2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Online Consistent Video Depth with Gaussian Mixture Representation

Chao Liu, Benjamin Eckart, and Jan Kautz

NVIDIA Research, Santa Clara, CA, USA

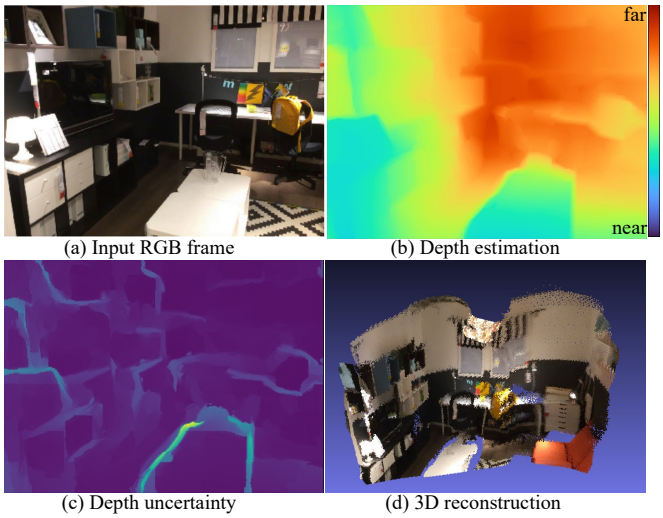
***Abstract*— We demonstrate how off-the-shelf single-image depth estimation methods can be augmented with guidance from optical flow to achieve consistent and accurate online depth estimation using video sequences of static scenes. While previous work has successfully leveraged the complementary nature of optical flow and depth estimation, these techniques use computationally expensive test time optimization strategies that do not generalize beyond a single video sequence and also require knowledge of the future. In contrast, we present a computationally efficient feed-forward design that runs in an online fashion by utilizing learned data priors from previously seen video sequences. To accomplish this, we propose a continuous geometric scene representation that parametrically and compositionally represents the scene as a Gaussian Mixture Model (GMM). Based on this representation, our pipeline learns to estimate consistent depths and associated camera poses from video sequences of static scenes without direct supervision. Our online method achieves state-of-the-art results compared against offline methods that require all sequence frames.**

I. Introduction

Estimating dense depth from an online sequence of monocular images, such as those coming from a video stream, is a crucial component of many computer vision applications. For example, dense depth estimates are useful for enabling special 3D effects in augmented reality type applications [33], or for various computational photography applications like geometrically consistent scene editing [43] or relighting [8]. Online estimation of monocular depth from video also has important uses in real-time applications, such as with self-driving vehicles that rely on consumer camera systems instead of or in addition to LiDAR and RADAR [10].

As opposed to traditional approaches like Structure from Motion (SfM) [42] and Multi-view Stereo (MVS) [15], some recent works have tried to estimate consistent, dense depth from videos by jointly optimizing over depth and poses with neural networks [26], [33]. These works optimize over a single video using a pre-trained single-image depth estimator. Since single-image depth prediction may be noisy and temporally inconsistent, these works fine-tune the network for that specific sequence in order to infer the best global set of dense depths that remain geometrically consistent. However, since these optimization methods need all frames both past and future, they are not deployable for online settings. Furthermore, fine tuning from scratch to a particular sequence limits generalizability to other sequences.

We therefore desire an online method that can take any off-the-shelf depth estimation neural network and 1) correct for frame-to-frame estimation variance while not needing to look ahead into the future, 2) leverage learned data priors from other similar sequences instead of being fine-tuned to a single sequence, and 3) perform sensor fusion while simultaneously predicting camera motion, all in a geometrically consistent and unsupervised way.



(a) Input RGB frame (b) Depth estimation

(c) Depth uncertainty (d) 3D reconstruction

Fig. 1. Our video depth estimation method takes an RGB video as input and outputs geometrically consistent dense depth, depth uncertainty maps, and camera poses between consecutive frames. Our method is online: for one specific frame, only the current frame and the single most recent frame are used; no future frames or distant previous frames are accessed. (a) One input RGB frame of the video; (b) and (c) The depth and uncertainty estimations; (d) Colored point cloud accumulated from 100 frames using the estimated depths and camera poses.

To solve these problems, we propose a framework that augments an off-the-shelf single-image depth estimator for online depth estimation using video of static scenes. Underlying the entire framework is a continuous geometric scene representation that we parametrize as a Gaussian Mixture Model (GMM). The 3D scene representation confers the following benefits: (1) *Generalizability*: Given a GMM's explicit parameterization and differentiability, we can leverage backpropagation for parameter optimization. (2) *Analytical Geometric Operators*: Compared to implicit methods that require volumetric integration via ray sampling [35], we derive closed form analytical ray-GMM equations for calculating occlusions, as well as depth map clustering into 3D GMMs. This allows us to move interchangeably between depth maps and 3D GMMs in a geometrically consistent manner. Analytical raycasting also makes the forward pass very efficient. (3) *Compositionality*: A GMM is compositional rather than monolithic; the latter of which is common in many MLP-

TABLE I
TAXONOMY COMPARISON

	Midas[41]	DELTAS[44]	RCVD[26]	DV2D[49]	Ours
Consistent			✓	✓	✓
Online	✓			✓	✓
Generalize	✓				✓
Cam. Pose	unknown	known	unknown	unknown	unknown
Scene	dynamic	static	dynamic	static	static
Supervision	depth	depth	flow	depth,pose	flow
GT needed	yes	yes	no	yes	no

based neural representations [35], [51]. Furthermore, during GMM formulation and raycasting to depth maps, only a sparse set of components are used. This also makes the forward pass efficient.

