Temporally Consistent Online Depth Estimation Using Point-Based Fusion

Numair Khan
numairkhan@meta.com

Eric Penner
epenner@meta.com

Douglas Lanman
douglas.lanman@meta.com

Lei Xiao
lei.xiao@meta.com

Abstract

Depth estimation is an important step in many computer vision problems such as 3D reconstruction, novel view synthesis, and computational photography. Most existing work focuses on depth estimation from single frames. When applied to videos, the result lacks temporal consistency, showing flickering and swimming artifacts. In this paper we aim to estimate temporally consistent depth maps of video streams in an online setting. This is a difficult problem as future frames are not available and the method must choose between enforcing consistency and correcting errors from previous estimations. The presence of dynamic objects further complicates the problem. We propose to address these challenges by using a global point cloud that is dynamically updated each frame, along with a learned fusion approach in image space. Our approach encourages consistency while simultaneously allowing updates to handle errors and dynamic objects. Qualitative and quantitative results show that our method achieves state-of-the-art quality for consistent video depth estimation.

1. Introduction

Depth reconstruction is a long-standing, fundamental problem in computer vision. For decades the most popular depth estimation techniques were based on stereo matching [37] or structure-from-motion [36]. However, more recently the best results have come from learning-based approaches [7]. As the overall reconstruction quality has improved, focus has shifted to new areas such as monocular estimation [21,32,33], edge quality [58], and temporal consistency [20, 26]. The latter is particularly important for video applications in computational photography and virtual reality [3, 52] as inconsistent depth can cause objectionable flickering and swimming artifacts.

Consistent video depth estimation, however, remains a difficult problem as even the best method will suffer from unpredictable errors and imperfections based on scene content, especially in textureless and specular regions. This difficulty is aggravated by many of the aforementioned applications requiring online reconstruction: future frames are
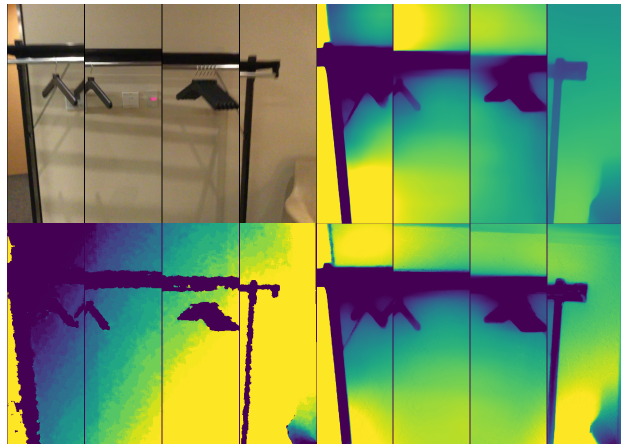

Figure 1. Comparing monocular depth estimation on the ScanNet dataset across four frames. *Clockwise from Top-left:* Input RGB image, Ranftl *et al.*'s DPT [32], our result with a DPT backbone, RGB-D sensor ground truth.

not known beforehand and temporal consistency must be balanced with error-correction. Furthermore, the presence of dynamic objects — which are inherently inconsistent — adds an additional layer of complication. Luo *et al.* [26] address these problems by assuming all frames are known beforehand and fine-tuning their method for each input video. Other approaches seek an online solution by encoding consistency in network weights either through a training loss [21], by using recurrent architectures [8, 30, 56], or by conditioning it on the input [24]. Each method, however, ultimately relies on the raw output of a neural network being consistent, which is difficult to achieve due to camera noise and aliasing in convolutional networks [14, 44, 57].

In this work, we propose the use of a global point cloud to encourage temporal consistency in online video depth estimation. We demonstrate how to tackle the twin problems of handling dynamic objects and updating a static — and potentially erroneous — point cloud when future frames are not known. With quantitative and qualitative results, we show our approach significantly improves the temporal consistency of both stereo and monocular depth estimation without sacrificing spatial quality. In summary, our contributions are as follows:
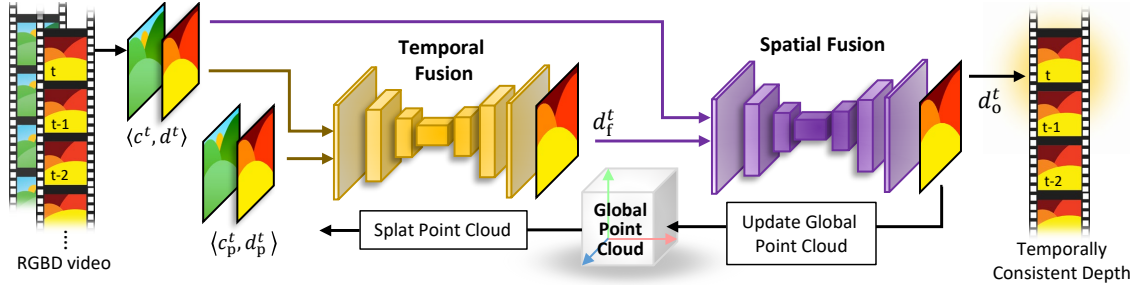
Figure 2. We generate temporally consistent depth maps for each RGB video frame t by fusing the projected depth from a prior point cloud d_{p}^t with the estimated depth d^t. This is done by temporally fusing d_{p}^t to update dynamic regions, followed by spatial fusion with d^t based on confidence estimates of accuracy. The final result d_{o}^t is used to update the point cloud for the next frame.

1. We propose point cloud-based fusion for temporally consistent video depth estimation.

2. We present a three-stage approach to encourage consistency in online settings, while simultaneously allowing updates to improve the accuracy of reconstruction and handle dynamic scenes.

3. We present an image-space approach to dynamics estimation and depth fusion that is lightweight and has low runtime overhead.

2. Related Work

Depth Estimation: The recovery of geometry from color images is an historically important problem in computer vision. In the last decade, deep learning has increasingly been applied to different stages of classic algorithms including geometric [4, 23, 39, 53], photometric [54] and multi-view stereo [12], as well as structure-from-motion (SfM) [50]. More recently, approaches based entirely on learnt priors have allowed high-quality depth estimation from single images [21, 32, 33, 45]. Traditionally, the focus of these methods has been on single frames, with most suffering from temporal artifacts on videos.

