}) \log(|\mathbf{S}_{L_i} + 1|)$ denotes the log-transform [2, 30], and BCE denotes the binary cross-entropy (BCE) loss. We have $\lambda_1 = \lambda_2$ for balancing the two loss terms in our training. We add the losses $\mathcal{L}^{(i)}_{\text{recon}}$ at each scale and apply mean reduction over batches.

The input images are at resolution 640×480, while the features at three levels are fetched from feature maps at resolutions 320×240, 160×120, 80×60 with channels $24, 40, 80$, respectively. GPMVS requires input images with resolution 320×256, which are obtained by bilinear interpolation-based downsampling. Dense depth maps are downsampled via

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

TABLE I: 3D geometry metrics evaluated on ScanNet test split. Note that the methods under the middle line are feature-volume-based incremental reconstruction methods, while the others are not.

Method	Comp ↓	Acc ↓	Recall ↑	Prec ↑	F-score ↑
COLMAP [31]	0.069	0.135	0.634	0.505	0.558
MVDepthNet [9]	**0.040**	0.240	0.831	0.208	0.329
DPSNet [12]	0.045	0.284	**0.793**	0.223	0.344
GPMVS [10]	0.105	0.191	0.423	0.339	0.373
Atlas [1]	0.083	0.101	0.566	0.600	0.579
Vortx [4]	0.081	**0.062**	0.605	**0.689**	**0.643**
NeuralRecon [2]	0.137	**0.056**	0.470	**0.678**	0.553
Ours	**0.110**	0.058	0.505	0.665	0.572
Ours (High Reso)	0.116	**0.056**	**0.525**	0.675	**0.589**

TABLE II: 2D depth metrics evaluated on ScanNet test split. Note that the methods under the middle line are feature-volume-based incremental reconstruction methods, while the others are not.

Method	Abs-rel ↓	Abs-diff ↓	Sq-rel ↓	RMSE ↓	$\delta < 1.25$ ↑	Comp ↑
COLMAP [31]	0.137	0.264	0.138	0.502	83.4	0.871
MVDepthNet [9]	0.098	0.191	0.061	0.293	89.6	0.928
DPSNet [12]	0.087	0.158	**0.035**	0.232	92.5	0.928
GPMVS [10]	0.088	0.206	0.053	0.359	90.3	0.928
Atlas [1]	0.065	0.123	0.045	0.249	92.4	**0.986**
Vortx [4]	**0.057**	**0.090**	**0.035**	**0.197**	**93.9**	0.951
NeuralRecon [2]	0.064	0.097	0.036	0.191	93.5	0.888
ours	0.057	0.092	0.030	0.183	94.2	**0.913**
ours (High Reso)	**0.052**	**0.087**	**0.025**	**0.175**	**94.8**	0.906

nearest neighbor to better preserve sharp edges. We utilize the TorchSparse [29] implementation of sparse 3D convolutions in our method.

IV. Experiments

A. Datasets and Metrics

For all the evaluations, we use the ScanNet dataset [32] for training, which consists of 1513 RGBD sequences collected in 707 indoor scenes. We follow the official training and test splits, which are 1201 and 100 sequences, respectively. Besides the ScanNet test split, we also test the ScanNet-trained network zero-shot on TUM-RGBD [33] (13 sequences following [4]), and our own collected dataset without any finetuning. For 3D metrics, we evaluate the final reconstructed surface mesh extracted from the predicted TSDF volume against the officially provided mesh on ScanNet, and we generate our own meshes on TUM-RGBD and our own collected datasets through conventional TSDF fusion using the ground truth depth. Following the evaluation protocol [1, 2] exactly, we calculate 3D metrics, including accuracy, completeness, precision, recall, and F-score, across uniformly sampled points with a 2-centimeter resolution from dense meshes. For computing these 3D metrics, a distance threshold of 5 centimeters was used. We regard the F-score to be the most representative metric to reflect the quality of 3D reconstruction, since it is involved with both precision and recall. Regarding 2D metrics, we evaluate the rendered depth maps at all the image views for the feature volume-based scene reconstruction methods, against the provided raw depth maps with a truncation of 10 meters. The MVS methods predicting dense depth maps directly allow for handy evaluations.