Our framework includes four parts: (1) a Depth Map Clustering module that converts a dense depth map to a 3D GMM; (2) a Ray Casting module that converts a transformed 3D GMM to a depth map; (3) a Pose Optimization module that geometrically aligns consecutive frames; (4) a FuseNet network that learns to combine dense depth maps given noisy and temporally inconsistent depth estimates from a single-image estimator. Our method requires only RGB videos and no depth/pose supervision for training. In contrast to other unsupervised consistent video depth methods that perform expensive optimization for consistency, our proposed method can do this task in a simple feed-forward, online, and frame-to-frame manner. We also demonstrate how our network design learns generalizable priors across multiple datasets and even across multiple depth/optical-flow estimators with our proposed GMM-based geometric scene representation.

II. RELATED WORK

Video Depth Estimation Single-view depth methods regress a dense depth map from a single RGB input image. The state-of-the-art performance is achieved by using a Deep Neural Network trained with direct depth supervision [7], [9], [28], [56], [41], [40], or cross-frame image intensity similarity [59], [14], [48], [17], [16], [27], [53]. However, for an RGB video, even the best single frame depth prediction networks may be noisy and have large frame-to-frame prediction variance since cross-frame geometrical consistency is not enforced during inference. Methods that model the cross-frame geometrical relation during inference have been proposed to combat this issue [33], [26], [49], [30]. Camera poses are estimated for pairs of frames, either via a separate optimization procedure [33], [26], [30] or feed-forward layers [49]. A comparison of those methods with ours is listed in Table I. Our method focuses on the online scene level reconstruction for RGB video input where the measured dense maps are not available [21], [37], [36], [55].

Scene representation To model the scene geometry, many underlying representations have been proposed, which we broadly categorize into three types:

(1) *Discrete Representations*, such as voxels [24] and multi-plane images [45], [52], can be directly used as input to CNN-based frameworks since the data entities reside in discrete regular grids. However, they are limited by discretization error and low memory efficiency, especially for fine-grained structure and empty space. Furthermore, applying continuous spatial transformations of the scene, which is a key element of our pipeline, can lead to aliasing artifacts.

(2) *Implicit Representations*, such as neural fields [51] modeling 3D density [35], occupancy [34], distance functions [39], or scene flow [58], [29] are free from discretization error and able to represent fine-grained structures with higher memory efficiency [3], [47]. These methods often leverage neural networks as a form of test-time optimization, such that the optimized weights encode the geometric properties of a single scene. For example, Neural Radiance Fields (NeRF) [35] optimizes for geometric consistency using a set of posed monocular images. However, to extract surfaces or geometry from an implicit representation, one needs to perform additional computation during both training and inference: ray sampling and integration in the case of radiance fields [54], [35], Marching Cubes [32] in the case of Occupancy Networks [34], or Sphere Tracing [18] in the case of distance fields [31], [47].

(3) *Gaussian Mixture Models* form a relatively new type of scene representation in neural networks. Being a continuous parametric form, the GMM facilitates efficient and exact rigid spatial transformations, differentiability, and high fidelity reconstruction. Genova *et al.* show how a GMM-like representation could be used in an implicit fashion to generate high quality 3D surface reconstructions, either through isosurface modeling [12] or by augmenting MLP-based implicit representations with locality and compositionality [11]. GMMs have been used to successfully encode hierarchical geometric concepts in neural networks [19], and also for robust neural 3D point cloud registration [57]. Learning to perform GMM clustering has also shown to be a strong self-supervised pretext task for learning 3D representations that transfer to downstream tasks [6]. For these reasons we also choose to adopt a GMM-based scene representation in our proposed pipeline.

III. CONTINUOUS GEOMETRIC REPRESENTATION

In this section, we begin with the definition of our GMM-based continuous geometric representation, followed by our two geometric operators: *depth map clustering*, the 3D GMM generation step from dense depth map, and *analytical raycasting* to get the depth map from another view.

A. 3D Gaussian Mixture Model

Given a dense depth map $\mathcal{D}^t = \{d_i^t\}$ from frame t of a video as a set of N pixels indexed by i, the corresponding point cloud can be estimated by back-projecting the pixels into the 3D space:

$$\mathcal{P}^t = \pi^{-1}(\mathcal{D}^t; K), \tag{1}$$

where π^{-1} is the back projection operation that maps points on the image plane to the 3D world given the dense depth map $\mathcal{D}^t$ and camera intrinsic matrix K.

