LightedDepth: Video Depth Estimation in light of Limited Inference View Angles

Shengjie Zhu and Xiaoming Liu
Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, 48824
zhusheng@msu.edu, liuxm@cse.msu.edu

Abstract

Video depth estimation infers the dense scene depth from immediate neighboring video frames. While recent works consider it a simplified structure-from-motion (SfM) problem, it still differs from the SfM in that significantly fewer view angels are available in inference. This setting, however, suits the mono-depth and optical flow estimation. This observation motivates us to decouple the video depth estimation into two components, a normalized pose estimation over a flowmap and a logged residual depth estimation over a mono-depth map. The two parts are unified with an efficient off-the-shelf scale alignment algorithm. Additionally, we stabilize the indoor two-view pose estimation by including additional projection constraints and ensuring sufficient camera translation. Though a two-view algorithm, we validate the benefit of the decoupling with the substantial performance improvement over multi-view iterative prior works on indoor and outdoor datasets. Codes and models are available at https://github.com/ShngJZ/LightedDepth.

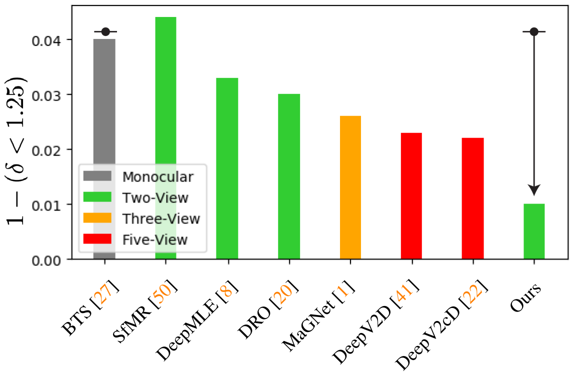


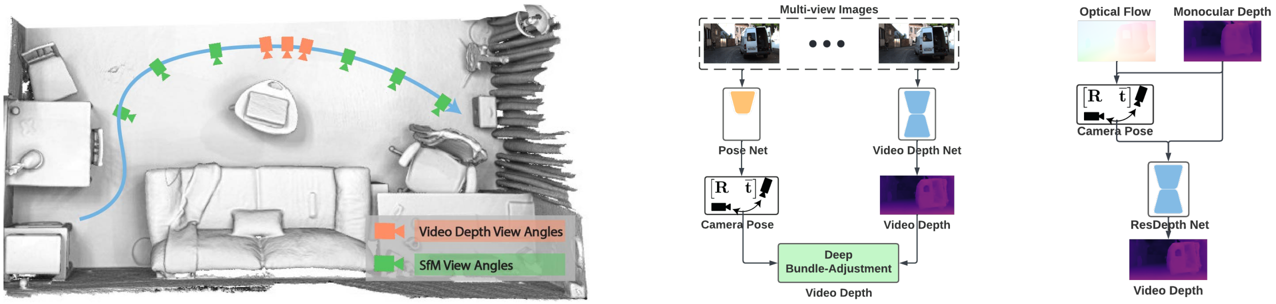
Figure 1. Video Depth Performance Comparison on KITTI Dataset. We mark the methods taking different numbers of frames with different colors. We propose a two-view video depth estimation method that substantially outperforms prior two-view, three-view, and five-view methods. Our method uses a monocular depth as initialization. The arrow marks our improvement when using the BTS [27] as the initialization. Comparison is detailed in Tab. 1.

1. Introduction

Depth estimation is a fundamental task for applications such as 3D reconstruction [3], robotics [26], and autonomous driving [59]. The depth is self-contained in the scene motion brought by the camera movement. The classic SfM methods [17, 31, 37, 38, 54] hence jointly recover the scene depth and camera poses by applying bundle-adjustment over the entire video sequence. However, the iterative optimization defined over all frames makes SfM a computationally intensive method. Video depth estimation simplifies the computation by only consuming the immediate neighboring frames. In consequence, only limited camera view angles are available, as shown in Fig. 2 (a).

The limited camera views, however, suit optical flow and monocular depth estimation. We are then motivated to connect video depth to mono-depth and flow estimation by decoupling the video-depth into two components. First, we use the flowmap to estimate a normalized up-to-scale camera pose, *i.e.*, camera pose with a unit-length translation vector. Second, we estimate video depth as a logged residual over the mono-depthmap. The two components are unified by an efficient off-the-shelf camera scale alignment algorithm, aligning the depthmap and flowmap, making the residual depth estimation a stereo matching.

