DeepVideoMVS: Multi-View Stereo on Video with Recurrent Spatio-Temporal Fusion

Arda Düzçeker[1], Silvano Galliani[2], Christoph Vogel[2], Pablo Speciale[2], Mihai Dusmanu[1], Marc Pollefeys[1,2]
[1]Department of Computer Science, ETH Zurich
[2]Microsoft Mixed Reality & AI Zurich Lab

(a) Groundtruth

(b) No Spatio-temporal Fusion (Backbone Network)
Runtime ~ 30 ms • FPS ~ 34 • Memory = 795 MB

(c) With Spatio-temporal Fusion
Runtime ~ 37 ms • FPS ~ 27 • Memory = 1061 MB

Figure 1: 3D reconstructions of a scene from ScanNet [11]. Extending our stereo backbone with our proposed spatio-temporal fusion module improves the temporal consistency and accuracy of the predicted depth maps, leading to better reconstructions with negligible computational overhead. Runtime is per forward pass on an NVIDIA GTX 1080Ti with image size 320×256.

Abstract

We propose an online multi-view depth prediction approach on posed video streams, where the scene geometry information computed in the previous time steps is propagated to the current time step in an efficient and geometrically plausible way. The backbone of our approach is a real-time capable, lightweight encoder-decoder that relies on cost volumes computed from pairs of images. We extend it by placing a ConvLSTM cell at the bottleneck layer, which compresses an arbitrary amount of past information in its states. The novelty lies in propagating the hidden state of the cell by accounting for the viewpoint changes between time steps. At a given time step, we warp the previous hidden state into the current camera plane using the previous depth prediction. Our extension brings only a small overhead of computation time and memory consumption, while improving the depth predictions significantly. As a result, we outperform the existing state-of-the-art multi-view stereo methods on most of the evaluated metrics in hundreds of indoor scenes while maintaining a real-time performance. Code available: https://github.com/ardaduz/deep-video-mvs

1. Introduction

Obtaining dense 3D information about the environment is key for a wide range of applications such as navigation for autonomous vehicles (*e.g.*, robots, drones [44]), mixed reality [1, 2, 24, 42], 3D modelling and industrial control. Compared to active depth sensing with LiDAR [41], time-of-flight [19] or structured-light cameras [13], camera-based passive sensing has the advantage of being energy and cost efficient, compact in size and operating in a wide range of conditions [33]. Among passive depth sensing approaches, monocular systems can offer highly mobile, low-maintenance solutions, while stereo devices often require baseline sizes that are infeasible for mobile devices [46].

One common denominator of the aforementioned applications is that the data is acquired as a video stream instead of sparse instances in time, and the depth is often reconstructed for selected keyframes. In this work, we assume a calibrated camera and known poses between acquisitions and focus on the dense depth recovery for each keyframe. Such pose information can, for instance, be obtained through visual-inertial odometry techniques [6, 40], which are readily available in mobile platforms (*e.g.* Apple ARKit and Android ARCore) [20] or mixed reality headsets such as Microsoft HoloLens. The presence of camera poses enables the computation of triangulation-based metric reconstructions, as opposed to the popular learning-based single image depth prediction methods [3, 12, 14, 29, 55] that have been extended to video [39, 50, 52, 58]. Finally, the real-time aspect of the applications and the potential of an on-device solution, imply targeting a lightweight online multi-view stereo (MVS) system that is memory and compute efficient. Therefore, similar to [20, 32], we specifically aim to harness the advantages of video, with limited variation in viewpoint at consecutive time steps, instead of pursuing unstructured MVS.

In this work, we present a framework that can extend many existing MVS methods, such that, when processing

video streams, partial scene geometry information from the past contributes to the prediction at the current time step, leading to improved and consistent depth outputs. We use convolutional long short-term memory (ConvLSTM) [45] and a hidden state propagation scheme to achieve such information flow in the latent space. Our approach is influenced by [20], where latent cost volume encodings are weakly coupled through a Gaussian Process, and by [39] where latent encodings of image features and sparse depth cues are fused through ConvLSTM to achieve temporally consistent depth predictions. However, we leverage geometry and explicitly account for the implications of perspective projection when propagating the latent encodings.

Our key contributions are as follows: **(i)** We propose a compact, cost-volume-based, stereo depth estimation network that solely relies on 2D convolutions to obtain real-time and memory efficient processing. **(ii)** We extend our model with a ConvLSTM cell, an explicit hidden state propagation scheme, and a training/inference strategy to enable spatio-temporal information flow. This extension significantly improves the depth estimation accuracy, while creating only a small computational overhead, *c.f*. Fig. 1. **(iii)** We set a new state-of-the-art on ScanNet [11], 7-Scenes [16], TUM RGB-D [47] and ICL-NUIM [18], *c.f*. Tab. 1, while obtaining the lowest runtime and a small memory footprint, *c.f*. Fig. 4.

2. Related Work

The most common representation for learning-based MVS is depth map, where 3D reconstruction is performed at a later stage, if needed. Compared to direct inference in 3D space with either an explicit voxel discretization [23, 25], point-based representations [8], implicit neural scene representations [34, 35], or direct regression of a truncated signed distance function (TSDF) [36], depth map representations are more versatile and can be used for various other tasks in addition to 3D reconstruction. Furthermore, this simple 2D representation appears to be more memory efficient and capable of delivering real-time information.

Depth Map Prediction with Learned MVS. Most learning-based MVS methods follow traditional plane-sweeping [10, 15] to generate a cost volume from a designated reference and a measurement frame. On the one hand, methods such as MVSNet [53] and DPSNet [21] build 4D feature volumes [26], and regularize the feature volumes by employing 3D convolutions, a process that delivers high accuracy, but is computationally demanding. On the other hand, MVDepthNet [51] and [54], directly generate 3D volumes by computing traditional cost measures on image features or RGB values. This allows basing the network architecture on 2D convolutions, which are faster than the 3D counterpart, and better suited for real-time applications. As a compromise, predetermined cost measures decimate the color and feature information, which is often tackled by providing the reference image to the network, in addition to the cost volume. Targeting real-time performance, we propose a similar, especially lightweight network based on these principles as our backbone. A notable exception to cost-volume-based MVS is DELTAS [46], which learns to detect and triangulate interest points in the input images, and densifies the sparse set of 3D points to produce dense depth maps. However, the different approach of DELTAS can still be extended by our fusion framework.