The 3D GMM is a set of parametric clusters from point cloud $\mathcal{P}^t$. More specifically, for the j-th cluster out of J

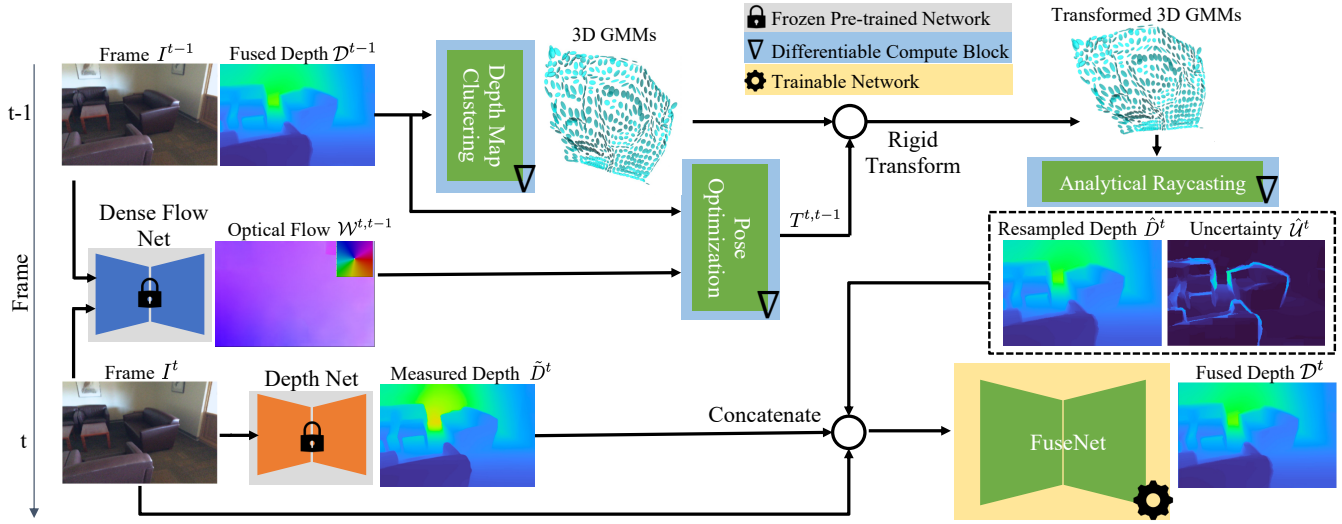


Fig. 2. Overview of our unsupervised learning method for online consistent video depth estimation. A 3D GMM (Sec. III-A) is maintained over frames to facilitate cross-frame geometric consistency. Given depth estimates from the previous frame, a 3D GMM is generated via our proposed *depth map clustering* technique (Sec. III-B). A dense depth map is resampled from the generated 3D GMM with respect to the new view of the current frame, using the *analytical raycasting* operator (Sec. III-C) and the relative camera pose between consecutive frames estimated by the Pose Optimization Block (Sec. IV-A). The resampled depth map is fused with the depth map predicted from a single-image depth estimator to get the final depth estimation (Sec. IV-B)

components, the locations, shapes and orientations of the clusters are modeled by 3D Gaussians in space:

$$\Theta_j^t = (w_j^t,\ \mu_j^t,\ \Sigma_j^t), \tag{2}$$

where the mean μ_j^t encodes the 3D location; the covariance Σ_j^t encodes the shape, size and orientation; the weight w_j^t encodes the contribution of a particular Gaussian since each point is modeled by multiple 3D Gaussian components.

An affinity matrix Γ_{ij}^t is defined by all $N \times J$ posterior probabilities, each entry describing the contribution of the i-th point $\mathbf{p}_i^t$ to the j-th cluster:

$$\Gamma_{ij}^t \overset{\text{def}}{=} p\left(\mathbf{p}_i^t | \Theta_j^t\right) = \frac{w_j^t S\left(\mathbf{p}_i^t; \mu_j^t, \Sigma_j^t\right)}{\sum_k w_k^t S\left(\mathbf{p}_i^t; \mu_k^t, \Sigma_k^t\right)}, \tag{3}$$

with $S\left(\mathbf{p}; \mu, \Sigma\right) = \exp\left(-\frac{1}{2}\left(\mathbf{p}-\mu\right)^T \Sigma^{-1}\left(\mathbf{p}-\mu\right)\right)$.

Our goal is to estimate a geometrically consistent depth stream $\{\mathcal{D}^t\}$ from an RGB input video by maintaining Γ^t and Θ_j^t for $j = 1, \cdots, J$ over each frame. To this end, we need to convert a dense depth map into 3D GMMs through depth map clustering, and recover the dense depth map from the 3D GMM given a new viewing camera pose.

B. Depth Map Clustering: Depth Map to GMM

Our parametric clustering technique to generate a 3D GMM from point clouds follows [5], [23]: given the point cloud or dense input features, we estimate the GMM parameters Θ by maximizing the likelihood of the points via an EM-like process [4]. However, unlike [23], rather than assuming isotropic Gaussians, we explicitly model fully anisotropic Gaussians to consider the shape and orientation of the local structure. Unlike [5] where a complete point cloud of a single object is given, our method handles partially observed point clouds in the FOV including multiple objects within the reconstructed scene.