Unlike our method, most prior video depth estimation works [41, 46, 50, 52, 55] formulate their solutions as deep SfM, shown in Fig. 2 (b). They can be grouped into two types [50]. Type **I** methods [41, 46, 52] execute SfM within a fixed frame window, embedding bundle-adjustment as a differentiable module within a network. Type **II** methods [50, 55] execute a consecutive-frame SfM. They sequentially estimate an up-to-scale pose and an up-to-scale depthmap. While prior works solve video depth estimation as a simplified SfM problem, our method differs in decoupling the video depth estimation to two sub-tasks which are robust to deficient camera views, *i.e.*, flow based normalized



(a) Limited view angles of video depth (b) Prior Multi-View (c) Ours Two-View

Figure 2. (a) Unlike classic SfM, video depth estimation possesses significantly fewer view angles during inference. (b) Prior multi-view video depth estimation works [40, 41, 46] mimic SfM pipeline, focusing on improving deep bundle-adjustment. (c) Considering the SfM alike pipelines are compromised by the limited view angles, we base the video depth estimation on two deficient view robust sub-tasks, *i.e.*, the relative camera pose estimation based on the flowmap, and the logged residual video depth estimation based on the monocular depthmap. The two sub-tasks are connected by a novel and efficient scale alignment algorithm. We skip RGB inputs for simplicity in (c).

pose estimation and logged residual depth estimation.

On pose estimation, we compare the optical flow with the projection flow computed from the pose and depthmap, using the State-of-The-Art (SoTA) methods of each side, *i.e.*, DeepV2D [41] and RAFT [42]. The results in Supp.Tab. 1 show that the optical flow is more robust than the projection flow. Since the flow performance is a bottleneck for pose performance, this suggests, instead of optimizing poses by bundle-adjustment together with the depthmap as the type **I** method, directly estimating the pose from flowmap can be more accurate, as the noise inside the depthmap is avoided. We follow [60] in using the five-point algorithm [29] with RANSAC [13] to estimate the normalized pose.

On video depth estimation, we treat it as a log space residual estimation over the monocular initialization. While prior works [41, 46, 52] already adopt mono-depthmap as initialization, the connection between monocular and video depth is under-explored. Prior works simply repeat the video depth estimation after updating the pose. Specifically, they estimate the video depth by a 3D cost volume constructed by sampling the next frame feature map at different projected locations specified by pre-defined depth candidates. Instead, we change the sampling from fixed candidates to fixed *log space residual* candidates. This brings three benefits: (1) It enables the video depth to benefit from SoTA monocular depth. (2) It improves the sampling efficiency in constructing the cost volume, as candidates are drawn dynamically, centering around the initial guess rather than fixed. (3) It provides a reliable lower-bound depth performance for moving foreground objects and static frames.

The residual video depth estimation is stereo matching via an estimated pose. Yet, we only estimate the normalized pose, still lacking the baseline. We then propose an efficient voting based scale alignment algorithm, estimating the camera scale by aligning the monocular depthmap with flowmap. This algorithm connects the two decoupled subtasks: the normalized pose and residual depth estimation.

Empirically, we find that the five-point algorithm runs less accurately in indoor scenarios. This is because indoor videos are taken by hand-held cameras, possessing much more rotation movement than outdoor videos taken by car-mounted cameras. The additional rotation movement weakens the epipolar constraint, which is required by the five-point algorithm. To tackle the issue, during each RANSAC consensus checking, we perform the scale alignment algorithm, turning normalized camera pose to metric space pose. Then, we include an additional projection constraint to the original epipolar constraint. It improves both indoor depth and pose performance.

We estimate the camera scale from the mono-depth instead of video depthmap. Ideally, similar to residual depth learning, we may use an additional cost volume based decoder to learn the residual camera scale. However, we show that under robust pose and flow estimate, *the camera scale learning loss can be converted to a relaxed depth learning loss*, as the two only differ by a constant in log space. This reduces camera scale learning to depth learning. Empirically and theoretically, we show that a single decoder is sufficient for both residual depth and camera scale learning.

We summarize the contributions of our work as follows:

- We propose a comprehensive two-view video depth estimation method. Unlike a simplified SfM, we decompose into two sub-tasks that are robust to deficient view angles, and connect them via an efficient scale alignment algorithm.

- We stabilize the indoor normalized pose estimation with the additional projection constraint.

- Theoretically and empirically, we prove the equality between scale and video depth learning.