Depth Estimation from Video. In typical video data, successive frames are strongly correlated and spatially close in the 3D space. Several single image depth prediction methods [39, 50, 58] and stereo depth estimation methods [57, 59] have shown that modelling the temporal relations or introducing optimization constraints among frames can improve the depth/disparity predictions.

In [50], only monocular image sequences are input and the temporal relations between image representations are modeled by placing many ConvLSTM cells in a single image depth estimation network. Similarly, [58] employs ConvLSTM to model the relations between the successive frames, and extend their model with a generative adversarial network [17] and a temporal loss, to enforce consistency among video frames. Despite achieving temporal consistency and visually pleasant results, [50] and [58] cannot ensure geometric correctness due to the lack of geometric foundation based on image measurements. In [39], input frames and sparse depth cues are encoded together, then relations between consecutive latent encodings are modeled through a ConvLSTM. Finally, the dense depth predictions are output by a decoder. [59] propose an unsupervised learning setting on stereo videos. They take the stereo image sequences as input and exploit the temporal dynamics by placing two separate ConvLSTMs in their network to predict more accurate disparity maps. [57] establishes temporal and stereo constraints on consecutive frames of the stereo sequence to improve the joint pose and depth estimations in their unsupervised framework. In contrast, known camera poses, *e.g*. MVS, do not only enable triangulated metric measurements from arbitrary set of frames, but also the transfer of the past scene geometry encodings into the current view geometry, which improves the accuracy in our model.

Lately, video input has been leveraged also in learning-based MVS. GP-MVS [20] extends MVDepthNet by introducing a Gaussian Process (GP) at the bottleneck between the cost volume encoder-decoder of MVDepthNet. The method constructs a GP prior kernel from a similarity measure between the known camera poses. This introduces soft constraints at the bottleneck layer and encourages the model to produce similar latent encodings for the frames that have similar poses. They also propose an inference scheme that evolves the GP in state-space for online operation. In com-

parison, we fuse the latent encodings through learned convolution filters, after perspective correction, whereas GP-MVS performs non-parametric fusion constrained by pose priors, without any perspective correction. Neural RGBD [32] estimates a depth probability distribution per pixel, and propagates the resulting probability volume through time. As successive video frames are processed, the depth probability volumes are aggregated under a Bayesian filtering framework, which contributes to the confidence of a pixel's depth hypothesis. In contrast, we propose to do the fusion at an earlier stage and operate on the latent encodings at the bottleneck, instead of explicitly working on probability volumes.

3. Method

For an online multi-view stereo system (with no temporal delay) that works on posed-video input, the supervised learning problem can be formulated as

$$\widehat{\mathbf{D}}_t = f_\theta(\mathbf{I}_t, \mathbf{I}_{t-1}, \dots, \mathbf{I}_{t-\delta}, \mathbf{T}_t, \mathbf{T}_{t-1}, \dots, \mathbf{T}_{t-\delta}, \mathbf{K}), \quad (1)$$

$$\theta^\star = \text{argmin}_\theta \; l\left(\widehat{\mathbf{D}}_t,\, \mathbf{D}_t\right). \quad (2)$$

The task is to learn a predictor f_θ with a set of learnable parameters θ that can infer a depth map $\widehat{\mathbf{D}}_t$, which is as close as possible to the groundtruth depth $\mathbf{D}_t$ [27] by having access to the camera intrinsic matrix $\mathbf{K}$, images $\mathbf{I}$ and camera poses $\mathbf{T} \in SE(3)$, corresponding to the current time step t and a number of δ previous time steps $t-1, \dots, t-\delta$. $l(\cdot,\cdot)$ is the loss between the inferred and the groundtruth depth map, which we want to minimize over a large dataset.

Our approach is discussed in two sections. Sec. 3.1 introduces a cost-volume-based, stereo depth prediction network, *c.f*. Fig. 2. This lightweight model serves as the backbone structure that we build upon. Sec. 3.2 discusses our novel approach that integrates information flow between successive frames in the latent space over time, *c.f*. Fig. 3.

3.1. Pair Network

This section introduces our *pair network*, a modified version of [51] that integrates additional feature extraction and feature pyramid network (FPN) [31] modules into the architecture, *c.f*. Fig. 2. The model function f_θ is acquired by assigning $\delta = 1$ in Eq. 1, *i.e*., it takes an intrinsic matrix, a reference image at time t, one measurement image from time $t-1$ and their poses, and predicts a depth map for the reference frame. The network consists of five main parts.

Feature Extraction with FPN. Our feature extraction is based on MnasNet [49], which is chosen due to its low-latency. To increase the receptive field and recover the spatial resolution, it is extended with a feature pyramid network (FPN), which is shown to be effective for object detection tasks [31]. MnasNet layers spatially scale down the feature maps until $\frac{H}{32}\times\frac{W}{32}$ and the FPN recovers the resolution up to $\frac{H}{2}\times\frac{W}{2}$. As the result of all convolution operations, a feature at the center of the half resolution feature map has a receptive field of 304×304. The output channel size of the FPN is $CH = 32$. The cost volume is constructed using only the feature maps at half resolution, while the lower resolution feature maps carried to the encoder with skip connections for additional high-level feature information.

Cost Volume Construction. We generate a cost volume by employing the traditional plane-sweep stereo [10, 15] with $M = 64$ plane hypotheses, each parameterized by its depth d_m and uniformly sampled in inverse depth space (uniform in pixel space) within the interval corresponding to $d_{near} = 0.25$, $d_{far} = 20$ meters. The cost induced by the m^{th} depth plane is calculated from the pixel-wise correlation between the reference feature map $\mathbf{F}$ and the warped measurement feature map $\tilde{\mathbf{F}}^m$. Our cost volume $\mathbf{V}$ is then composed as

$$V_{i,j,m} = -\langle \mathbf{F}_{i,j,:}\,, \tilde{\mathbf{F}}^m_{i,j,:} \rangle / CH, \quad \text{for } 1 \le m \le M. \quad (3)$$

Cost Volume Encoder-Decoder. The key purpose of the encoder-decoder is to spatially regularize the raw cost volume with a U-Net [43] style architecture. The encoder extracts the high-level, global scene information and aggregates the pixel-wise matching costs with the help of the feature maps coming from the feature extraction step. The decoder gradually upsamples the high-level encoding to the finer resolutions, using the low-resolution inverse depth maps and the skip connections coming from the encoder as guidance.