The parametric clustering process consists of iterations of E-step and M-step. In the E-step, given the current estimation of the clusters $\{\Theta_j\}$ defined in Eq. 2 and the point cloud $\mathcal{P}$, we update the entries of the affinity matrix Γ using Eq. 3. For notation simplicity, we omit the frame index t unless it is otherwise required.

During the M-step, to update the parameter Θ_j for the clusters, we first define the zeroth, first and second moments of the points as:

$$M_0^j = \sum_i \Gamma_{ij},\ M_1^j = \sum_i \Gamma_{ij}\ p_i,\ M_2^j = \sum_i \Gamma_{ij}(p_i \otimes p_i),$$

with $\otimes$ notating the outer product of vectors. For a 3D point cloud with N points, the updated 3D GMM parameter Θ_j is estimated from these moments as:

$$w_j = \frac{M_0^j}{N},\ \mu_j = \frac{M_1^j}{M_0^j},\ \Sigma_j = \frac{M_2^j}{M_0^j} - \mu_j \otimes \mu_j \tag{4}$$

Enforcing sparsity for affinity matrix During the EM iterations, we want to enforce the Gaussians are spatially local so that no component will dominate all the points leading to mode collapse. More specifically, the affinity matrix should be sparse with only the entries for nearby point and 3D cluster pairs being non-zero: we divide the depth map into J patches and initialize the entries corresponding to the pixels and their 3×3 neighboring patches to be non-zero [1]. To enforce sparsity during the EM step, we keep the topology of the pixel-to-cluster pairs the same as in the initialization step and only update the non-zero entries in Γ from the initialization step.

We denote the parametric clustering procedure that generates anisotropic 3D GMMs from a dense point cloud as

$$\Theta, \Gamma = g(\mathcal{P}) \tag{5}$$

with $\mathcal{P}$ from the back-projection operation in Eq. 1; Θ is the collection of 3D GMM parameters for all the J clusters. Note that the procedure is differentiable so we can back-propagate it during the training session.

C. Analytical Ray Casting: GMM to Depth Map

Given the camera pose, we want to predict a depth map from the 3D GMM representation Θ. We can do this by considering each pixel in the depth map as the expected point of occlusion of a 3D ray defined by the camera intrinsics and extrinsics and passing through this GMM representation. Areas of high probability density with respect to the GMM representation are more likely to form the point of occlusion and vice versa.

An analytical ray-GMM interaction therefore can be formed as follows: casting any ray $\mathbf{r}$ corresponds to a 1D slice operation through a set of 3D Gaussians. This slice can be written in closed form: given the camera center $\mathbf{o}$ and ray direction $\mathbf{d}_i$, casting the i-th ray $\mathbf{r}_i(t) = \mathbf{o} + t\mathbf{d}_i$ across the j-th component defined by Θ_j results in a 1-dimensional weighted Gaussian function with with weight, first and second moments calculated as:

$$\begin{aligned} w_{ij} &= w_j S\left(\mathbf{r}_i\left(\mu_{ij}\right); \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right), \\ \mu_{ij} = \sigma_{ij}^2 \mathbf{d}_i^T \boldsymbol{\Sigma}_j^{-1} \boldsymbol{\mu}_j, &\quad \sigma_{ij}^2 = \left(\mathbf{d}_i^T \boldsymbol{\Sigma}_j^{-1} \mathbf{d}_i\right)^{-1}, \end{aligned} \tag{6}$$

with S the similarity function also used in Eq. 3.

For the i-th pixel in the image, we calculate its resampled depth $\hat{d}_i$ and uncertainty $\hat{u}_i$ as a weighted linear combination over 3D Gaussian components:

$$\hat{d}_i = \frac{\sum_j w_{ij}\Gamma_{ij}\mu_{ij}}{\sum_j w_{ij}\Gamma_{ij}}, \quad \hat{u}_i = \frac{\sum_j w_{ij}\Gamma_{ij}\sigma_{ij}}{\sum_j w_{ij}\Gamma_{ij}}. \tag{7}$$

Since the camera center $\mathbf{o}$ and the ray direction $\mathbf{d}_i$ depend on the camera pose and intrinsics, the resampled depth values from raycasting are a function of them as well. In summary, given the 3D GMMs Θ, the affinity matrix Γ, the camera pose T and intrinsics K, we can generate the resampled depth map $\hat{\mathcal{D}} = \{\hat{d}_i\}$ and its corresponding uncertainty map $\hat{\mathcal{U}} = \{\hat{u}_i\}$:

$$\hat{\mathcal{D}},\ \hat{\mathcal{U}} = \pi_d\left(T \circ \Theta, \Gamma; K\right), \tag{8}$$

where $T \circ \Theta$ is the 3D GMM composed with the rigid transformation T from the estimated camera pose, and which can be done in closed form by simply transforming the mean and covariance parameters.

IV. Unsupervised Learning for Online Consistent Video Depth Estimation

In this section, based on the assumption that the scene is static, we first introduce the pose optimization block that unrolls iterative updates for relative camera pose among frames. Then we introduce the FuseNet that fuses the resampled and measured depths based on estimated uncertainties. Finally we describe how we train the model from monocular RGB videos without depth and pose supervision.