- On KITTI [16] and NYUv2 [35] datasets, our two-view sequential method reduces 56.5% and 34.1% error on the metric $\delta < 1.25$ of video depth estimation over SoTA multi-view iterative work [41].

2. Prior Works

Pose and Depth from Multi-View System Structure-from-motion (SfM) [17,31,37,38,54] is the classic approach to recover scene geometry and camera motion from video. After proper initialization, the pose and 3D points are fine-tuned by bundle-adjustment over the input point correspondences. Visual simultaneously localization and mapping (vSLAM) methods [10, 11, 34, 36, 39, 43, 44, 54] are similar to SfM but focus on odometry.

Video depth estimation is the other multi-view system. It contrasts to SfM as operating on fixed frame windows, providing limited camera views. Recent works [12, 20, 22, 40, 41, 46, 52, 61] solve video depth estimation as an SfM problem. Inspired by classic SfM, they propose different deep bundle-adjustment modules, minimizing a residual term during the network inference. For instance, [52] and [41] separately propose a first-order and second-order deep optimization scheme. [52] applies an exhaustive search over a local region in the pose parameter space. Given the projection flow computed by the current depth and pose, [41] employs a motion module to estimate a residual flow term. The pose is refined via applying a Gauss-Newton update [53]. Surprisingly, compared to estimating residual pose in inference, none of the prior works estimate residual depth.

Our work solves the video depth estimation from the other perspective. Instead of emphasizing the improved deep bundle-adjustment module, we decompose the video depth into sub-tasks that are robust to narrow view angles. Our work can benefit other multi-view methods via serving as their two-view initialization module [41, 52].

Deep Two-View Structure-from-Motion SfMR [50] revisits the classic two-view SfM [7, 25] with deep learning. They first solve a normalized pose from the input flowmap and then estimate a normalized depthmap, *i.e.*, depthmap divided by the camera scale.

Our method improves [50] in multiple perspectives. First, we validate that the optical flow is more robust than the projection flow between immediate frames (detailed in Supp.Tab. 1.). This completes the motivation of estimating normalized pose from the flowmap instead of applying deep bundle-adjustment. In comparison, [50] only discusses its improvement over classic SIFT [32] based two-view SfM. Second, we improve indoor pose estimation performance by including the additional projection constraint. Third, the normalized depth in [50] is poorly ranged, varying from zero to infinity, while the proposed logged residual depth is well ranged. As a result, our model with 32 depth candidates outperforms [50] with 128 depth candidates. Fourth, our method does not require groundtruth pose to produce normalized depth. The normalized pose and camera scale are learned from synthetic flow and groundtruth depth labels, avoiding the noise from the IMU or GPS device.

Multi-View-Stereo With the optimized camera poses, video depth estimation is treated as a multi-view-stereo (MVS) problem. Similar to SfM, most MVS methods [6, 30, 48, 49, 56, 57] assume sufficient view variations, estimating without an init mono-depthmap. A concurrent MVS work [1], however, positions itself to infer depth within a limited frame window. [1] skips the non-trivial pose estimation and models depth as a Gaussian distribution. The video depth is estimated by selecting the residual that max-a-posteriori. However, unlike us, they do not align depthmap with the camera pose scale, lacking geometric constraint. In return, though [1] uses groundtruth poses and more frames, we still outperform this iterative method, as in Tab. 1.

3. Proposed Method

Our objective is to jointly solve the interdependent pose and depth given two video frames. Take the process of reconstructing image $\mathbf{I}_m$ at frame m from image $\mathbf{I}_n$ at frame n under a depthmap $\mathbf{D}$ and pose $\mathbf{P}$ as $\mathbf{I}_m^* = g\left(f\left(\mathbf{D}, \mathbf{P}\right), \mathbf{I}_n\right)$, where $\mathbf{I}_m^*$ is the reconstructed image. $f(\cdot)$ produces 2D projection locations in $\mathbf{I}_n$, as a function of $\mathbf{D}$, $\mathbf{P}$, and the intrinsic matrix $\mathbf{K}$ (skipped in $f(\cdot)$ for simplicity). $g(\cdot)$ applies bilinear sampling to $\mathbf{I}_n$ at 2D locations from $f\left(\mathbf{D}, \mathbf{P}\right)$. Formally, we aim to compute the depth $\mathbf{D}^\dagger$ and pose $\mathbf{P}^\dagger$ by optimizing the photometric constraint:

$$\mathbf{P}^\dagger, \mathbf{D}^\dagger = \arg\min_{\mathbf{P}, \mathbf{D}} h_p\left(g\left(f\left(\mathbf{D}, \mathbf{P}\right), \mathbf{I}_n\right), \mathbf{I}_m\right), \tag{1}$$

where $h_p(\cdot)$ can be defined in forms such as structural similarity index measure (SSIM) [51, 63]. Recent multi-view works [12,20,22,40,41,46,52,61] focus on improved mechanisms which, **in inference time**, enforce Eqn. 1. Typically, they adopt an iterative and alternative optimization scheme, minimizing Eqn. 1 by iteratively solving:

$$\begin{cases} \mathbf{P}^\dagger = \arg\min_{\mathbf{P}} h_p\left(g\left(f\left(\mathbf{D}, \mathbf{P}\right), \mathbf{I}_j\right), \mathbf{I}_i\right) & \text{(2a)} \\ \mathbf{D}^\dagger = \arg\min_{\mathbf{D}} h_p\left(g\left(f\left(\mathbf{D}, \mathbf{P}\right), \mathbf{I}_j\right), \mathbf{I}_i\right). & \text{(2b)} \end{cases}$$

For simplicity, Eqn. 2 is written with two-view inputs. Interestingly, their optimization is primarily for pose estimation. If an optimal pose $\mathbf{P}^\dagger$ is given, video depth is estimated through a single forward inference [12, 20, 22, 40, 41, 46, 52, 61]. In comparison, our method runs *sequentially*. Given the input flow $\mathbf{O}$ and mono-depth initialization $\mathbf{D}^*$, we decouple the video depth estimation into two narrow-view robust objectives:

$$\begin{cases} \overline{\mathbf{P}}^\dagger, s^\dagger = \arg\min_{\overline{\mathbf{P}}, s} \left(h_e\left(\overline{\mathbf{P}}, \mathbf{O}\right) + \right. & \\ \qquad\qquad \left. \lambda \cdot h_c\left(f\left(\mathbf{D}^*, p\left(\overline{\mathbf{P}}, s\right)\right), \mathbf{O}\right)\right) & \text{(3a)} \\ \mathbf{D}^\dagger = \arg\min_{\mathbf{D}} h_p\left(g\left(f\left(\mathbf{D}^*, p\left(\overline{\mathbf{P}}, s\right)\right), \mathbf{I}_j\right), \mathbf{I}_i\right). & \text{(3b)} \end{cases}$$

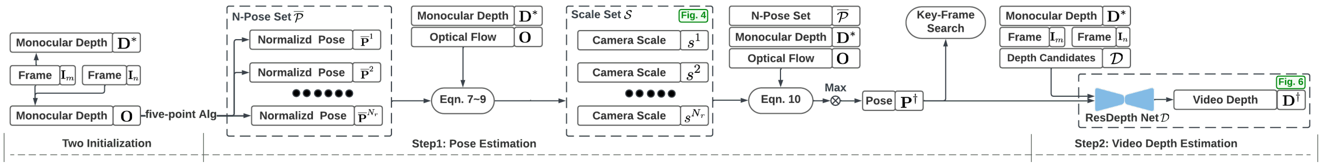

Monocular Depth $\mathbf{D}^*$
Frame $\mathbf{I}_m$
Frame $\mathbf{I}_n$
Monocular Depth $\mathbf{O}$
five-point Alg
Two Initialization
N-Pose Set $\overline{\mathcal{P}}$
Normalizd Pose $\overline{\mathbf{P}}^1$
Normalizd Pose $\overline{\mathbf{P}}^2$
Normalizd Pose $\overline{\mathbf{P}}^{N_r}$
Monocular Depth $\mathbf{D}^*$
Optical Flow $\mathbf{O}$
Eqn. 7~9
Scale Set $\mathcal{S}$
Fig. 4
Camera Scale s^1
Camera Scale s^2
Camera Scale s^{N_r}
N-Pose Set $\overline{\mathcal{P}}$
Monocular Depth $\mathbf{D}^*$
Optical Flow $\mathbf{O}$
Eqn. 10
Max
Pose $\mathbf{P}^\dagger$
Key-Frame Search
Step1: Pose Estimation
Monocular Depth $\mathbf{D}^*$
Frame $\mathbf{I}_m$
Frame $\mathbf{I}_n$
Depth Candidates $\mathcal{D}$
Fig. 6
ResDepth Net $\mathcal{D}$
Video Depth $\mathbf{D}^\dagger$
Step2: Video Depth Estimation