Depth Regression and Refinement. After acquiring the encoding $\mathbf{Y}$ at the output of each decoder block, we apply one 3×3 convolution filter $\mathbf{w}$ and a sigmoid activation σ. We pass this tensor to the next decoder block or the refinement block as a guidance, and also regress a depth map from it. For a pixel location (i,j) in $\widehat{\mathbf{D}}$, the regression is

$$\widehat{D}_{i,j} = \left(\left(\frac{1}{d_{near}} - \frac{1}{d_{far}}\right)\sigma\left((\mathbf{w} * \mathbf{Y})_{i,j}\right) + \frac{1}{d_{far}}\right)^{-1}. \quad (4)$$

The finest resolution that the decoder produces is $\frac{H}{2}\times\frac{W}{2}$. To acquire $\mathbf{Y}$ at full resolution, we first upsample the output encoding of the final decoder block together with the corresponding inverse depth map prediction and concatenate these with the input image. Then, we pass this tensor to two more convolutional layers.

Loss Function. For our loss function, we accumulate the average $L1$ error over the inverse depth maps at each output resolution considering only valid groundtruth values.

3.2. Spatio-temporal Fusion

We now present our framework that extends our pair network to incorporate knowledge about the past into the current prediction when processing video streams. In essence, we include a ConvLSTM cell to our network between the encoder and the decoder at the bottleneck to model the spatio-temporal relations, *c.f*. Fig. 3.

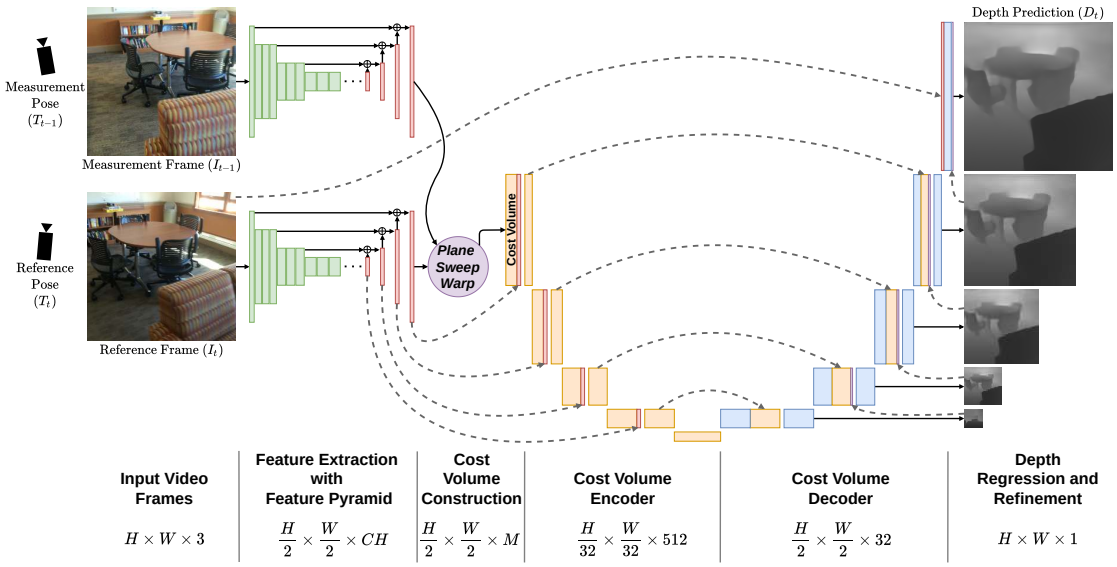

Figure 2: Sketch of our pair network that consists of five main parts. First is the shared feature extraction with feature pyramid network to acquire feature maps from input images. Second is the cost volume construction module that builds a 3D cost volume using the extracted features. Then, an encoder-decoder network regularizes the cost volume. We allow the encoder to take cues from extracted feature maps by placing skip connections from the feature extraction module. Finally, the model regresses depth maps at multiple resolutions including the input image resolution after a small refinement block.

Our ConvLSTM cell logic is based on [38], a variant of the original version [45]. Let $\mathbf{H}$ and $\mathbf{C}$ denote the hidden state and the cell state and $\mathbf{X}$ denote the output of the encoder at the bottleneck, then, the logic is written as

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{w}_{xi} * \mathbf{X}_t + \mathbf{w}_{hi} * \mathbf{H}_{t-1}\right) \\
\mathbf{f}_t &= \sigma\left(\mathbf{w}_{xf} * \mathbf{X}_t + \mathbf{w}_{hf} * \mathbf{H}_{t-1}\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{w}_{xo} * \mathbf{X}_t + \mathbf{w}_{ho} * \mathbf{H}_{t-1}\right) \\
\mathbf{g}_t &= \mathrm{ELU}\left(\mathrm{layernorm}(\mathbf{w}_{xg} * \mathbf{X}_t + \mathbf{w}_{hg} * \mathbf{H}_{t-1})\right) \\
\mathbf{C}_t &= \mathrm{layernorm}(\mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t) \\
\mathbf{H}_t &= \mathbf{o}_t \odot \mathrm{ELU}\left(\mathbf{C}_t\right), \qquad (5)
\end{aligned}$$

where $*$ denotes convolution, $\odot$ the Hadamard product, σ the sigmoid activation and $\mathbf{w}$ are learned convolution filter weights. We empirically found that ELU activation [9] leads to better results than $\tanh$, and Layer Normalization [4], without learnable parameters, stabilizes the model by preventing values from growing uncontrollably and enforcing zero mean / unit variance per channel in $\mathbf{C}$. We also observed that ConvGRU [5] cell, while being a resource-wise cheaper option, perform worse than ConvLSTM when used in our fusion scheme. Hence we opt for the latter. For analyses on the recurrent cell choice and the activation-normalization options, please refer to the supplementary material.

Naive Fusion. Ensuring that $\mathbf{H}$'s and $\mathbf{X}$'s dimensions are equal (so that the decoder of the pair network can be used without any modification), and letting $\mathbf{S}$ denote the skip connections from the encoder to the decoder, a simple model for the fusion could be written as

$$\begin{aligned}
\mathbf{X}_t, \mathbf{S}_t &= \mathrm{encoding}(\mathbf{I}_t,\ \mathbf{I}_{t-1},\ \mathbf{T}_t,\ \mathbf{T}_{t-1},\ \mathbf{K}) \\
\mathbf{H}_t, \mathbf{C}_t &= \mathrm{cell}(\mathbf{X}_t,\ \mathbf{H}_{t-1},\ \mathbf{C}_{t-1}) \\
\widehat{\mathbf{D}}_t &= \mathrm{decoding}(\mathbf{H}_t,\ \mathbf{S}_t). \qquad (6)
\end{aligned}$$

Note that (*c.f.* Eq. 5), $\mathbf{X}_t$, the output of the encoder at the bottleneck and the hidden state $\mathbf{H}_{t-1}$ are directly interacting with each other, while being encoded from different viewpoints. Under the assumption that $\mathbf{X}$ and $\mathbf{H}$ are purpose-wise similar latent representations, encapsulating both the visual (image features) and the geometric (cost encoding) information, Eq. 6 forces the ConvLSTM cell to capture the pose-induced image motion between the two encodings, which can be challenging due to large disparities in near objects, occlusions, rapid rotations, *etc*.