A. Pose Optimization

To estimate the relative camera pose, we solve an optimization problem that minimizes reprojection error for dense correspondences between frames. More specifically, for the i-th pixel in frame $t-1$, we can get its 3D point $\mathbf{p}_i^{t-1}$ in frame $t-1$ and the ray directional vector $\mathbf{d}_i^t$ passing through its corresponding pixel in frame t, given the dense depth map $\mathcal{D}^{t-1}$, camera intrinsics and dense pixel correspondence from optical flow. Then we minimize the point-to-ray distance after applying the rigid transformation $T = [R \mid \mathbf{t}]$ induced by camera rotation R and translation $\mathbf{t}$:

$$\begin{aligned} &\min_{T=[R \mid \mathbf{t}]} \quad \sum_i \left\|\mathbf{v}(\mathbf{d}_i^t, \tilde{\mathbf{p}}_i^{t-1}; T)\right\| \\ &\text{with} \quad \mathbf{v}(\mathbf{d}_i^t, \tilde{\mathbf{p}}_i^{t-1}; T) = \left[\mathbf{d}_i^t\right]_\times T\tilde{\mathbf{p}}_i^{t-1} \end{aligned} \tag{9}$$

where $\mathbf{v}(\mathbf{d}_i^t, \tilde{\mathbf{p}}_i^{t-1}; T) \stackrel{\text{def}}{=} \mathbf{v}_i$ is the point-to-ray displacement vector for the i-th pixel after transformation; $[\mathbf{d}_i^t]_\times$ is the skew symmetrical matrix for the ray vector $\mathbf{d}_i^t$; $\tilde{\mathbf{p}}_i^{t-1}$ is the homogeneous coordinate of the 3D point $\mathbf{p}_i^{t-1}$. We ignore the superscript for T for notation simplicity.

We use the 6D vector $\mathbf{t}_{6\text{d}} = [\mathbf{t}, \theta]$ to represent the rigid transformation T, where θ is the 3D Euler angles for the rotation. The gradient of the point-to-ray distance $\sum_i \|\mathbf{v}_i\|$ *w.r.t.* $\mathbf{t}_{6\text{d}}$ can be written as:

$$\frac{\delta \sum_i \|\mathbf{v}_i\|}{\delta \mathbf{t}_{6\text{d}}} \sim \sum_i \hat{\mathbf{v}}_i^T \left[\mathbf{d}_i^t\right]_\times \left[\left[R\mathbf{p}_i^{t-1} + \mathbf{t}\right]_\times, I\right] \tag{10}$$

with $\hat{\mathbf{v}}_i$ the normalized vector of $\mathbf{v}_i$. The last term is based on the assumption that each update on $\mathbf{t}_{6\text{d}}$ is small such that we can linearize $T\tilde{\mathbf{p}}_i^{t-1}$ around the current state of $\mathbf{t}_{6\text{d}}$. As the gradient *w.r.t.* the pose can be obtained analytically, we can implement the gradient descent steps as unrolled layers.

B. FuseNet

Given the relative camera pose $T^{t,t-1}$ and dense point cloud $\mathcal{P}^{t-1}$ from the previous frame, we first generate the 3D GMM from $\mathcal{P}^{t-1}$ (Sec. III-B). Then we align the GMM towards the current frame t and perform analytical raycasting (Sec. III-C) to resample the dense depth and uncertainty maps from the current view:

$$\hat{\mathcal{D}}^t, \hat{\mathcal{U}}^t = \pi_d\left(T^{t,t-1} \circ \Theta^{t-1}, \Gamma^{t-1}; K\right) \tag{11}$$

The alignment and resampling steps are crucial for geometrical consistency among frames and relate the previous estimated state to the the current state, which will be updated given the incoming RGB frame using the FuseNet.

The FuseNet uses the Stacked Hourglass Network [38] as its backbone and takes the concatenation of the RGB image at frame t, the single-image depth estimation, the resampled depth and uncertainty as the input, as shown in Fig. 2. It learns to gate the less reliable estimations based on resampling uncertainty, geometric consistency, and image appearance. We regress for the gain G^t rather than the fused depth map $\mathcal{D}^t$ directly:

$$\mathcal{D}^t = \tilde{\mathcal{D}}^t + G^t\left(\hat{\mathcal{D}}^t - \tilde{\mathcal{D}}^t\right), \tag{12}$$

with $G^t = f(I^t, \tilde{\mathcal{D}}^t; \hat{\mathcal{D}}^t, \hat{\mathcal{U}}^t)$, where f is the FuseNet; I^t and $\tilde{\mathcal{D}}^t$ are the incoming RGB frame and its depth estimation from a single-view depth estimator.

C. Loss functions

Our loss function mainly includes three terms: the optical flow, depth consistency and depth smoothness losses.

Flow loss The optical flow loss models the discrepancy between the reference flow $\mathcal{W}^{t-1,t}$ and the induced flow $\hat{\mathcal{W}}^{t-1,t}$ given the estimated depth and camera pose. For the i-th pixel in frame t with image coordinate x_i^t and the corresponding 3D point at p_i^t, the optical flow for it from frame t to $t-1$ is

$$\hat{\mathcal{W}}^{t-1,t}(x_i^t) = \hat{x}_i^{t-1} - x_i^t$$

$\hat{x}_i^{t-1}$ is the corresponding reprojected location in the previous frame, given the depth and camera pose.