Figure 3. Our algorithm takes two RGB inputs $(\mathbf{I}_m, \mathbf{I}_n)$, the initial mono-depth $\mathbf{D}^*$, and flowmap $\mathbf{O}$ as inputs. Our proposed framework consists of 2 key steps: (1) An improved five-point algorithm. Given flowmap $\mathbf{O}$ and mono-depth map $\mathbf{D}$, apply consensus check over randomly initiated normalized pose set $\overline{\mathcal{P}}$ and its corresponding scale set $\mathcal{S}$. (2) Residual video depth estimation with a cost volume network. Between the two steps, we perform key-frame search if under insufficient camera translation, *i.e.*, re-estimate flowmap and pose with the next frame. Scale set $\mathcal{S}$ estimation and video depth $\mathbf{D}^\dagger$ estimation are further detailed in Fig. 4 and 6.

Function $p(\cdot)$ combines normalized pose $\overline{\mathbf{P}}$ with scale s: $p\left(\overline{\mathbf{P}}, s\right) = \left[\mathbf{R} \quad s \cdot \bar{\mathbf{t}}\right]$. $\mathbf{D}^*$ and $\mathbf{O}$ are initial mono-depthmap and flowmap. $\mathbf{D}^\dagger$ and λ are the optimized video depthmap and a predefined weighting parameter. Functions $h_e(\cdot)$ and $h_c(\cdot)$ are epipolar and projection consistency constraints detailed in Sec. 3.1.

The rest of the section presents our sequential pose and video depth estimation. We discuss about the equality between scale and depth learning at the end of the section. The overall framework is illustrated in Fig. 3.

3.1. Pose Estimation

We optimize Eqn. 3a in camera pose estimation. Given the flowmap $\mathbf{O}$ and mono-depthmap $\mathbf{D}^*$, we reformulate the five-point [29] algorithm with RANSAC [13] to include an additional projection consistency constraint. Specifically, for each normalized pose $\overline{\mathbf{P}}$ initiated by the five-point algorithm, a pixel-wise camera scale is determined given the pixel-wise depth and flow pair. The optimal scale is therefore selected by voting, see Fig. 4. This enables us to include a projection constraint in addition to the epipolar constraint during the RANSAC consensus checking.

Random Normalized Pose Initiates. We denote the N_k pixels randomly sampled from frame $\mathbf{I}_m$, flowmap $\mathbf{O}$ and monocular depthmap $\mathbf{D}^*$ as $\{\mathbf{p}\}$, $\{\mathbf{o}\}$ and $\{d\}$. Then frame $\mathbf{I}_n$'s corresponded pixels $\{\mathbf{q}\}$ are given as $\{\mathbf{q}_k \mid \mathbf{q}_k = \mathbf{p}_k + \mathbf{o}_k, k \in N_k\}$, where N_k is the number of randomly sampled correspondence. For simplicity, we assume the RANSAC algorithm loops to the max iteration number N_r, where r indexes each RANSAC loop. Meanwhile, in each loop, a quick chirality check [29] is applied to convert the essential matrix to the normalized pose. As such, we initiate N_r random normalized pose with the five-point algorithm, denoted as the set $\overline{\mathcal{P}} = \{\overline{\mathbf{P}}^r \mid r \in N_r\}$.

Pixel-wise scale estimation. Given any normalized pose $\overline{\mathbf{P}} = \left[\mathbf{R} \quad \bar{\mathbf{t}}\right]$, the depth value of each pixel can determine a camera scale. We name the set of camera scales determined by each depth pixel as pixel-wise scale $\mathbf{s}$. Set $\mathbf{p} = \left[p^x \quad p^y \quad 1\right]^\intercal$ and $\mathbf{q} = \left[q^x \quad q^y \quad 1\right]^\intercal$ are the homogeneous pixel coordinates in $\mathbf{I}_m$ and $\mathbf{I}_n$, connected by flow $\mathbf{O}$ at pixel $\mathbf{p}$. Set camera projection as:

$$d'\mathbf{q} = d' \left[q^x \quad q^y \quad 1\right]^\intercal = d\mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1}\,\mathbf{p} + s\mathbf{K}\,\bar{\mathbf{t}}. \tag{4}$$