B. Training Details

We use ScanNet dataset [32] for training. Ground truth TSDF volumes are generated from raw depth maps and given camera poses by conventional TSDF fusion [7, 17]. Note that we discard all the depths over $3m$ like existing methods [2, 4]. Before training the feature volume pipeline, we need to fine-tune the lightweight GPMVS [10] depth prediction network for 12 epochs, then train the depth variance prediction network for 4 epochs from randomly initialized weights. Note that, GPMVS is frozen in subsequent training, while the weights of the variance network are kept updated. We have two phases for training the feature volume pipeline for incremental reconstruction from monocular videos. At the first phase, we train the fragment-wise reconstruction network, regressing TSDF from feature volume for 20 epochs. The network learns how to predict TSDF from aggregated features in a fragment volume with size $3.84m \times 3.84m \times 3.84m$. Finally, we train the network at the second phase with the GRU fusion network together for 30 epochs. Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ is adopted for training the networks, and the learning rate is $1e-3$ at the beginning and decreased by half at epochs $12, 24, 48$ in the two-phase training. The finest voxel resolution is $4cm$ by default, and is reduced to $2cm$ for our high-resolution variant. The batch size is 32 by default, and 8 for the higher resolution variant.

C. Evaluation on ScanNet

The evaluation results with both 3D metrics and 2D metrics are shown in Table I and Table II, respectively. We compare the proposed method, *Ours*, with state-of-the-art traditional structure-from-motion method, COLMAP [31], multi-view stereo networks including MVDepthNet [9], DPSNet [12], and GPMVS [10], as well as all the open-source feature-volume-based methods including Atlas [1], Vortx [4], and NeuralRecon [2]. Our method with high-resolution feature volume is named *Ours (High Reso)*. It should be noted all the compared methods are evaluated with the same protocol. All compared deep-learning-based methods have been trained or fine-tuned on the ScanNet dataset. The results of COLMAP, MVDepthNet, and DPSNet are taken from [1], while GPMVS, Atlas, Vortx, and NeuralRecon are evaluated by ourselves.

The proposed incremental feature volume-based method outperforms the existing incremental method, NeuralRecon, regarding the representative 3D reconstruction metric – the F-score. The advantages on F-score are mainly from the higher recall, while the proposed method and NeuralRecon have very similar precision. With the incorporation of MVS depth and depth uncertainty in our method, more structures can be recovered compared to the pure feature volume-based NeuralRecon. When the voxel size of feature volume is decreased from $4cm$ to $2cm$, dubbed as *Ours (High Reso)*, the overall quality of reconstruction can be further improved. We attempted to train NeuralRecon [2] with a high-resolution feature volume as well for a direct comparison, but the training network was unable to fit in GPU memory (see Sec. IV-G). It should also be noted that both Atlas [1] and Vortx [4] are offline methods, and have the access to full batch data with global context before predicting TSDF values from features, and they are expected to exhibit better performance than our real-time incremental method. Actually, *Ours* as the real-time incremental method has very close performance to Atlas [1] and slightly better performance than traditional offline method COLMAP [31].

We also show qualitative results of the meshes generated from different methods in Fig. 3. The MVS method, GPMVS [10], which is also leveraged in our pipeline to guide the feature allocation, suffers from significant artifacts. With