Temporal Consistency: A number of works have recently demonstrated the advantage of using temporal information for improving depth accuracy and consistency [24, 25, 30]. Long *et al.* [25] propose temporal coherence by fusing the cost volumes of neighboring frames with a transformer. Similarly, Liu *et al.*'s [24] method accumulates depth probability over consecutive frames using Bayesian filtering. However, the final outcome in both cases is generated by a refinement network which does not have any explicit temporal constraints. Luo *et al.* [26] fine-tune a monocular depth network at inference-time using constraints from optical flow and structure-from-motion. Kopf *et al.* [17] build on this to refine camera poses too. Both methods run offline with the inference-time fine-tuning taking around twenty minutes.

Luo *et al.* and Kopf *et al.* are special instances of a more general approach to enforcing consistency in video tasks using an optical flow-based warping loss during training [2, 18, 20]. Used in isolation, however, this approach fails to generalize well to unseen data, often requiring fine-tuning for each new dataset. But it does have the advantage of not relying on camera poses or external structures. Lai *et al.* [18] and Cao *et al.* [2] pair a warping loss with a convolutional recurrent module which allows the network to learn temporal affinities more effectively.

In fact, the use of recurrent structures is another way of enforcing temporal consistency in video tasks [8, 30, 56]. Zhang *et al.* [56] propagate spatio-temporal depth information across frames using a novel convolutional LSTM module which is trained using an adversarial loss. While this loss does not have any explicit temporal constraint like the warping loss, it is nonetheless shown to improve depth consistency. Patil *et al.*'s [30] work on depth prediction and completion demonstrates that recurrent modules enforce consistency and improve accuracy even without special loss functions. Convolutional LSTMs, however, can be difficult to train due to their large memory requirements.

Depth Fusion: Fusion methods [5, 29, 48, 49] achieve scene reconstruction by blending weighted signed distance field (SDF) volumes for each frame. The use of SDF volumes makes them scale poorly to large scenes and high resolutions. Keller *et al.* [15] and Lefloch *et al.* [19] propose point clouds as an alternative to SDFs. Traditional fusion methods, however, are not directly applicable to depth reconstruction as their focus is geometric reconstruction over multiple frames. Completeness, in the sense of reconstructing each visible point in a frame, is not a requirement.

3. Consistent Video Depth Estimation

Given a sequence of RGB images c^t with known poses, our goal is to estimate a depth map d^t for each c^t, $t \in \{0, 1, 2, ...\}$ such that the geometric representation of all scene points visible at time t is accurate and consistent from d^{t-1} to d^t. Furthermore, we want to solve this problem online; that is, at any time instant t future frames d^i, $i > t$ are not known. Accuracy subsumes consistency since a 100% accurate reconstruction is also 100% consistent, but
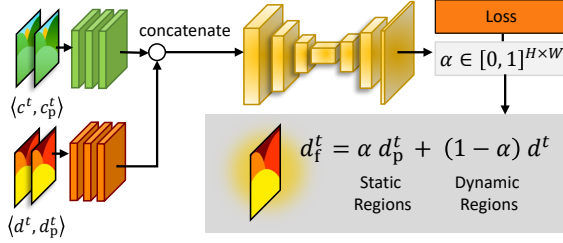
Figure 3. The temporal fusion stage generates a 2D mask α for scene regions that change due to motion in frame t. This mask is used to update the reprojected point cloud depth d_{p}^t with dynamic changes from the observed depth d^t.

not vice versa. As a result, the majority of past depth estimation methods have focused on optimizing accuracy of reconstruction [4,23,39,53]. In practice, however, a perfectly accurate reconstruction is hardly ever achievable such that most of these methods suffer from inconsistency.

A simple way of enforcing temporal consistency is to use the known camera pose to elevate a depth map d^t to a 3D point cloud and then reproject it into future frames. However, such a reprojection-based approach suffers from the following shortcomings: it fails to handle dynamic scenes, it creates holes in disoccluded regions, and it will propagate any errors in d^t to all subsequent frames.

Our proposed approach seeks to exploit the advantages of a global point cloud while addressing the above-mentioned shortcomings. We do this in three steps: (i) The temporal fusion step (Sec. 3.1) identifies and updates regions of the scene that changed from frame $t-1$ to t by blending d^t with prior depth from the global point cloud in image space. (ii) Spatial fusion (Sec. 3.2) recovers spatial details and corrects any errors in the blended output depth map from the first stage. (iii) Finally, we update the global point cloud to incorporate changes from frame t (Sec. 3.3).

3.1. Temporal Fusion

Let $\mathcal{P} \in \mathbb{R}^3$ be a prior 3D point cloud representation of a scene at time t generated using the depth input from previous frames. For each $\mathbf{x} \in \mathcal{P}$ let $\varsigma(\mathbf{x}) \in \mathbb{R}^3$ represent the RGB color, and $\rho(\mathbf{x}) \in \mathbb{R}$ represent a confidence estimate of the point. Given the camera pose $\mathbf{T}_t$ and intrinsics matrix $\mathbf{K}$, we define d_{p}^t, c_{p}^t and w_{p}^t as the $H \times W$ depth, color and confidence projection, respectively, of $\mathcal{P}$.