Proposed Fusion. Hence, we find it beneficial to partially account for the viewpoint changes while propagating the hidden state $\mathbf{H}_{t-1}$ to the next time step. Having access to the past and current camera poses and using the current reconstruction as a proxy, we warp the hidden representation $\mathbf{H}_{t-1}$ to the current viewpoint to acquire $\widetilde{\mathbf{H}}_{t-1}$. Such warping can be implemented as either forward or inverse mapping. The former is quite involved [37] and would require non-trivial visibility handling and differentiable rendering, with an additional impact on processing time. Using bilinear grid sampling [22], the latter is a fully differentiable and lightweight operation, and lets us keep the overall runtime low. To compute the sampling locations, we estimate a (partial) depth map $\widetilde{\mathbf{D}}_t$ for the current time step *before* starting the forward

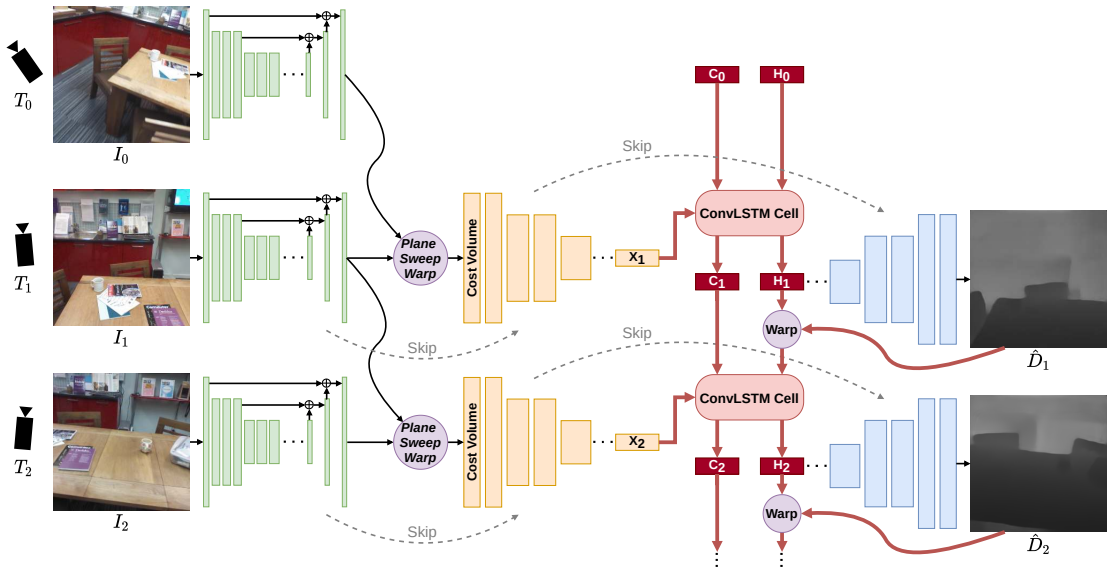

Figure 3: In our fusion approach, the pair network is extended with a ConvLSTM cell placed between the encoder and the decoder. The current frame and the previous frame(s) are used for computing a cost volume. The encoder takes the cost volume and produces a latent encoding at the bottleneck. The current latent encoding then participates in the convolutions in the ConvLSTM cell together with the *warped* hidden state coming from the previous time step forming a Markov chain. After the fusion, the new hidden state is passed through the decoder which outputs the depth predictions in the same way as the pair network.

pass. To that end, we project (handling occlusions) a small point cloud, which is estimated from the previous depth prediction $\hat{\mathbf{D}}_{t-1}$, onto the current camera plane. Then, we sample the hidden representation $\mathbf{H}_{t-1}$ to get $\widetilde{\mathbf{H}}_{t-1}$. Here, recall that the input resolution get spatially downscaled by $1/32$, while the channel size increases significantly at the bottleneck. This relaxes the need for perfect warpings as it enables the receptive fields in the ConvLSTM cell to perceive a considerable amount of information. Therefore, we transform the model into

$$\begin{aligned}
\widetilde{\mathbf{D}}_t &= \text{projection}(\hat{\mathbf{D}}_{t-1},\ \mathbf{T}_{t-1},\ \mathbf{T}_t,\ \mathbf{K}) \\
\widetilde{\mathbf{H}}_{t-1} &= \text{warping}(\mathbf{H}_{t-1},\ \widetilde{\mathbf{D}}_t) \\
\mathbf{X}_t, \mathbf{S}_t &= \text{encoding}(\mathbf{I}_t,\ \mathbf{I}_{t-1},\ \mathbf{T}_t,\ \mathbf{T}_{t-1},\ \mathbf{K}) \\
\mathbf{H}_t, \mathbf{C}_t &= \text{cell}(\mathbf{X}_t,\ \widetilde{\mathbf{H}}_{t-1},\ \mathbf{C}_{t-1}) \\
\hat{\mathbf{D}}_t &= \text{decoding}(\mathbf{H}_t,\ \mathbf{S}_t). \qquad (7)
\end{aligned}$$

We argue that our formulation results in an easier learning problem for the ConvLSTM cell, since Eq. 7 alleviates the need to capture the flow of the visual representations along with learning to fuse the encodings. We interpret this scheme as the ConvLSTM cell aggregating the prior geometric cost and the current geometric cost for the overlapping regions.

To set the training in motion and stabilize the behaviour, we use the groundtruth depth map $\mathbf{D}_t$ in place of $\widetilde{\mathbf{D}}_t$ in Eq. 7 and switch to the testing strategy only at a late stage, where we solely finetune the cell. This guides the ConvLSTM first with accurate warpings, and then allows it to adapt to the testing configuration slowly.