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

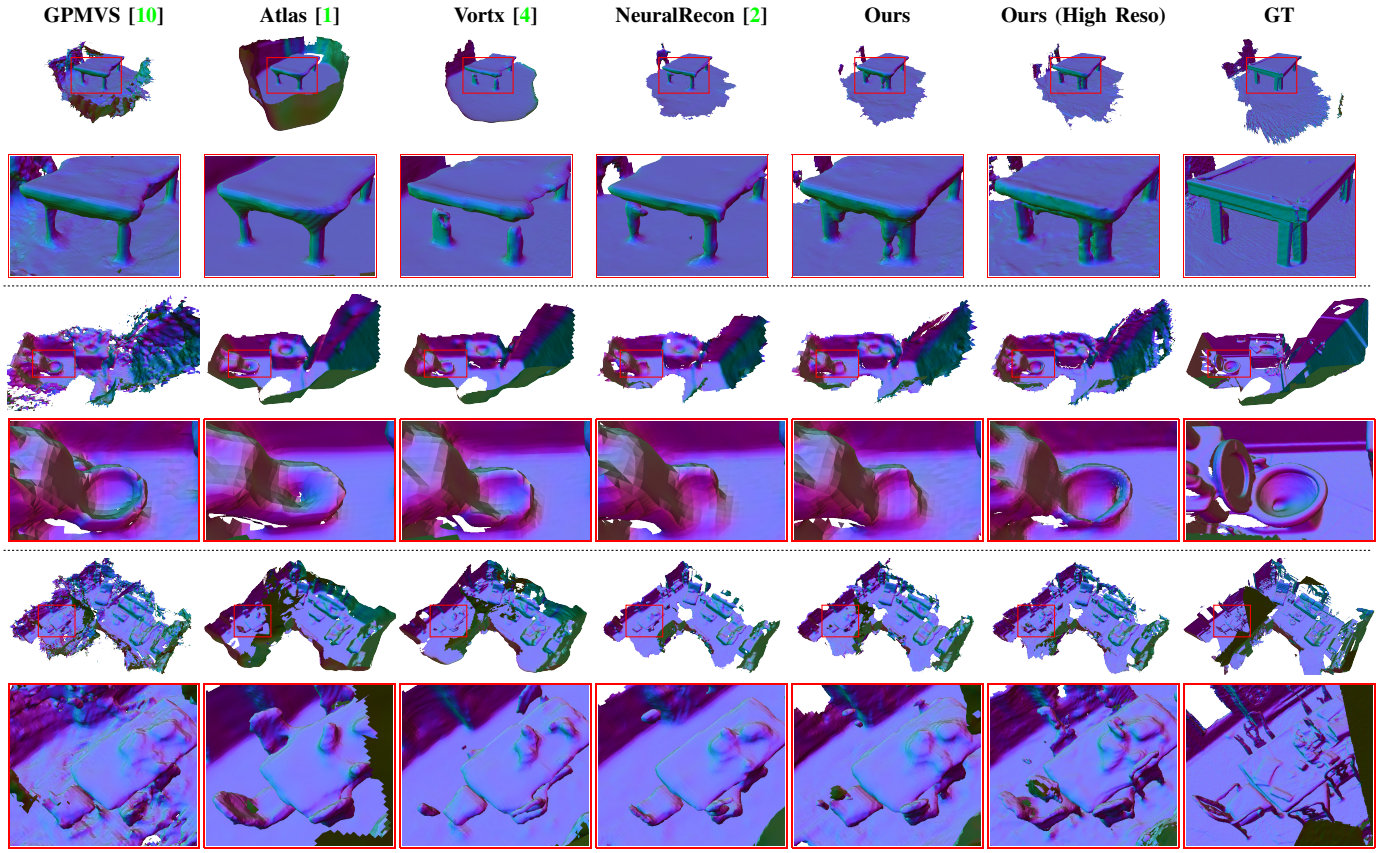

Fig. 3: qualitative results on Scannet test sequences [32]. We also zoom in the region in red rectangles for clear views. The proposed method, *Ours*, can recover more 3D structures than the incremental feature-volume-based method NeuralRecon [2]. With a higher resolution, *Ours (High Reso)* is able to recover more fine-detailed structures.

the feature volume-based regularization and denoising, our method is able to remove the main artifacts in MVS reconstruction. This also verifies that the predicted depth and depth uncertainty for feature allocation only near the surface voxels are reasonable, and have provided the rough shape sufficient for the proceeding feature volume-based reconstruction. It is found that *Ours* and *Ours (High Reso)* have the potential to recover more details about objects, such as the table leg, toilet, and chair. Atlas [1] and Vortx [4] are prone to recover more complete walls and floors, but their predictions can be over-smoothed to miss details of objects, and over-filled with hallucinated structures.