The goal of the temporal fusion stage is to generate a 2D mask $\alpha \in [0,1]^{H \times W}$ for the dynamic regions in d_{p}^t. Specifically, we want α to identify visible points that change position from frame $t-1$ to t solely due to motion, and not because of camera movement or uncertainty in the depth estimation. We then compute a fused depth map d_{f}^t with the dynamic parts updated from d^t while preserving consistent depths from prior frames in the static regions:

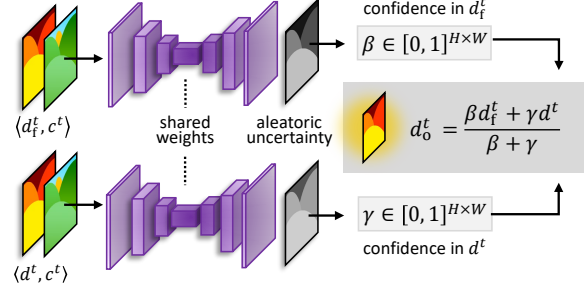$$d_{\mathrm{f}}^t = \alpha\, d^t + (1 - \alpha)\, d_{\mathrm{p}}^t \qquad (1)$$

Figure 4. The spatial fusion stage uses the observed depth d^t to improve static regions in the temporally fused depth d_{f}^t based on confidence in the accuracy of both depths. The confidence is estimated as the learnt uncertainty in the depth prediction model.
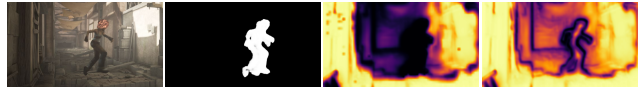

Figure 5. *Left to right*: The input c^t, the dynamic change mask α (Eq. 1), the confidence estimate β (Eq. 4), and the confidence estimate γ (Eq. 3). Original image © Blender Foundation | www.sintel.org

The blending mask α may be computed as the residual between consecutive frames warped using rigid flow. We observe, however, that a hand-crafted threshold for the residual does not generalize to all scenes, and is prone to failure in the presence of noise and depth artifacts. Thus, we use $\alpha = \Theta(d^t, d_{\mathrm{p}}^t, c^t, c_{\mathrm{p}}^t)$ instead, where $\Theta(\cdot)$ is a convolutional neural network (Fig. 3). We want α to have a strong bias towards the prior depth map in static regions. This is encouraged by using the ground truth — rather than the generated — prior point cloud $\mathcal{P}$ to render c_{p}^t and d_{p}^t during training. Using an L1 loss on the blended output d_{f}^t then ensures that $\alpha = 1$ minimizes the loss in scene regions that do not change from $t-1$ to t. As observed earlier, training $\Theta(\cdot)$ to directly output d_{f}^t does not guarantee consistency due to CNN aliasing and rotational invariance.

For the very first frame $t = 0$, we use $d_{\mathrm{p}}^t = d^t$. Subsequently, we render the point cloud $\mathcal{P}$ by splatting each $\mathbf{x} \in \mathcal{P}$, along with color $\varsigma(\mathbf{x})$ and confidence $\rho(\mathbf{x})$, to the nearest rounded pixel with a Z-buffer check. While this introduces aliasing artifacts, it is much faster than surfel-based methods [31]. We ameliorate the aliasing through super-sampling, and we use Rosenthal and Linsens's method [35] to fill holes and remove background points.

3.2. Spatial Fusion

Using the ground truth to generate d_{p}^t allows for easily training the temporal fusion stage to maintain consistency in static regions. However, the ground truth is not available at inference-time so that the prior point cloud $\mathcal{P}$ — and consequently d_{p}^t — is likely to have errors from imperfect depth estimation in previous frames. We propose to address this problem in the spatial fusion stage.

The goal here is to assign a pixel-wise weight to d^t and d_f^t based on a confidence estimate of the accuracy of each depth. This is then used to fuse the two depths in image space to generate the final output d_o^t:

$$d_o^t = \frac{\beta \, d_f^t + \gamma \, d^t}{\beta + \gamma} \tag{2}$$

We estimate the confidence weights $\beta, \gamma \in \mathbb{R}^{H \times W}$ using a neural network $\Phi(\cdot)$ (Fig. 4). Note that while formulated similar to Eq. 1, the weights in Eq. 2 serve a different purpose and, hence, require a different training procedure for $\Phi(\cdot)$. In particular, we represent the weights as the input-dependent aleatoric uncertainty [16] which can be learnt in a self-supervised manner. The confidence γ for d^t is then inversely related to the uncertainty as,

$$\gamma = \exp\big[-\Phi(d^t, c^t)\big] \tag{3}$$

The confidence β for d_f^t, in addition to the aleatoric uncertainty $\Phi(d_f^t)$, also incorporates the prior confidence w_p^t weighted by the temporal mask α:

$$\beta = \alpha \, w_p^t \exp\big[-\Phi(d_f^t, c^t)\big] \tag{4}$$

We box-filter w_p^t before computing β to suppress any high-frequency artifacts from the splatting process. Note that $\Phi(\cdot)$ can model uncertainty in d_f^t resulting from errors in temporal filtering, so that Eq. 2 is not the same as α-blending d^t with $w_p^t \exp\big[-\Phi(d_p^t, c^t)\big] d_p^t$.

3.3. Global Point Cloud Update

In a final step, we update the global point cloud $\mathcal{P} \subseteq \mathbb{R}^3$ to reflect changes in the scene from frame t. This includes updating the confidence $\rho(\mathbf{x})$, and the color $\varsigma(\mathbf{x})$ of all points $\mathbf{x} \in \mathcal{P}$ observed at t. This is similar to the TSDF volume integration step in traditional fusion methods [13, 29]. However, here we have a discrete point cloud representation of the scene which cannot be directly averaged like a TSDF. Therefore, achieving a similar effect involves: (i) associating points in $\mathcal{P}$ with pixels in d_o^t, and updating the position, confidence and color of points for which a correspondence is found; (ii) adding pixels from d^t for which no association is found as newly observed points into $\mathcal{P}$; and (iii) removing points with low confidence from $\mathcal{P}$.

Let $z^t \in \mathbb{R}^{H \times W \times 3}$ represent the inverse projection of depth map d^t to 3D. Updates to each $\mathbf{x} \in \mathcal{P}$ are computed in continuous image space to minimize quantization artifacts from using a point cloud. Thus, we update a given $\mathbf{x}$ by sampling z^t and the color image c^t at the projected 2D location $\hat{\mathbf{x}}$ of $\mathbf{x}$ on the image plane. We denote the sampled values as $c^t[\hat{\mathbf{x}}] \in \mathbb{R}^3$ and $z^t[\hat{\mathbf{x}}] \in \mathbb{R}^3$, respectively, using square brackets to represent bilinear sampling of the image at the indexed location. We only update unoccluded points

identified by $\alpha[\hat{\mathbf{x}}] < 0.5$, where α is from Eq. 1. The updated $\mathbf{x}$, color $\varsigma(\mathbf{x})$ and confidence $\rho(\mathbf{x})$ are computed as,