4. Experiments

Our model is implemented in PyTorch and trained using one NVIDIA GTX 1080Ti GPU, Intel i7-9700K CPU. The fusion network is trained with a mini-batch size of 4 and a subsequence length of 8. Input image size is 256×256 for which we crop the original image to a square, then scale. Additional technical details and the exact training procedure are provided in the supplementary. In summary, we first train the pair network independently and use the weights to partially initialize our fusion network. We start by training the cell and the decoder, which are randomly initialized, and then gradually unfreeze the other modules. Finally, we finetune only the cell while warping the hidden states with the predictions. To prevent overfitting of the regression part, we employ a variant of geometric scale augmentation [51], with a random geometric scale factor between 0.666 and 1.5.

4.1. Dataset

Training our spatio-temporal fusion network based on short-term memory demands longer input sequences. Therefore, we opt for the ScanNet [11] dataset's official training split to train and validate our models. For testing, we use ScanNet's 100 sequence official test split and a diverse collection of sequences, without large dynamic objects, from other datasets. We select 13 sequences from [16], 8 from [28], 13 from [47], and 4 from [18,48] (*c.f.* Tab. 1). Altogether, there are around 31K images in the test set. The official distribution of [16] does not supply aligned color and depth images, so we use the rendered depth maps provided by [7] for evaluation.

	MVDep	MVDep (FT)	DPSNet	DPSNet (FT)	DELTAS	Ours (Pair)	NRGBD	GPMVS	GPMVS (FT)	Ours (Fusion)
SCANNET										
abs	0.1953	0.1671	0.2185	0.1607	0.1492	0.1454	0.2361	0.2027	0.1491	**0.1187**
abs-rel	0.0970	0.0869	0.1192	0.0828	0.0783	0.0736	0.1220	0.1088	0.0762	**0.0599**
abs-inv	0.0621	0.0540	0.0710	0.0530	0.0506	0.0468	0.0745	0.0669	0.0490	**0.0381**
$\delta < 1.25$	0.8947	0.9252	0.8682	0.9254	0.9383	0.9459	0.8501	0.8893	0.9396	**0.9654**
7-SCENES										
abs	0.2029	0.2012	0.2486	0.1966	0.1911	0.1860	0.2143	0.1962	0.1737	**0.1448**
abs-rel	0.1157	0.1165	0.1486	0.1148	0.1139	0.1073	0.1312	0.1178	0.1003	**0.0829**
abs-inv	0.0732	0.0708	0.0847	0.0729	0.0717	0.0653	0.0756	0.0756	0.0641	**0.0537**
$\delta < 1.25$	0.8687	0.8768	0.8257	0.8708	0.8821	0.8936	0.8645	0.8723	0.9034	**0.9380**
RGB-D V2										
abs	0.1387	0.1310	0.1520	0.1320	0.1581	0.1433	**0.1227**	0.1576	0.1275	0.1256
abs-rel	0.0853	0.0846	0.1069	0.0835	0.1098	0.0891	0.0871	0.1029	0.0773	**0.0765**
abs-inv	0.0617	0.0637	0.0763	0.0623	0.0811	0.0667	0.0700	0.0761	**0.0559**	0.0566
$\delta < 1.25$	0.9417	0.9562	0.9057	0.9422	0.9137	0.9461	0.9373	0.9256	0.9575	**0.9707**
TUM RGB-D										
abs	0.2902*	0.3260	0.2958*	0.3045	0.3525	0.3535	0.3185	**0.2443***	0.2938	0.2878
abs-rel	0.1172*	0.1259	0.1335*	0.1184	0.1273	0.1247	0.1209	0.1038*	0.1103	**0.0975**
abs-inv	0.0643*	0.0715	0.0758*	0.0722	0.0753	0.0736	0.0681	0.0604*	0.0643	**0.0551**
$\delta < 1.25$	0.8597*	0.8304	0.8314*	0.8347	0.8177	0.8249	0.8402	**0.8892***	0.8563	0.8845
ICL-NUIM										
abs	**0.1392**	0.1574	0.1695	0.1491	0.1953	0.1771	0.1730	0.1667	0.1558	0.1496
abs-rel	**0.0581**	0.0637	0.0756	0.0612	0.0844	0.0723	0.0764	0.0709	0.0623	0.0587
abs-inv	0.0305	0.0328	0.0367	0.0359	0.0458	0.0402	0.0365	0.0356	0.0323	**0.0297**
$\delta < 1.25$	0.9468	0.9477	0.9348	0.9401	0.9192	0.9331	0.9318	0.9381	0.9442	**0.9571**

Table 1: Performance on: **i.** ScanNet test set, **ii.** 13 sequences from 7-Scenes , **iii.** 8 sequences from RGB-D Scenes V2, **iv.** 13 sequences from TUM RGB-D and **v.** 4 sequences from Augmented ICL-NUIM. Except Neural RGBD that uses 4 measurement frames, all evaluated methods use a *single* measurement frame. *FT* denotes finetuned on ScanNet. *Bold* is the best score, *overline* indicates the second best score. The vertical line separates video agnostic (*left*) from video aware (*right*) methods. * the method is already trained on most of the test frames.

4.2. Frame Selection

View selection, *i.e*. picking the right measurement frame for a reference frame, is an often overlooked but crucial aspect for balancing triangulation quality, matching accuracy, and view frustum overlap. To that end, we utilize the pose-distance measure proposed by [20]

$$dist\left[\mathbf{T}_{\text{rel}}\right] = \sqrt{\|\mathbf{t}_{\text{rel}}\|^2 + \frac{2}{3}\,\text{tr}\left(\mathbb{I} - \mathbf{R}_{\text{rel}}\right)}, \tag{8}$$

where $\mathbf{T}_{\text{rel}} = [\,\mathbf{R}_{\text{rel}} \mid \mathbf{t}_{\text{rel}}\,]$ denotes the relative pose between two cameras. We empirically found that a maximum pose-distance of 0.35 ± 0.05 and a translation of 10 cm $\pm$ 5 cm ensures sufficient baselines and image overlaps for indoor scenes, while being realistic for potential hand-held operation cases. For training, we augment our dataset further by sampling subsequences with thresholds from the prescribed range, where each consecutive frame obeys a given threshold. For testing, we simulate an online system that buffers the last 30 keyframes and updates the buffer frequently. If the pose-distance from the most recent keyframe is above 0.1, we add a new one, and rank the buffered keyframes w.r.t.

$$penalty(\mathbf{T}_{rel}) = \alpha(\|\mathbf{t}_{rel}\| - 0.15)^2 + \frac{2}{3}\,\text{tr}\left(\mathbb{I} - \mathbf{R}_{rel}\right)$$

$$\alpha = \begin{cases} 5.0 & \text{if } \|\mathbf{t}_{rel}\| \leq 0.15 \\ 1.0 & \text{if } \|\mathbf{t}_{rel}\| > 0.15, \end{cases} \tag{9}$$

which prefers relative camera distances of 15 cm to select the desired number of measurement frames.