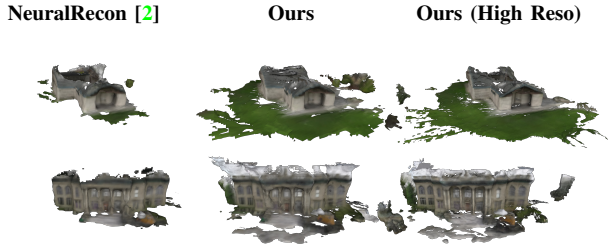

Fig. 4: Qualitative results in outdoor large-scale scenarios on Tanks & Temples [34] sequences.

D. Generalization on TUM-RGBD

To examine the generalization of the proposed method, we also conduct comparisons on TUM-RGBD dataset [33]. We only compare to the incremental feature volume-based methods. We follow exactly the identical keyframing strategy as the one on Scannet dataset. The 3D evaluation metrics are shown in Table III. We can find that *Ours* outperforms NeuralRecon [2]. Our high-resolution variant has obvious advantages over the default resolution.

TABLE III: Evaluated on TUM-RGBD dataset.

Method	Comp ↓	Acc ↓	Recall ↑	Prec ↑	F-score ↑
NeuralRecon [2]	0.285	0.101	0.109	0.391	0.169
Ours	0.252	0.123	0.133	0.369	0.195
Ours (High Reso)	**0.192**	**0.084**	**0.199**	**0.586**	**0.295**

E. Generalization on Tanks & Temples

We further conduct evaluations in two large-scale outdoor scenarios, “Barn” and “Courthouse”, using the Tanks & Temples dataset [34] without any fine-tuning of the network weights trained on the indoor ScanNet dataset [32]. Our proposed method is designed to maintain sparsity in the feature volume, making it well-suited for large-scale volumetric reconstruction where most voxel grids are physically empty. In contrast, allocating features to a large number of voxels in dense volume can become computationally intractable. In order to run NeuralRecon [2] successfully on these large-scale scenarios, the poses of images for all the compared methods are scaled down to be 5 to 10 times smaller. The dense reconstruction results are qualitatively depicted in Fig. 4. Our method demonstrates strong generalization capabilities in outdoor large-scale scenarios, visibly outperforming NeuralRecon with much more desirable reconstruction results. Moreover,

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

our high-resolution reconstruction can recover finer details and more accurate scene structures.

F. Generalization on Self-Collected Data

We also collect our own dataset with 8 sequences in indoor scenarios by the Realsense D455 RGBD camera [1]. We record the grayscale stereo images, RGB images, depth maps, and IMU data streamed from the camera. The stereo images and IMU data are fed into a visual-inertial SLAM system, OKVIS 2.0 [35], for getting accurate 6DoF poses. A snapshot of a typical scenario, and the reconstructed meshes from compared methods are shown in Fig. 5, where we can easily find that our methods can recover more geometric details than NeuralRecon [2]. The quantitative 3D metric evaluations are also shown in Table IV.

TABLE IV: Evaluated on our own collected data.