$$\mathbf{x} = \frac{\beta[\hat{\mathbf{x}}] \cdot \mathbf{x} + \gamma[\hat{\mathbf{x}}] \cdot z^t[\hat{\mathbf{x}}]}{\beta[\hat{\mathbf{x}}] + \gamma[\hat{\mathbf{x}}]} \tag{5}$$

$$\varsigma(\mathbf{x}) = \frac{\beta[\hat{\mathbf{x}}] \cdot \varsigma(\mathbf{x}) + \gamma[\hat{\mathbf{x}}] \cdot c^t[\hat{\mathbf{x}}]}{\beta[\hat{\mathbf{x}}] + \gamma[\hat{\mathbf{x}}]} \tag{6}$$

$$\rho(\mathbf{x}) = \beta[\hat{\mathbf{x}}] + \gamma[\hat{\mathbf{x}}] \tag{7}$$

where $\beta, \gamma \in \mathbb{R}^{H \times W}$ are the fusion weights from Eqs. 3 and 4. For any $\mathbf{x}$ outside the field-of-view, or which is occluded, we set $\rho(\mathbf{x}) = \rho(\mathbf{x}) - 1$. We identify newly observed points as pixel locations in z^t for which $\alpha \geq 0.5$, and add them to $\mathcal{P}$. Finally, we remove all points $\mathbf{x} \in \mathcal{P}$ for which $\rho(\mathbf{x}) < \epsilon$. We find $\epsilon = 3 \times 10^{-2}$ to work well empirically.

4. Training Procedure

We train the temporal fusion network $\Theta(\cdot)$ using samples (d^t, c^t, d_p^t, c_p^t) where (c^t, d^t) are the color image and estimated depth of the current frame t, and (d_p^t, c_p^t) are generated by reprojecting the ground truth depth and color from frame $t - k$ to t, for some arbitrary k between -7 and 7. We use future frames ($k < 0$) only during training as a data augmentation technique — during inference, our method only sees the current frame t and the global point cloud.

$\Theta(\cdot)$ consists of two sets of three residual layers followed by a four-layer U-Net [34]. The residual layers extract features from the concatenated depths $\langle d_p^t, d^t \rangle$ and concatenated color images $\langle c_p^t, c^t \rangle$ which are then fed into the U-Net to generate the blending mask α (Fig. 3). This network architecture is much simpler and has fewer parameters than either Mask-RCNN [11] or optical flow methods, both of which have been previously used [22, 26, 55] to mask dynamic elements of a scene (Table 5). We use ReLU activations on all but the last layer of the U-Net, which has a sigmoid. As batch normalization adds a depth-dependent bias [48], we use layer normalization instead.

The motivation behind using the ground truth rather than the estimated depth for d_p^t is to encourage a strong bias towards the prior depth ($\alpha = 0$) in static regions. This ensures $\Theta(\cdot)$ only identifies changes due to motion. The reprojected ground truth remains optimal in regions that do not change from frame $t - k$ to t. Thus, an L1 loss on d_f^t encourages $\alpha = 0$ in these parts. For dynamic objects, the prior depth may or may not be optimal based on the quality of the estimated depth, and the network learns a suitable blending weight. We found a binary cross entropy (BCE) loss on α improves convergence and pushes it closer to binary:

$$\mathrm{L_{BCE}} = \mathrm{BCE}\Big(\alpha, \mathbb{1}\big(\|d^t - g^t\|_1 < \|d_p^t - g^t\|_1\big)\Big) \tag{8}$$

where $\mathbb{1}(\cdot)$ is the indicator function of the pixels satisfying the inequality and g^t is the ground truth depth. We also

	OPW↓	SC↓	RTC↑	TCM↑	TCC↑	SD(L1)↓	RAE↓	RMS↓	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	N
MPI Sintel Dataset												
MiDaS [33]	0.611	0.675	0.211	0.233	0.413	0.597	0.279	3.013	0.610	0.798	0.896	1
WSVD [45][‡]	0.613	0.704	0.288	0.356	0.449	0.654	0.360	3.775	0.466	0.705	0.825	2
DPT [32]	0.424	0.493	0.320	0.417	0.482	0.539	0.224	2.678	0.686	0.864	0.932	1
Ours-DPT[†‡]	**0.255**	**0.295**	**0.489**	**0.470**	**0.559**	**0.474**	**0.197**	**2.400**	**0.710**	**0.885**	**0.942**	1
ScanNet Dataset												
ST-CLSTM [56][‡]	0.087	0.087	0.310	0.134	0.400	0.076	0.289	0.751	0.772	0.932	0.983	5
TCMD [20][‡]	0.031	0.031	0.628	0.166	0.548	0.083	0.300	0.811	0.740	0.927	0.981	1
N-RGB→D [24][†‡]	0.043	0.042	0.557	0.169	0.482	0.088	0.285	0.674	0.814	0.952	0.983	1
Demon [43]	0.160	0.159	0.143	0.107	0.271	0.098	0.228	0.651	0.839	0.951	0.982	2
DeepV2D [40]	0.038	0.038	0.483	0.143	0.495	0.058	**0.153**	**0.502**	0.957	0.995	**0.999**	2
DPT [32]	0.033	0.033	0.540	0.141	0.536	0.055	0.213	0.582	0.971	**0.996**	**0.999**	1
Ours-DPT[†‡]	**0.011**	**0.010**	**0.863**	**0.179**	**0.639**	**0.052**	0.210	0.576	**0.974**	**0.996**	**0.999**	1

[†]Method uses temporal data structure. [‡] Method has temporal consistency constraints.