Given the induced optical flow, the optical flow loss is defined as its L1-norm difference from the reference flow:

$$\mathcal{L}_{\text{flow}} = \sum_i |\mathcal{W}^{t-1,t}\left(x_i^t\right) - \hat{\mathcal{W}}^{t-1,t}\left(x_i^t\right)|$$

Depth consistency loss We further enhance the consistency by adding a depth consistency loss between consecutive frames. The consistency is defined to be the difference between the z-component of the aligned point clouds, where the z-axis is aligned with the optical axis for the camera:

$$\mathcal{L}_{\text{geom}} = \sum_i |\left(T^{-1} \circ p_i^t\right)_z - \mathcal{D}^{t-1}\left(x_i^t + \mathcal{W}^{t-1,t}\left(x_i^t\right)\right)|$$

Depth smoothness loss We regularize the dense depth map to be smooth except for the regions with a strong appearance gradient:

$$\mathcal{L}_{\text{smooth}} = \sum_i |\nabla_x \mathcal{D}^t\left(x_i\right)| \exp\left(-|\nabla_x I^t\left(x_i\right)|\right)$$

The loss function for our unsupervised learning method is the sum of the loss terms for optical flow, depth consistency and smoothness:

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_g \mathcal{L}_{\text{geom}} + \lambda_s \mathcal{L}_{\text{smooth}}, \tag{13}$$

with $\lambda_g = 0.1$ and $\lambda_s = 1$. For all experiments, we apply the trained model on test videos without any fine-tuning.

V. Experiments

Performance comparisons We compare with state-of-the-art single-frame based methods: MiDaS and DPT [41], [40]. For comparison, we test with both the pre-trained MiDaS and DPT as the depth module in our framework. The weights of the depth module are fixed. Then we compare our model with the corresponding baselines (*i.e.* the single depth module). As shown in Table II, our model with either the DPT or MiDaS depth module improve over the corresponding baselines.

We compare with Robust CVD (RCVD) [26]. RCVD handles dynamic scenes by using a dynamic mask predicted from individual RGB frames. During optimization, the loss weights on pixels over dynamic objects are adjusted based on the dynamic mask. To make the comparison fair for RCVD, we fix the dynamic masks over all frames assuming that the scene is static, so the influence of the mask prediction procedure and re-weighting scheme of the pixels is ruled out.

TABLE II
PERFORMANCE ON SCANNET TEST SET

	$\sigma < 1.25 \uparrow$	$\sigma < 1.25^2 \uparrow$	abs. rel$\downarrow$	rmse$\downarrow$	scale inv.$\downarrow$
MiDaS [41]	62.64	93.32	0.1870	0.5913	0.2418
DPT [40]	63.08	95.97	0.1955	0.7538	0.2328
RCVD [26]	**71.63**	92.65	0.1953	0.8324	0.2637
Ours-MiDaS	64.97	94.27	**0.1775**	**0.5394**	0.2313
Ours-DPT	62.92	**97.51**	0.1904	0.7249	**0.2269**

TABLE III
COMPARISON WITH DV2D[49] WITH DEPTH AND POSE SUPERVISION.

Test	12scenes			TUM		
	$\sigma < 1.25 \uparrow$	rmse$\downarrow$	scale inv.$\downarrow$	$\sigma < 1.25 \uparrow$	rmse$\downarrow$	scale inv.$\downarrow$
DV2D	41.03	0.2601	0.2527	21.84	3.713	0.4691
Ours	**62.78**	0.2953	0.2677	**61.04**	**3.114**	**0.2934**
Ours+DV2D	57.47	**0.2031**	**0.2297**	27.10	3.270	0.4040

As shown in Table II, the performance of our proposed framework is better than RCVD for most metrics. More importantly, our method is online and only requires a single forward pass for each frame. In contrast, RCVD requires access to pairs of frames spanning the entire video, as well time-consuming test-time optimization to refine the dense depth maps. On a 200-frame input video, our method takes 0.5 sec/frame while RCVD takes around 18 sec/frame.

We also compare our method with DeepV2D [49], a learning based method supervised by both dense depth maps and camera poses during training. We conduct training and testing cross-dataset for both methods for fairness: the pre-trained DeepV2D model is trained on NYU dataset and our model is trained on ScanNet without depth or camera pose supervision. We test the models on 12-Scenes and TUM [46]. The results are listed in Tab. III. On the TUM dataset, our method performs significantly better than DeepV2D, indicating better generalization ability. On 12-Scenes, the performance of our method (Ours) is close to DeepV2D (DV2D). The better accuracy from DeepV2D is due to that the sequence of the 12-Scenes is similar to those in training dataset. Thus the model trained with GT depth and pose supervison can be more easily adopted on the test dataset. On the other hand, we use the output depth from DeepV2D as the measured depth (*i.e.* output from the depth estimtor in our pipeline), and plug it into our pipeline without finetuning the FNet (Ours+DV2D), we gain the performance boost due to geometrical consistent. The cross-dataset generalization ability and plug-and-play property of our model are further evaluated in the following.