4.3. Comparison with Existing Methods

We compare our method with five state-of-the-art, learning-based MVS approaches: MVDepthNet [51], GP-MVS [20] (in online mode of operation), DPSNet [21], Neural RGBD [32], and DELTAS [46]. Recall that MVDepthNet, DPSNet, DELTAS and our pair network do not exploit the sequential structure of the video input and only consider a given amount of frames at a time. In contrast, GP-MVS, Neural RGBD, and our proposed fusion approach are tailored for operating on video streams. We finetune MVDepthNet, GP-MVS and DPSNet on the ScanNet training set for up to 200K iterations and use the model with the best validation loss. We evaluate these both before and after finetuning.

Quantitative Evaluation. We use the following standard metrics [12,46] to acquire quantitative results: mean absolute depth error (abs), mean absolute relative depth error (abs-rel), mean absolute inverse depth error (abs-inv), and inlier ratio with threshold 1.25 ($\delta < 1.25$). Since most of our competitors are limited to a minimum depth of 0.5 meters, we do not consider the groundtruth measurements below this threshold for evaluation. For a fair comparison on full field of views, we run the inference for our models at 320×256 resolution without cropping. We acquire the predictions of each method at their native input resolutions by following their suggested scaling, then upsample the predictions with nearest neighbour interpolation to the original size (640×480) before calculating the metrics.

	1 Measurement Frame			2 Measurement Frames			3 Measurement Frames		
	abs	abs-rel	abs-inv	abs	abs-rel	abs-inv	abs	abs-rel	abs-inv
MVDepthNet	0.1770	0.0892	0.0548	0.1757	0.0877	0.0542	0.1739	0.0872	0.0544
GPMVS	0.1589	0.0786	0.0498	0.1588	0.0779	0.0496	0.1609	0.0800	0.0513
DELTAS	0.1659	0.0837	0.0533	0.1600	0.0798	0.0505	0.1562	0.0776	0.0492
Ours (Pair)	0.1614	0.0783	0.0492	0.1587	0.0757	0.0475	0.1537	0.0729	0.0457
Ours (Fusion)	**0.1318**	**0.0634**	**0.0397**	**0.1320**	**0.0619**	**0.0387**	**0.1298**	**0.0609**	**0.0381**

Table 2: Effect of using multiple measurement frames. Our view selection heuristic is used.

	Our Keyframe Buffer			Every 10th Frame			Every 20th Frame		
	abs	abs-rel	abs-inv	abs	abs-rel	abs-inv	abs	abs-rel	abs-inv
MVDepthNet	0.1757	0.0877	0.0542	0.1841	0.0940	0.0594	0.1865	0.0989	0.0628
GPMVS	0.1588	0.0779	0.0496	0.1682	0.0838	0.0545	0.1799	0.0942	0.0610
DELTAS	0.1600	0.0798	0.0505	0.1694	0.0858	0.0555	0.1712	0.0885	0.0574
Ours (Pair)	0.1587	0.0757	0.0475	0.1686	0.0814	0.0530	0.1583	0.0770	0.0500
Ours (Fusion)	**0.1320**	**0.0619**	**0.0387**	**0.1398**	**0.0663**	**0.0426**	**0.1376**	**0.0663**	**0.0429**

Table 3: Effect of applying our view selection heuristic. Two measurement frames are used.

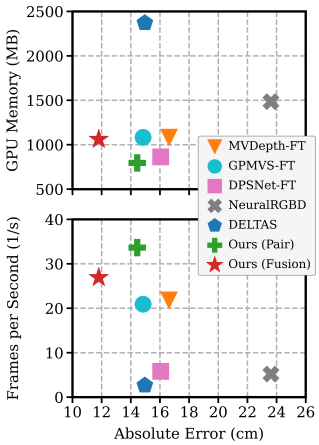

Figure 4: Speed and memory consumption in relation to depth prediction performance on ScanNet.

Tab. 1 summarizes our results. Our pair network performs on par with state-of-the-art competitors. With spatio-temporal fusion, we outperform the existing methods in 70% of all the metrics. Notably, our spatio-temporal fusion approach steadily improves the performance of the backbone across all test sets. For instance, when averaged over all test sets, we get 19.3% improvement in absolute inverse depth error, *c.f*. Tab. 2. In comparison, the finetuned GP-MVS improves over its MVDepthNet backbone by only 9.1%. Note that, both fusion strategies can extend many other existing approaches to leverage the past information. Being trained on ScanNet's official training split, DELTAS performs well on the ScanNet test sequences, but their performance drops below many of the other methods on the rest of the test sets. In contrast, our fusion approach generalizes well and delivers across the board. Even on TUM RGB-D, it is competitive against methods that are already trained on most of the *test* frames. Overall, our fusion model outperforms the best competitor, finetuned GP-MVS by a large margin, *e.g*. by 20.3% in absolute inverse depth error (*c.f*. Tab. 2).

Qualitative Evaluation. Fig. 5 compares the methods qualitatively. Reconstructions of Neural RGBD appear noisy and blurry. MVDepthNet and DPSNet suffer from prevalent gridding artifacts, visible in (b), (c) and (g). Being based on MVDepthNet, GP-MVS cannot completely eliminate the artifact, *e.g*. visible in (c). The depth maps of DELTAS are visually pleasant but blurry around edges at depth discontinuities. Overall, our fusion approach appears to be the one that is closest to the groundtruth and visibly improves upon our pair network. For instance, the corner of the kitchen in (a), the shelves in (c), arm of the sofa in (e), or the far wall in (g) are assigned more coherent depth values.

Fig. 6 shows TSDF reconstructions, acquired using the toolbox from [56], from ScanNet and 7-Scenes. The increased consistency among the depth predictions of our fusion network result in less noisy reconstructions, the walls appear flat and geometry of the scene is preserved the best. For instance, only our method allows to identify the armchair in the bottom row.

	abs	abs-rel	abs-inv
Pair Network	0.1614	0.0783	0.0492
Fusion without warping (Eq. 6)	0.1495	0.0697	0.0436
Fusion with warping (Eq. 7)	**0.1318**	**0.0634**	**0.0397**

Table 4: Effect of the proposed hidden state propagation scheme.

Runtime and Memory. Fig. 4 relates the absolute error to the memory consumption and average runtime of a single forward pass, measured on NVIDIA GTX 1080Ti. Exact numbers are provided in the supplementary. Our methods are the fastest and most accurate, especially our fusion approach delivers unmatched prediction quality at a high frame rate.