Table 1. Comparing our method using a DPT backbone to SOTA monocular depth methods. N is the number of frames required at inference. Our method achieves the best results overall with significantly higher consistency, and similar or better accuracy.

include an L1 loss L_G on the depth gradients, and a VGG loss on the fused depth. We found the latter helps capture fine detail better than an L1 loss alone. Thus, we supervise the temporal fusion network with loss:

$$L_\Theta = \|d_f^t - g^t\|_1 + L_{BCE} + L_G + VGG(d_f^t, g^t) \quad (9)$$

We weigh the terms as 10, 0.1, 0.05 and 0.05, respectively. We implement the depth fusion network $\Phi(\cdot)$ as a four-layer U-Net with instance normalization, and ReLU activations on all layers. We train the network using a self-supervised loss on the uncertainty $s^t = \Phi(d^t, c^t)$ of the input depth:

$$L_\Phi = \frac{1}{m} \big\| \exp(-s^t)\,|\,d^t - g^t\,| + \lambda_s s^t \big\|_1 \quad (10)$$

where m is the number of pixels and the second term is a regularizer to prevent the trivial $s^t = \infty$. We use $\lambda_s = 0.03$. As Kendall *et al*. [16] show, this loss term minimizes the log variance of a Gaussian likelihood model for the uncertainty.

We train both networks on the FlyingThings3D dataset [27]. This consists of rendered sequences of objects moving along random 3D trajectories, and has ground truth depth and camera poses. For fine-tuning we render 50 60-frame sequences from a custom Blender scene with human-like motion in an indoor setting. We estimate d^t with RAFT-Stereo [23] and use the inverse depth for training.

5. Experiments

5.1. Implementation Details

Both networks are implemented in PyTorch and trained on six Nvidia Tesla V100 GPUs. The temporal fusion network is trained for five epochs on 500×500 crops from the
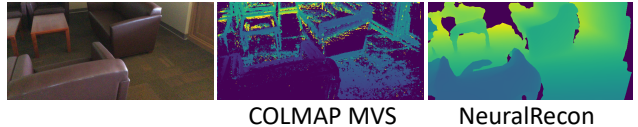

COLMAP MVS   NeuralRecon

Figure 6. MVS and fusion methods like NeuralRecon [38] suffer from holes in the output making them ill-suited to AR applications.

FlyingThings3D dataset in batches of 18. We use a learning rate of 5×10^{-4} and a weight decay of 1×10^{-4}. The spatial fusion network is trained on the depth output of the temporal filtering stage d_f^t using the FlyingThings3D dataset with a similar batch and input size, and a learning rate of 1×10^{-4}. We fine-tune both networks on 320×320 crops of our Blender data for 14 epochs at a learning rate of 1×10^{-6}.

5.2. Testing Datasets

MPI Sintel: The MPI Sintel dataset [1, 51] consists of long synthetic stereo sequences with large motion and depth range. We evaluate both stereo and monocular depth estimation on the final pass of the 22 "training scenes" which have ground truth optical flow, pose, and depth. Given the extremely large motion and depth of some scenes, we follow Teed *et al*. [42] and restrict our evaluation to pixels with flow ≤ 250 pixels and depth ≤ 30 meters.

ScanNet: The ScanNet dataset [9] consists of video sequences of static indoor scenes captured with an RGB-D sensor and annotated with camera poses. We evaluate the monocular variant of our method on 20 60-frame sequences from different scenes. ScanNet does not have ground truth optical flow, so we estimate it using RAFT [41].

MPI Sintel Dataset											
	OPW↓	SC↓	RTC↑	TCM↑	TCC↑	SD(L1)↓	RAE↓	RMS↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	N
MC [21]	0.553	0.608	0.233	0.423	0.402	0.619	0.371	3.771	0.462	0.690	0.820	1
CVD-MC [26]	0.226	0.245	0.402	0.515	0.460	0.589	0.338	3.078	0.493	0.702	0.819	All
	-59.1%	-59.7%	+72.5%	+21.7%	+14.4%	-4.85%	-8.89%	-18.4%	+6.71%	+1.74%	-0.12%	
Ours-MC	0.377	0.399	0.360	0.443	0.423	0.583	0.361	3.711	0.466	0.696	0.823	1
	-31.8%	-34.4%	+54.5%	+4.73%	+5.22%	-5.82%	-2.70%	-1.59%	+0.87%	+0.87%	+0.37%	

Table 2. We compare our approach to CVD using the Mannequin Challenge (MC) backbone for both methods. The percentage change in each metric over the baseline (MC) is listed in color. Our method realizes a large portion of the gains of CVD, even though CVD fine-tunes the backbone network for each input (∼20 mins) and requires all frames beforehand. Our method is trained once and runs online.

MPI Sintel Dataset											
	OPW↓	SC↓	RTC↑	TCM↑	TCC↑	SD(L1)↓	RAE↓	RMS↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
RAFT-Stereo [23]	0.409	0.507	0.615	0.424	0.642	0.795	0.239	31.26	**0.915**	**0.944**	**0.961**
Ours-RAFT-S.	0.268	0.340	**0.701**	0.492	0.674	0.745	0.227	21.86	**0.915**	0.943	0.959
	-34.5%	-32.9%	+14.0%	+16.0%	+4.98%	-6.29%	-5.02%	-30.1%	+0.00%	-0.11%	-0.21%
HITNet [39]	0.225	0.264	0.599	0.441	0.641	0.341	0.087	12.32	0.878	0.915	0.941
Ours-HITNet	**0.158**	**0.197**	0.675	**0.506**	**0.675**	**0.316**	**0.086**	**8.900**	0.878	0.916	0.943
	-29.8%	-25.4%	+12.7%	+14.7%	+5.30%	-7.33%	-1.15%	-27.8%	+0.00%	+0.11%	+0.21%

Table 3. Evaluating two SOTA stereo methods when used with our point cloud-based fusion approach. The percentage change is listed in color below our result for each metric. Our method leads to significant improvement in temporal consistency and preserves spatial quality.

5.3. Baselines

Our choice of baselines is motivated by the use cases we envision for an online method. These include low-latency AR/VR applications [3, 52] requiring temporal consistency, completeness (*i.e.* no holes), and the handling of dynamic objects. Hence, we evaluate stereo depth methods [23, 39] on the MPI Sintel dataset, and monocular methods [20, 24, 32, 33, 45, 56] on both the MPI Sintel and ScanNet datasets. Additionally, we evaluate learned structure-from-motion methods [40, 43]. We do not compare with traditional multi-view stereo or TSDF fusion methods as they suffer incompleteness, and fail in dynamic scenes (Fig. 6).