Generalize across datasets As mentioned in Sec. IV-C, our framework learns only from the RGB videos without direct supervision of depth or camera pose. Although we could finetune the model on the given test video, we focus on the generalization ability to *unseen* videos for two reasons. First, our system is online. This stands in contrast to optimization-based or test-time finetuning methods [26], [42] that require

TABLE IV

CROSS-DATASET TRAINING AND TESTING PERFORMANCE GAIN.

Test \ Train	% Improvement (abs. rel, scale inv.) 12-Scenes	7-Scenes	T&T
ScanNet	10.99, 3.141	5.080, 4.342	11.25, 3.982
TUM	0.242, 0.765	1.155, 1.005	0.633, 1.324
12-Scenes	4.105, 2.825	4.815, 3.285	3.939, 2.727

TABLE V

CROSS-MODULE PERFORMANCE GAIN.

Train \ Test	% Improvement (scale inv.) DPT-l	Midas	DELTAS	FlowNet2	PerceiverIO
DPT-h [40]	1.746	4.564	5.193	1.566	1.401
DPT-l [40]	5.554	6.661	9.794	6.693	6.445
Midas [41]	7.084	4.342	7.632	7.398	6.669
DELTAS [44]	3.150	4.450	4.022	4.114	4.016

the system to be offline such that all frames are accessible; in addition, hundreds of iterations on pairs of frames in the test video are required for optimization. Instead, our method only requires a single forward inference pass. Second, the design of our system allows generalization to unseen data: the learned geometric fusion and analytic GMM operators are suitably generic such that they can be applied directly on the test video without finetuning for domain adaptation. We train our model on one dataset and test across other datasets without finetuning. To evaluate how the method generalizes to new dataset, we compare our method with the single image depth estimator within our pipeline (Ours-MiDaS *vs.* MiDas). If the performance increases, it means that the model has learned to fuse depth maps adaptively using estimated uncertainties. If the performance decreases, it indicates that the method has simply memorized the mappings between RGB images and the refined depth maps. As shown in Table IV, our method generalizes well across datasets: the performance increases for all cross-dataset combinations, even across indoor and outdoor videos, *e.g* from Tanks and Temples [25] (T&T) to 12-Scenes.

Generalize across depth and flow modules In the previous sections we show that our proposed pipeline increases the performance of off-the-shelf depth estimators while also comparing favorably to expensive and offline consistency optimizers. However, does this mean that our performance is specifically tied to the particular depth/flow estimation module on which it is trained?

To shed light on these questions, we evaluate the generalization ability of our method in the extreme scenario where after training the *depth and optical flow modules themselves* get replaced with completely different backbones. If test performance remains high even after swapping to a completely different backbone, it would be a strong indication that our approach has learned data priors that are invariant to the particularities of any single depth/flow estimation method. For this experiment, for the depth module, we use DPT and MiDaS, as before, and additionally include the 2-frame based DELTAS [44] as a third type of depth module. We

Fig. 3. **Qualitative Comparison** Accumulated point clouds from 100 frames of ScanNet [2] scenes using estimated depth maps and camera poses Our model is trained on 7scenes dataset [13] videos without using depth or pose for supervision, then tested on ScanNet without finetuning.

choose DELTAS to use the current frame and the most recent previous frame to match our proposed pipeline. For the optical flow module, we train our pipeline using RAFT [50] and test with FlowNet2 [20] and PerceiverIO [22]. The cross-module results are shown in Table V. Surprisingly, the performance increases for most combinations, *even when an entirely different depth or optical flow module* is used at test time. This means that our model learns to enforce the geometric consistency itself, rather than only eliminate the measurement noise for a specific depth/flow estimator.

We show qualitative results in Fig. 3. We accumulate the back-projected point clouds over 100-frames on ScanNet. The model is trained on 7scenes and tested on ScanNet, with the DPT-large depth module used in training being replaced with DELTAS. The camera poses are calculated by integrating the estimated relative camera poses between consecutive frames from our method. The results from our method have less ghosting artifacts and thus more sharp due to more geometric consistency.

VI. CONCLUSION

We present a method to augment an off-the-shelf depth estimation to be used in an online fashion for videos. We employ a continuous geometric representation in a novel way through the use of efficient geometric operators upon which we add the capability to learn generalizable priors. We hope our results inspire more future work in the direction of continuous or parametric geometric scene representations in an effort to produce systems that are efficient, compositional, and demonstrate robust geometric inductive biases.