4.4. Ablation Studies

For our ablation studies, we average the performance over all test sequences. We present only the most important results here and more studies can be found in the supplementary.

Propagating the Hidden State by Warping. Tab. 4 shows that already the naive fusion scheme achieves on average about 10% improvement over the pair network, while the proposed propagation scheme delivers a similar gain on top.

Number of Measurement Frames. MVS methods typically use several measurement frames to gain robustness and coverage at the expense of time and memory. Our fusion approach is orthogonal to this idea, and one can similarly construct and average multiple cost volumes. As Tab. 2 shows, additional views tend to improve the performance, but it can stagnate after a number of measurement frames. While DELTAS and our models steadily benefit from more measurements, the abs-inv error of MVDepthNet and GP-MVS, for instance, increases at three views.

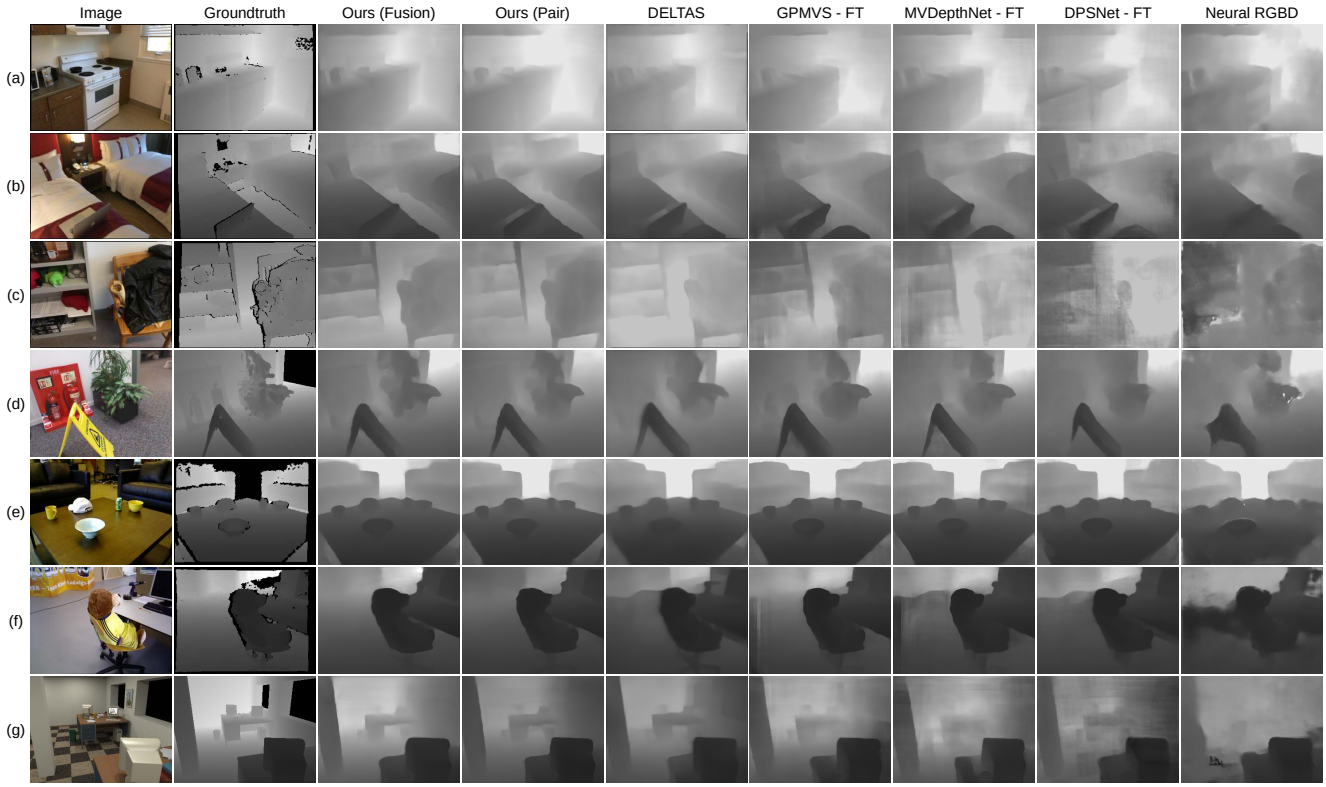

Figure 5: Example depth predictions from all test sets. Examples (a), (b) and (c) are taken from ScanNet, (d) from 7-Scenes, (e) from RGB-D Scenes V2, (f) from TUM RGB-D SLAM, and (g) from Augmented ICL-NUIM.

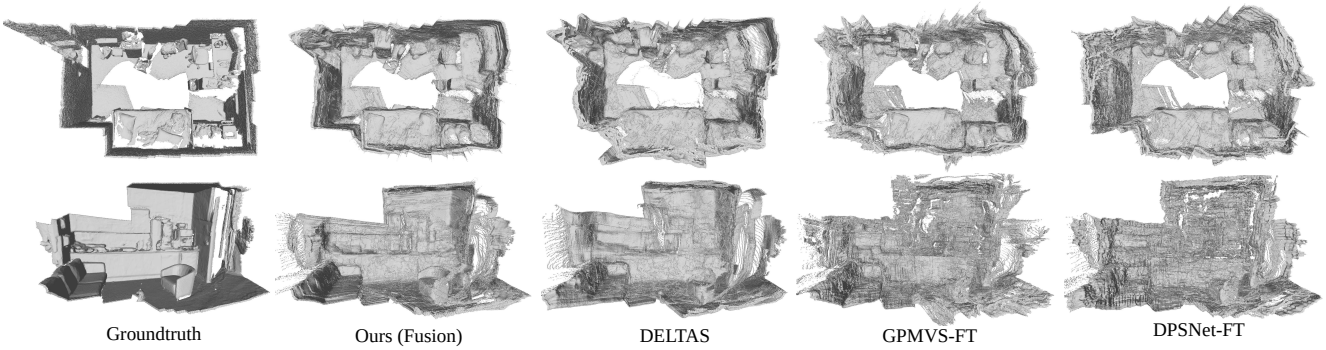

Figure 6: TSDF reconstructions from ScanNet (top) and 7-Scenes (bottom). Our fusion approach produces less noisy and geometrically more consistent depth predictions that result in accurate 3D reconstructions, *e.g.* straight walls, perpendicular corners, armchair, sofa.