For each baseline, we use pre-trained models provided by the authors and evaluate it only on the dataset it has originally been tested on — with the only exception being models trained on the NYU-V2 dataset [28]. We evaluate these models on ScanNet instead (NYU-V2 is a similar sensor-captured RGB-D dataset, but does not have ground truth poses). We use the efficient *Hybrid* variant of Ranftl *et al.*'s DPT [32] in all experiments, and the five-layer *3D-GAN* version of Zhang *et al.*'s ST-CLSTM [56] .

5.4. Evaluation Metrics

Given a sequence of estimated depth maps $d^0, d^1, ..., d^n$ along with color images $c^0, c^1, ..., c^n$ and ground truth $g^0, g^1, ..., g^n$, we evaluate depth estimation quality both temporally and spatially. A common metric for tempo-

ral consistency of video uses the optical flow $O_{t \to t+1}$ to evaluate the change in a pixel's depth from frame t to $t+1$ [2, 18, 46]. Let d_w^{t+1} be the estimated depth d^{t+1} backward warped into frame t using $O_{t \to t+1}$. Then the optical flow-based warping error (OPW) is defined as:

$$\text{OPW} = \frac{1}{n-1} \sum_t \| M^t (d_w^{t+1} - d^t) \|_1 \qquad (11)$$

where $M^t = \exp(-\kappa(c_w^{t+1} - c^t))$ is a pixel-wise occlusion mask. Intuitively, OPW is the average meter change in the depth of a point across frames. It is expected to be higher for the outdoor scenes in MPI-Sintel than the indoor ScanNet dataset. Li *et al.* [20] present a variant of this error known as the relative temporal consistency (RTC) metric, which is shown to align with humans' perception of consistency:

$$\text{RTC} = \frac{1}{mn} \sum_t \left\| \mathbb{1}\left(M^t \max\left(\frac{d_w^{t+1}}{d_t}, \frac{d_t}{d_w^{t+1}}\right) < \tau \right) \right\|_0 \quad (12)$$

where m is the number of pixels, $\mathbb{1}(\cdot)$ is the indicator function of the set of pixels satisfying the enclosed inequality, and the 0-norm measures the number of non-zero pixel values. We set $\kappa = 50$ and $\tau = 1.01$ for our evaluation.

A disadvantage of OPW and RTC is that both metrics are optimized by the degenerate case $d^t = k \ \forall \ t \in \{0, 1, ..., n\}$ for any constant k. Hence we augment OPW and RTC with
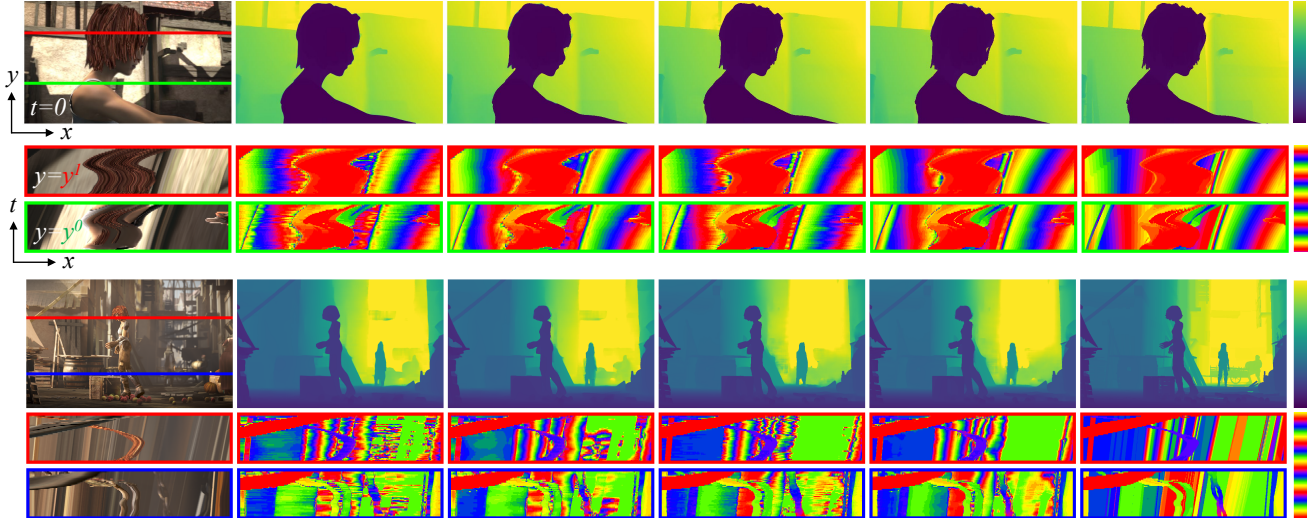
Figure 7. The improvement in consistency from our method, visualized as the variation in an epipolar plane image (EPI). An EPI represents a slice of the video in the x-t dimensions for a fixed y (indicated in green, red and blue). We use a high-gradient color-map to better visualize changes in the EPI. Our method reduces high-frequency oscillation along t indicating better frame-to-frame consistency, and preserves the spatial (x-y) quality of depth. We omit the axis labels in subsequent figures. Original images © Blender Foundation | www.sintel.org

the temporal change consistency (TCC) and the temporal motion consistency(TCM) metrics of Zhang *et al.* [56]:

$$\text{TCC} = \frac{1}{n-1} \sum_t \text{SSIM}\big(|d^t - d^{t+1}|, |g^t - g^{t+1}| \big) \quad (13)$$

$$\text{TCM} = \frac{1}{n-1} \sum_t \text{SSIM}(O(d^t, d^{t+1}), O(g^t, g^{t+1})) \quad (14)$$

where SSIM is the structural self-similarity [47], and O is the optical flow $O_{t \to t+1}$. In addition, we propose the self-consistency (SC) metric which uses the estimated depth d^t to compute d_w^{t+1} instead of the optical flow. Like OPW, it measures the average meter change in depth frame-to-frame. Finally, as in Long *et al.* [25] we measure the standard deviation of the L1 error SD(L1) across all frames.

For the spatial error we use the standard metrics: relative absolute error (RAE); root mean squared error (RMS); and the percentage of bad pixels δ_1, δ_2, δ_3 at thresholds 1.25, 1.25^2 and 1.25^3, respectively [56]. We align the affine-invariant monocular methods for evaluation by using the ground truth to calculate a scale and shift factor [33].