Method	Comp ↓	Acc ↓	Recall ↑	Prec ↑	F-score ↑
NeuralRecon [2]	0.205	**0.034**	0.215	**0.774**	0.335
Ours	0.144	0.040	0.255	0.711	0.374
Ours (High Reso)	**0.143**	0.045	**0.268**	0.683	**0.385**

G. Memory and Runtime

We conducted runtime and memory evaluations of the inference stage on a desktop computer equipped with an RTX5000@16GB GPU and 8 Intel i7-11700k CPU Cores@3.6GHz. In Table V, we report the averaged time taken for MVS depth recovery, feature encoding, feature aggregation, TSDF-Fusion of MVS depths, 3D sparse CNN, GRU fusion, and the total processing time for a volume chunk with 9 keyframes. The experiment is conducted on a typical large-scale indoor sequence of ScanNet at the inference stage. Despite incorporating MVS depth and self-attention, our method can run incremental reconstruction in real-time at 12.70 keyframes per second , with a slightly increased memory footprint compared to NeuralRecon. At the high resolution, our method runs at 5 keyframes per second. Although this is admittedly slower than NeuralRecon, which can run at 41 keyframes per second on the same device, it should be noted that our method is still real-time capable since keyframes in a typical real-time SLAM system (e.g., [35]) are created at a far lower frequency than the framerate.

Our method is additionally highly memory-efficient in terms of voxel allocation. At the inference stage, compared with the 'dense' feature volume of NeuralRecon, the numbers of the non-empty voxel with allocated features in our sparse volume are significantly decreased by $67.71\%, 48.24\%, 30.71\%$ at three levels respectively. The feature volume in NeuralRecon is initially dense, and is sparsified by the network from coarse to fine levels. This means that, in the initial phases of training, before the network learns to sparsify well, a nearly dense volume makes its way through the sparse 3D convolution network. In our experiments, we did not train NeuralRecon at the high resolution due to this issue since the required GPU memory exceeded 42GB – even with batch size 1. In contrast, due to our MVS-guided sparse volume allocation, which makes feature volumes sparse at all three levels, we are able to train at the higher resolution even with the memory-intensive attention mechanism involved, consuming GPU memory around 28GB during training.

[1]https://www.intelrealsense.com/depth-camera-d455

H. Ablation Study

To examine the effectiveness of our design choices, we conduct ablation studies on the ScanNet dataset at the default resolution. The results are reported in Table VI. We first examine our method without sparsification. In this case, the F-score is a bit higher, while running our method without the sparsification incurs a higher memory consumption due to more voxels needing to be allocated. We further ablate our method by removing the predicted depth uncertainty for feature allocation – allocating features within a constant distance of 5 voxels around the surface recovered from the predicted MVS depth. Noticeably, our full method has better performance due to the probabilistic feature allocation accounting for the uncertainty of predicted MVS depth. It is also clear from the table that our choice of feature augmentation using the TSDF values and weights generated from trivial TSDF fusion of MVS depth, as well as our choice of self-attention for feature aggregation, have significant effects on the system performance. The hints from MVS TSDF values and weights can guide the network for better convergence. The attention mechanism for feature aggregation from multiple views enables selectively absorbing informative deep features for 3D structure recovery. For the last ablation study in Table VI, we investigate whether it is possible to reduce the number of parameters by sharing the feature encoder of the MVS Network with the 2D CNN for feature extraction (see Fig. 1), instead of using separate feature encoders for the two modules in our design choice. Notably, the performance is significantly deteriorated due to the fact that MVS and feature volume fusion require different features.

I. Limitations and Discussions

Our proposed method leverages MVS depth to guide incremental feature volume-based reconstruction. MVS can provide rough locations of the physical surface and enable sparse allocation, while feature volume-based fusion can further regularize, refine, and denoise the recovered 3D structures from MVS depth. With the MVS guidance, more geometric details can be recovered. However, it pays the cost that local smoothness can be slightly degraded. Since the sparse feature allocation is critical for our proposed method, if the MVS network fails to predict a depth distribution surrounding the true physical surface, the feature volume-based pipeline can not recover appreciable 3D structures from empty voxels.