References

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(11), 2274–2282 (2012). https://doi.org/10.1109/TPAMI.2012.120

[2] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)

[3] Davies, T., Nowrouzezahrai, D., Jacobson, A.: On the effectiveness of weight-encoded neural implicit 3d shapes. arXiv preprint arXiv:2009.09808 (2020)

[4] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)

[5] Eckart, B., Kim, K., Troccoli, A., Kelly, A., Kautz, J.: Accelerated Generative Models for 3D Point Cloud Data. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5497–5505 (2016). https://doi.org/10.1109/cvpr.2016.593

[6] Eckart, B., Yuan, W., Liu, C., Kautz, J.: Self-supervised learning on 3d point clouds by learning discrete generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8248–8257 (June 2021)

[7] Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NeurIPS (2014)

[8] El Helou, M., Zhou, R., Susstrunk, S., Timofte, R.: Ntire 2021 depth guided image relighting challenge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 566–577 (2021)

[9] Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep Ordinal Regression Network for Monocular Depth Estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

[10] Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)

[11] Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4857–4866 (2020)

[12] Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7154–7164 (2019)

[13] Glocker, B., Izadi, S., Shotton, J., Criminisi, A.: Real-time rgb-d camera relocalization. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 173–179 (2013). https://doi.org/10.1109/ISMAR.2013.6671777

[14] Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth prediction. In: The International Conference on Computer Vision (ICCV) (October 2019)

[15] Goesele, M., Curless, B., Seitz, S.M.: Multi-view stereo revisited. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 2402–2409. IEEE (2006)

[16] Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras (2019)

[17] Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

[18] Hart, J.C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. The Visual Computer **12**(10), 527–545 (1996)

[19] Hertz, A., Hanocka, R., Giryes, R., Cohen-Or, D.: Pointgmm: A neural gmm network for point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12054–12063 (2020)

[20] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)

[21] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., 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. pp. 559–568 (2011)

[22] Jaegle, A., Borgeaud, S., Alayrac, J.B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Brock, A., Shelhamer, E., Hénaff, O., Botvinick, M.M., Zisserman, A., Vinyals, O., Carreira, J.: Perceiver io: A general architecture for structured inputs & outputs. In: ICLR (2022)

[23] Jampani, V., Sun, D., Liu, M.Y., Yang, M.H., Kautz, J.: Superpixel Sampling Networks. In: ECCV 2018. pp. 363–380 (2018). https://doi.org/10.1007/978-3-030-01234-2_22

[24] Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2307–2315 (2017)

[25] Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017)

[26] Kopf, J., Rong, X., Huang, J.B.: Robust Consistent Video Depth Estimation. CVPR (2021)

[27] Li, H., Gordon, A., Zhao, H., Casser, V., Angelova, A.: Unsupervised monocular depth learning in dynamic scenes (2020)

[28] Li, Z., Dekel, T., Cole, F., Tucker, R., Snavely, N., Liu, C., Freeman, W.T.: Learning the depths of moving people by watching frozen people. In: Proc. Computer Vision and Pattern Recognition (CVPR) (2019)

[29] Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6498–6508 (2021)

[30] Liu, C., Gu, J., Kim, K., Narasimhan, S.G., Kautz, J.: Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10986–10995 (2019)

[31] Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2019–2028 (2020)

[32] Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)

[33] Luo, X., Huang, J.B., Szeliski, R., Matzen, K., Kopf, J.: Consistent video depth estimation. ACM Transactions on Graphics (TOG) **39**(4), 71–1 (2020)

[34] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)

[35] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)

[36] Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 343–352 (2015)

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

[38] Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 483–499. Springer International Publishing, Cham (2016)

[39] Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)

[40] Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. ICCV (2021)

[41] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)

[42] Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[43] Shih, M.L., Su, S.Y., Kopf, J., Huang, J.B.: 3d photography using context-aware layered depth inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8028–8038 (2020)
[44] Sinha, A., Murez, Z., Bartolozzi, J., Badrinarayanan, V., Rabinovich, A.: Deltas: Depth estimation by learning triangulation and densification of sparse points. In: ECCV (2020)
[45] Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 175–184 (2019)
[46] Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS) (Oct 2012)
[47] Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C., Nowrouzezahrai, D., Jacobson, A., McGuire, M., Fidler, S.: Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11358–11367 (2021)
[48] Tang, C., Tan, P.: BA-net: Dense bundle adjustment networks. In: International Conference on Learning Representations (2019)
[49] Teed, Z., Deng, J.: Deepv2d: Video to depth with differentiable structure from motion. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020 (2020)
[50] Teed, Z., Deng, J.: RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. arXiv (2020)
[51] Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Wang, Y., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., Simon, T., Theobalt, C., Niessner, M., Barron, J.T., Wetzstein, G., Zollhoefer, M., Golyanik, V.: Advances in Neural Rendering. arXiv (2021)
[52] Tucker, R., Snavely, N.: Single-view view synthesis with multiplane images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 551–560 (2020)
[53] Wang, C., Miguel Buenaposada, J., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
[54] Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5610–5619 (2021)
[55] Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., Davison, A.: Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems (2015)
[56] Wimbauer, F., Yang, N., von Stumberg, L., Zeller, N., Cremers, D.: MonoRec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
[57] Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., Kautz, J.: Deepgmr: Learning latent gaussian mixture models for registration. In: European Conference on Computer Vision. pp. 733–750. Springer (2020)
[58] Zhang, Z., Cole, F., Tucker, R., Freeman, W.T., Dekel, T.: Consistent depth of moving objects in video. ACM Transactions on Graphics (TOG) **40**(4), 1–12 (2021)
[59] Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)