5.5. Results

Table 1 presents the quantitative evaluation of our method and the monocular baselines, with a qualitative comparison in Fig. 8. Across both datasets, our depth has a 50% lower OPW error than the next best method. Intuitively, this means for an average scene point, the frame-to-frame meter difference in depth is less than half that of the other method. Furthermore, the structural similarity of the temporal change in our depths and ground truth (TCC) is 15% higher, indicating that dynamic regions are not simply

	TCM↑	TCC↑	OPW↓	RAE↓
No Temporal Fusion	-11.4%	-12.1%	+27.7%	+65.7%
No Spatial Fusion	-7.44%	-9.06%	+39.6%	+17.1%
No Global PC	-7.45%	-9.57%	+39.5%	+13.0%

Table 4. Ablating our pipeline to evaluate its contribution to the final result. We list the percentage change with respect to **Ours-DPT** in Table 1 on the dynamic MPI Sintel dataset. For TCM and TCC higher is better, and vice versa for OPW and RAE.

	Params. (M)	GMACs	Runtime (ms)
RAFT (20 iters.)	5.26	309.5	86.2
Mask-RCNN	44.2	134.6	42.4
Our $\Theta(\cdot)$	4.45	35.04	7.20
Our $\Phi(\cdot)$	4.44	37.14	5.10

Table 5. Run-time and complexity evaluation of our two networks $\Theta(\cdot)$ and $\Phi(\cdot)$ with a 512×512 image on an NVIDIA GeForce RTX 3080 GPU. We use the default 20 iterations for RAFT.

averaged across frames. A similar trend is seen in the improved performance of the stereo methods in Table 3 with an improvement in OPW and TCC of around 30% and 5% respectively. Fig. 7 provides a qualitative visualization of the stereo results. The improvement from our method is compared to Luo *et al.*'s CVD [26] in Table 2. CVD runs offline, fine-tuning the backbone network for ∼20 minutes at inference-time. As such, it represents the high bar for temporal consistency. Our method achieves a good portion of the gains of CVD while being generalizable and online.

Table 5 lists the size and runtime of our two networks, comparing $\Theta(\cdot)$ to RAFT [41] and Mask-RCNN [11] — the
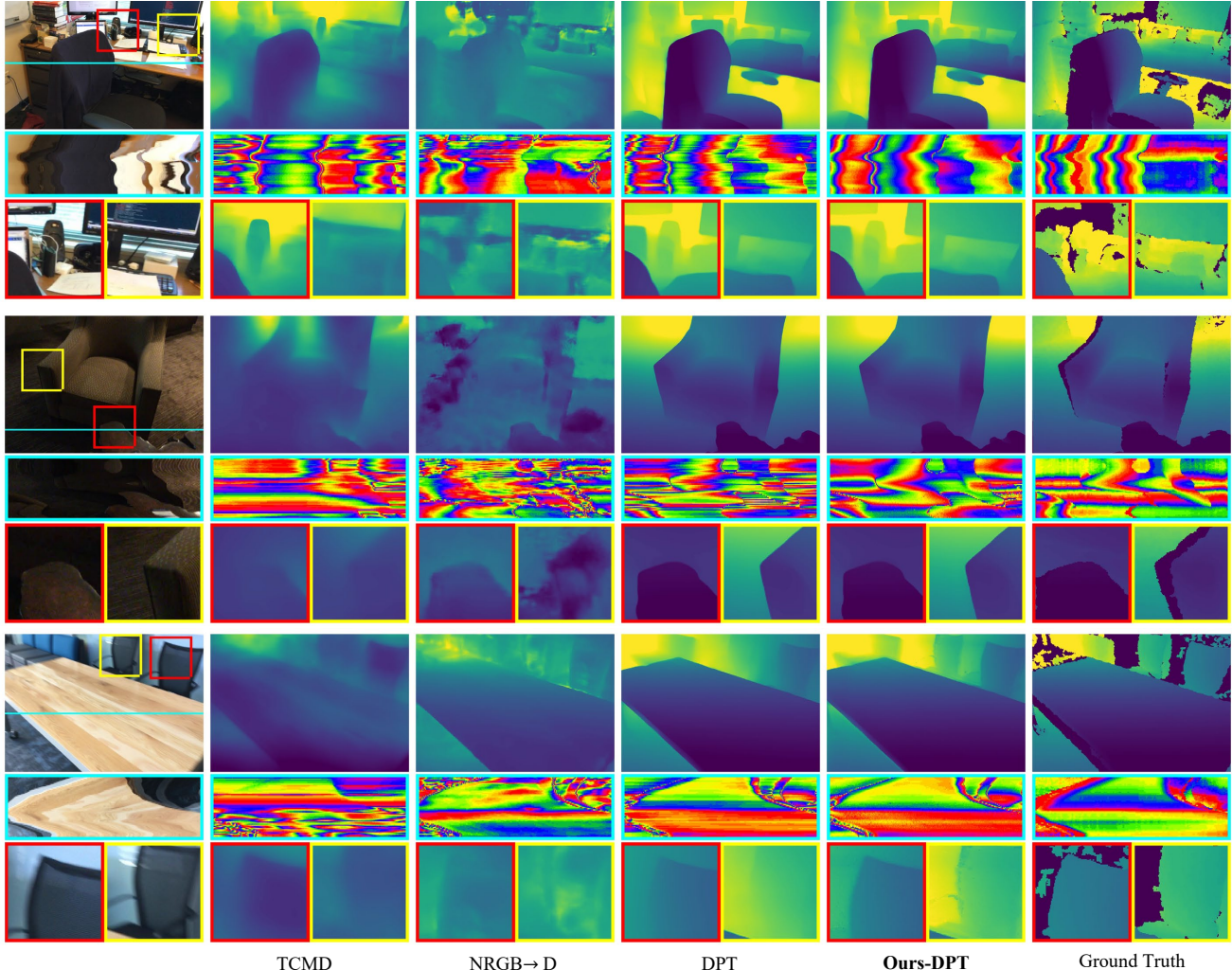
Figure 8. Qualitative comparison of the baseline monocular methods, and our approach with a DPT [32] backbone. We visualize consistency as variation in the cyan slice along the temporal dimension. Our method is more consistent, and captures finer spatial details.

latter having been used for dynamics filtering [20, 26, 55]. Table 4 show how ablating each stage affects the performance of our method. For *No Temporal Fusion* we set $\alpha = 0$ in Eq. 1; *No Spatial Fusion* uses $d_o^t = d_f^t$; and *No Global PC* does not use a global point cloud, using the reprojected points from frame $t-1$ only. The temporal fusion stage has the strongest effect on consistency and accuracy. Setting $\alpha = 0$ effectively averages the frames via Eq. 2 yielding temporally smooth — albeit erroneous — output. Hence, the drop in TCC but the lower OPW error as the latter is minimized when depth variance across frames is low.