V. Conclusion and Future Work

We presented a real-time incremental 3D dense reconstruction method from monocular videos based on MVS, attention mechanism, sparse 3D CNN, and GRU fusion. Predicted depth maps and uncertainties from MVS neural networks provide an initial guess of the physical surface locations in feature volume. Then feature volume-based pipeline temporally fuses the deep features into the sparsified feature volume to refine 3D geometries and impose local smoothness and 3D priors learned from data. The proposed method is demonstrated to perform accurate 3D dense reconstruction on several datasets, and can scale up to high-resolution reconstruction due to its memory-efficient nature in terms of sparse feature allocation.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2023.3273509

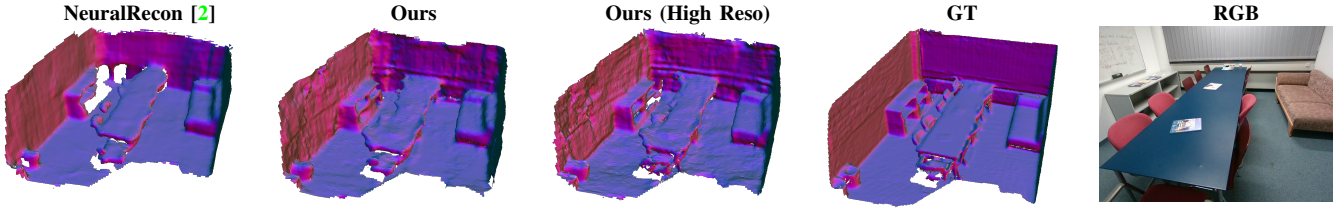

Fig. 5: Qualitative results of a representative sequence from our self-collected dataset.

TABLE V: Mean of runtime (Second), memory consumption (GB), and number of nonempty voxels in three-level feature volumes, during the reference on a typical sequence of ScanNet.

Method	MVS Dep.	Feat. Enc.	TSDF-Fusion	Feat. Agg.	3D SpCNN	GRU Fusion	Total
NeuralRecon [2]	-	0.037	-	0.012	0.096	0.063	0.219
Ours	0.377	0.025	0.061	0.099	0.060	0.057	0.711
Ours (High Reso)	0.384	0.025	0.181	0.573	0.225	0.315	1.779

Kf./Sec.	Voxels L0	Voxels L1	Voxels L2	Memory (GB)
41.10	4307.436	11893.692	46665.385	0.988
12.66	1390.744	6156.615	32336.641	1.68
5.06	6962.026	39470.615	230968.897	9.06

TABLE VI: Ablation study: 3D geometry metrics evaluated on ScanNet test split.

Method	Comp ↓	Acc ↓	Recall ↑	Prec ↑	F-score ↑
Ours (High Reso)	0.116	0.056	0.525	0.675	0.589
Ours	0.110	0.058	0.505	0.665	0.572
Ours: w/o sparsification	0.123	0.047	0.487	0.713	0.577
Ours: w/o depth uncer.	0.111	0.061	0.496	0.651	0.561
Ours: w/o tsdf augment.	0.120	0.063	0.472	0.637	0.540
Ours: w/o attention	0.119	0.073	0.451	0.592	0.511
Ours: w/o sep. feat. enc.	0.121	0.065	0.463	0.625	0.530

ACKNOWLEDGMENT

We thank Noah Stier [4], Jiaming Sun, and Yiming Xie [2] for discussions about the evaluations of baselines.

REFERENCES

[1] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich. "Atlas: End-to-end 3D scene reconstruction from posed images". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer. 2020, pp. 414–431.

[2] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao. "NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15598–15607.

[3] A. Božič, P. Palafox, J. Thies, A. Dai, and M. Nießner. "TransformerFusion: Monocular RGB Scene Reconstruction using Transformers". In: *Proc. Neural Information Processing Systems (NeurIPS)* (2021).

[4] N. Stier, A. Rich, P. Sen, and T. Höllerer. "VoRTX: Volumetric 3D reconstruction with transformers for voxelwise view selection and fusion". In: *2021 International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 320–330.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *NeurIPS Workshop on Deep Learning*. 2014.