Frame Selection. Tab. 3 shows that even a simple view selection heuristic enables consistently better predictions, compared to the common sampling rates (about 10% on average). Note that, our frame selection heuristic affects all methods in a similar manner. Still, all evaluated MVS methods, including ours, are robust enough to also work under naive sampling of every 10^{th} or 20^{th} frame in a video.

5. Conclusion

In this work, we tackle the problem of predicting depth maps from posed-video streams. Our approach exploits the temporally structured input and can be operated as an online multi-view stereo system, while estimating very accurate depth maps in real-time. Starting from a lightweight stereo backbone, we integrate a memory cell that acts as a fusion module and agglomerates the information obtained within the bottleneck encodings of our backbone over time. To remedy the viewpoint dependence of the fused encodings, we explicitly transform the hidden state of the ConvLSTM cell, while propagating it through time. Our proposed approach outperforms the existing state-of-the-art methods in most of the evaluation metrics and generalizes well to all considered test sets. We also achieve a significantly faster inference time than all competitors, while keeping a low memory consumption.

References

[1] Inside Facebook Reality Labs: Research updates and the future of social connection. https://tech.fb.com/inside-facebook-reality-labs-research-updates-and-the-future-of-social-connection/, 2019.

[2] Rony Abovitz. What Is The Magicverse (And Why)? https://www.magicleap.com/en-us/news/op-ed/magicverse, 2019.

[3] Ibraheem Alhashim and Peter Wonka. High Quality Monocular Depth Estimation via Transfer Learning. *arXiv:1812.11941 [cs]*, 2019.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv:1607.06450 [cs, stat]*, 2016.

[5] Nicolas Ballas, Li Yao, Chris Pal, and Aaron C. Courville. Delving deeper into convolutional networks for learning video representations. In *International Conference on Learning Representations (ICLR)*, 2016.

[6] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 2017.

[7] E. Brachmann and C. Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4654–4662, 2018.

[8] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-Based Multi-View Stereo Network. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[9] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *International Conference on Learning Representations (ICLR)*, 2016.

[10] R. T. Collins. A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 358–363, 1996.

[11] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017.

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. *arXiv:1406.2283 [cs]*, 3:2366–2374, 2014.

[13] David Fofi, Tadeusz Sliwa, and Yvon Voisin. A comparative survey on invisible structured light. In *Machine Vision Applications in Industrial Inspection XII*, 2004.

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2002–2011, 2018.

[15] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[16] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time RGB-D camera relocalization. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *International Conference on Neural Information Processing Systems - Volume 2*, page 2672–2680, 2014.

[18] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[19] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Horaud. *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer Publishing Company, Incorporated, 2012.

[20] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-View Stereo by Temporal Nonparametric Fusion. *International Conference on Computer Vision (ICCV)*, 2019.

[21] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. DPSNet: End-to-end Deep Plane Sweep Stereo. In *International Conference on Learning Representations (ICLR)*, 2019.

[22] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *International Conference on Neural Information Processing Systems - Volume 2*, page 2017–2025, 2016.

[23] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfaceNet: An End-to-end 3D Neural Network for Multiview Stereopsis. *International Conference on Computer Vision (ICCV)*, 2017.

[24] Neena Kamath. Announcing Azure Spatial Anchors for collaborative, cross-platform mixed reality apps. https://azure.microsoft.com/en-us/blog/announcing-azure-spatial-anchors-for-collaborative-cross-platform-mixed-reality-apps/, 2019.

[25] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a Multi-View Stereo Machine. In *International Conference on Neural Information Processing Systems*, page 364–375, 2017.

[26] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-End Learning of Geometry and Context for Deep Stereo Regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[27] Hamid Laga. A Survey on Deep Learning Architectures for Image-based Depth Reconstruction. *arXiv:1906.06113 [cs, eess]*, 2019.

[28] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3D scene labeling. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[29] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From Big to Small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation. *arXiv:1907.10326 [cs]*, 2020.

[30] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the Loss Landscape of Neural Nets. In *Neural Information Processing Systems*, 2018.

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[32] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa Narasimhan, and Jan Kautz. Neural RGB->D Sensing: Depth and Uncertainty from a Video Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] Fangchang Ma and Sertac Karaman. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4796–4803, 2018.

[34] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[35] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision(ECCV)*, 2020.

[36] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-End 3D Scene Reconstruction from Posed Images. In *European Conference on Computer Vision (ECCV)*, 2020.

[37] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] Andrea Palazzi. ConvLSTM_pytorch. https://github.com/ndrplz/ConvLSTM_pytorch, 2020.

[39] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't Forget The Past: Recurrent Depth Estimation from Monocular Video. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[40] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[41] Thinal Raj, Fazida Hashim, Aqilah Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on liDAR scanning mechanisms. *Electronics*, 2020.

[42] Tilman Reinhardt. Google Visual Positioning Service. https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html, 2019.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[44] Davide Scaramuzza and Zichao Zhang. *Visual-Inertial Odometry of Aerial Robots*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2020.

[45] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *International Conference on Neural Information Processing Systems*, page 802–810, 2015.

[46] Ayan Sinha, Zak Murez, James Bartolozzi, Vijay Badrinarayanan, and Andrew Rabinovich. DELTAS: Depth Estimation by Learning Triangulation And densification of Sparse points. In *European Conference on Computer Vision (ECCV)*, 2020.

[47] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[48] Sungjoon Choi, Q. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[49] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2815–2823, 2019.

[50] Denis Tananaev, Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Temporally Consistent Depth Estimation in Videos with Recurrent Architectures. In *European Conference on Computer Vision (ECCV)*, 2019.

[51] Kaixuan Wang and Shaojie Shen. MVDepthNet: Real-time Multiview Depth Estimation Neural Network. In *International Conference on 3D Vision (3DV)*, pages 248–257, 2018.

[52] Rui Wang, Stephen M. Pizer, and Jan-Michael Frahm. Recurrent Neural Network for (Un-)supervised Learning of Monocular VideoVisual Odometry and Depth. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[53] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth Inference for Unstructured Multi-view Stereo. *European Conference on Computer Vision (ECCV)*, 2018.

[54] Kyle Yee and Ayan Chakrabarti. Fast Deep Stereo with 2D Convolutional Processing of Cost Signatures. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 183–191, 2020.

[55] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5683–5692, 2019.

[56] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[57] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[58] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1725–1734, 2019.

[59] Yiran Zhong, Hongdong Li, and Yuchao Dai. Open-world stereo video matching with deep rnn. In *European Conference on Computer Vision (ECCV)*, September 2018.