6. Conclusion

We presented a method for temporally consistent video depth estimation in online settings. Our approach uses a global point cloud, and learnt image-space fusion to encourage consistency and preserve accuracy. While we assumed

camera poses and the scale of monocular depth was known, these can also be computed using a point cloud alignment algorithm such as Iterative Closest Point (ICP) [13, 29], or any visual SLAM system [10]. In the interest of generality, our method does not assume any particular depth-estimation algorithm. But the results may be improved by conditioning the estimation of d^t on the reprojected depth d_p^t. Furthermore, while our approach does not utilize the second view in stereo settings, it may be possible to exploit left-right view consistency in addition to temporal consistency [6]. We leave these directions for future work.

Acknowledgements

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 5

[2] Yuanzhouhan Cao, Yidong Li, Haokui Zhang, Chao Ren, and Yifan Liu. Learning structure affinity for video depth estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 190–198, 2021. 2, 6

[3] Gaurav Chaurasia, Arthur Nieuwoudt, Alexandru-Eugen Ichim, Richard Szeliski, and Alexander Sorkine-Hornung. Passthrough+ real-time stereoscopic view synthesis for mobile mixed reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(1):1–17, 2020. 1, 6

[4] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3

[5] Jaesung Choe, Sunghoon Im, Francois Rameau, Minjun Kang, and In So Kweon. Volumefusion: Deep depth fusion for 3d scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16086–16095, 2021. 2

[6] Steven D Cochran and Gerard Medioni. 3-d surface description from binocular stereo. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 14(10):981–994, 1992. 8

[7] Middlebury College. Middlebury stereo benchmark. https://vision.middlebury.edu/stereo/eval3/. 1

[8] Arun CS Kumar, Suchendra M Bhandarkar, and Mukta Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 283–291, 2018. 1, 2

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 5

[10] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015. 8

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4, 7

[12] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[13] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 4, 8

[14] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 1

[15] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013. 2

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 4, 5

[17] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[18] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 2, 6

[19] Damien Lefloch, Tim Weyrich, and Andreas Kolb. Anisotropic point-based fusion. In *Proceedings of International Conference on Information Fusion (FUSION)*, pages 1–9. ISIF, July 2015. 2

[20] Siyuan Li, Yue Luo, Ye Zhu, Xun Zhao, Yu Li, and Ying Shan. Enforcing temporal consistency in video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2021. 1, 2, 5, 6, 8

[21] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4521–4530, 2019. 1, 2, 6

[22] Zhaoshuo Li, Wei Ye, Dilin Wang, Francis X Creighton, Russell H Taylor, Ganesh Venkatesh, and Mathias Unberath. Temporally consistent online depth estimation in dynamic scenes. *arXiv preprint arXiv:2111.09337*, 2021. 4

[23] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021. 2, 3, 5, 6

[24] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural rgb → d sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10986–10995, 2019. 1, 2, 5, 6

[25] Xiaoxiao Long, Lingjie Liu, Wei Li, Christian Theobalt, and Wenping Wang. Multi-view depth estimation using epipolar spatio-temporal networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8258–8267, June 2021. 2, 7

[26] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 1, 2, 4, 6, 7, 8

[27] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134. 5

[28] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 6

[29] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 2, 4, 8

[30] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020. 1, 2

[31] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000. 3

[32] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 1, 2, 5, 6, 8

[33] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1, 2, 5, 6, 7

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[35] Paul Rosenthal and Lars Linsen. Image-space point cloud rendering. In *Proceedings of Computer Graphics International*, pages 136–143, 2008. 3

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[37] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[38] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021. 5

[39] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14372, 2021. 2, 3, 6

[40] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 5, 6

[41] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 5, 7

[42] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2021. 5

[43] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 5, 6

[44] Cristina Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Rob Romijnders, Nicolas Le Roux, and Ross Goroshin. Impact of aliasing on generalization in deep convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10529–10538, 2021. 1

[45] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *2019 International Conference on 3D Vision (3DV)*, pages 348–357. IEEE, 2019. 2, 5, 6

[46] Yiran Wang, Zhiyu Pan, Xingyi Li, Zhiguo Cao, Ke Xian, and Jianming Zhang. Less is more: Consistent video depth estimation with masked frames modeling. *arXiv preprint arXiv:2208.00380*, 2022. 6

[47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[48] Silvan Weder, Johannes L. Schönberger, Marc Pollefeys, and Martin R. Oswald. Routedfusion: Learning real-time depth map fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 4

[49] Silvan Weder, Johannes L. Schonberger, Marc Pollefeys, and Martin R. Oswald. Neuralfusion: Online depth fusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3162–3172, June 2021. 2

[50] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *European conference on computer vision*, pages 230–247. Springer, 2020. 2

[51] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black. Lessons and insights from creating a synthetic optical flow benchmark. In A. Fusiello et al. (Eds.), editor, *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, Part II, LNCS 7584, pages 168–177. Springer-Verlag, Oct. 2012. 5

[52] Lei Xiao, Salah Nouri, Joel Hegland, Alberto Garcia Garcia, and Douglas Lanman. Neuralpassthrough: Learned real-time view synthesis for vr. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1, 6

[53] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. Attention concatenation volume for accurate and efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12981–12990, 2022. 2, 3

[54] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 38(4):1–13, 2019. 2

[55] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020. 4, 8

[56] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 5, 6, 7

[57] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 1

[58] Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13116–13125, 2020. 1