[7] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. "KinectFusion: Real-time dense surface mapping and tracking". In: *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE. 2011, pp. 127–136.

[8] R. T. Collins. "A space-sweep approach to true multi-image matching". In: *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Ieee. 1996, pp. 358–363.

[9] K. Wang and S. Shen. "MVDepthNet: Real-time multiview depth estimation neural network". In: *2018 International conference on 3D vision (3DV)*. IEEE. 2018, pp. 248–257.

[10] Y. Hou, J. Kannala, and A. Solin. "Multi-view stereo by temporal nonparametric fusion". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2651–2660.

[11] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. "Mvsnet: Depth inference for unstructured multi-view stereo". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 767–783.

[12] S. Im, H.-G. Jeon, S. Lin, and I. S. Kweon. "DPSNet: End-to-end deep plane sweep stereo". In: *arXiv preprint arXiv:1905.00538* (2019).

[13] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys. "Patchmatchnet: Learned multi-view patchmatch stereo". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14194–14203.

[14] A. Duzceker, S. Galliani, C. Vogel, P. Speciale, M. Dusmanu, and M. Pollefeys. "DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15324–15333.

[15] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang. "DeepMVS: Learning multi-view stereopsis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2821–2830.

[16] M. Sayed, J. Gibson, J. Watson, V. Prisacariu, M. Firman, and C. Godard. "SimpleRecon: 3D Reconstruction Without 3D Convolutions". In: *European Conference on Computer Vision*. Springer. 2022, pp. 1–19.

[17] B. Curless and M. Levoy. "A volumetric method for building complex models from range images". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, pp. 303–312.

[18] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan. "Cascade cost volume for high-resolution multi-view stereo and stereo matching". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2495–2504.

[19] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. "PatchMatch: A randomized correspondence algorithm for structural image editing". In: *ACM Trans. Graph.* 28.3 (2009), p. 24.

[20] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang. "SurfaceNet: An end-to-end 3D neural network for multiview stereopsis". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2307–2315.

[21] A. I. Mourikis, S. I. Roumeliotis, et al. "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation." In: *ICRA*. Vol. 2. 2007, p. 6.

[22] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. "Keyframe-based visual-inertial slam using nonlinear optimization". In: *Proceedings of Robotis Science and Systems (RSS) 2013* (2013).

[23] J. Engel, V. Koltun, and D. Cremers. "Direct sparse odometry". In: *IEEE transactions on pattern analysis and machine intelligence* 40.3 (2017), pp. 611–625.

[24] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang. "LIC-Fusion: Lidar-inertial-camera odometry". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 5848–5854.

[25] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le. "MnasNet: Platform-aware neural architecture search for mobile". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2820–2828.

[26] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[27] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. "CodeSLAM—learning a compact, optimisable representation for dense visual SLAM". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2560–2568.

[28] X. Zuo, N. Merrill, W. Li, Y. Liu, M. Pollefeys, and G. Huang. "CodeVIO: Visual-inertial odometry with learned optimizable dense depth". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 14382–14388.

[29] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han. "Searching efficient 3D architectures with sparse point-voxel convolution". In: *European Conference on Computer Vision*. Springer. 2020, pp. 685–702.

[30] A. Dai, C. Diller, and M. 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*. 2020, pp. 849–858.

[31] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. "Pixelwise view selection for unstructured multi-view stereo". In: *European Conference on Computer Vision*. Springer. 2016, pp. 501–518.

[32] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. "ScanNet: Richly-annotated 3D reconstructions of indoor scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5828–5839.

[33] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 573–580.

[34] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. "Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction". In: *ACM Transactions on Graphics* 36.4 (2017).

[35] S. Leutenegger. "OKVIS2: Realtime Scalable Visual-Inertial SLAM with Loop Closure". In: *arXiv preprint arXiv:2202.09199* (2022).

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY OF DELAWARE LIBRARY. Downloaded on May 11,2023 at 13:00:32 UTC from IEEE Xplore. Restrictions apply.