The d and d' refer to depth at frame $\mathbf{I}_m$ and $\mathbf{I}_n$. By arranging Eqn. 4, we acquire the relationship between depth d and scale s at horizontal and vertical directions separately as:

$$d^x = s\frac{x - q^x \cdot z}{q^x \mathbf{m_3}^\intercal\,\mathbf{p} - \mathbf{m_1}^\intercal\,\mathbf{p}}, \quad d^y = s\frac{y - q^y \cdot z}{q^y \mathbf{m_3}^\intercal\,\mathbf{p} - \mathbf{m_2}^\intercal\,\mathbf{p}}. \tag{5}$$

Here $\left[\mathbf{m}_1 \quad \mathbf{m}_2 \quad \mathbf{m}_3\right]^\intercal = \mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1}$, $\left[x \quad y \quad z\right]^\intercal = \mathbf{K}\,\bar{\mathbf{t}}$. As in Fig. 4 (a), optical flow induced pixel $\mathbf{q}$ may not reside on the epipolar line $\mathbf{l}_\mathbf{p}$, making d^x and d^y possess different values. To pursue a unique mapping between scale s and depth d, we compute the optimal pixel-wise scale s by minimizing the L_2 distance between input monocular depth d and d^x, d^y:

$$s = \arg\min_s \left(d^x - d\right)^2 + \left(d^y - d\right)^2. \tag{6}$$

Then the pixel-wise mapping from depth d to scale s is:

$$\log(s) = \log(d) + m, \tag{7}$$

where $m = -\log\frac{1}{2}\left(\frac{x - q_k^x \cdot z}{q_k^x \mathbf{m}_3^\intercal\,\mathbf{p}_k - \mathbf{m}_1^\intercal\,\mathbf{p}_k} + \frac{y - q_k^y \cdot z}{q_k^y \mathbf{m}_3^\intercal\,\mathbf{p}_k - \mathbf{m}_2^\intercal\,\mathbf{p}_k}\right)$. The proof is detailed in the supplementary material.

Camera Scale Estimation. Next, we determine the unique camera scale s^r from the pixel-wise scale set $\mathbf{s}^r$ under normalized pose $\overline{\mathbf{P}}^r$ by majority voting, as shown in Fig. 4. Specifically, we produce the histogram of the scale set $\mathbf{s}^r$ as a B-dim vector $\mathbf{r}$. For the b_{th} element of $\mathbf{r}$, its value $\mathbf{r}[b]$ is:

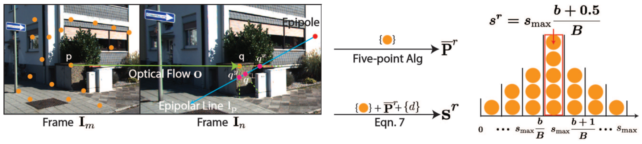
$$\mathbf{r}[b] = \sum_{k=1}^{N_k}\left(\frac{b}{B} \cdot s_{\max} \leq s_k < \frac{b+1}{B} \cdot s_{\max}\right). \tag{8}$$

Hyper-parameter $s_{\max}$ is the max scale value we record. The optimal scale s^r under normalized pose $\overline{\mathbf{P}}^r$ is then:

$$s^r = s_{\max}\frac{b + 0.5}{B}, \qquad b = \arg\max_{0 \leq b < B} \mathbf{r}[b]. \tag{9}$$

To this step, for the N_r randomly sampled normalized pose $\overline{\mathcal{P}}$ in RANSAC, we conclude the corresponded N_r scale estimate, denoted as set $\mathcal{S} = \{s^r \mid r \in N_r\}$.

Consensus Check. As in Fig. 5, we introduce an additional projection constraint h_c to stabilize the five-point algorithm


Epipole
p
q
q^x
q^y
q
Optical Flow $\mathbf{o}$
Epipolar Line $\mathbf{l}_\mathbf{p}$
Frame $\mathbf{I}_m$
Frame $\mathbf{I}_n$
{●}
Five-point Alg
$\overline{\mathbf{P}}^r$
{●} + $\overline{\mathbf{P}}^r$ + $\{d\}$
Eqn. 7
$\mathbf{s}^r$
$s^r = s_{\max}\frac{b+0.5}{B}$
0
$\cdots$ $s_{\max}\frac{b}{B}$ $s_{\max}\frac{b+1}{B}$ $\cdots$ $s_{\max}$

(a) Pixel-wise scale estimation (b) Camera scale estimation

Figure 4. We randomly sample N_k pixels $\{\mathbf{p}\}$ on frame $\mathbf{I}_m$, marked in orange. Corresponded frame $\mathbf{I}_n$'s pixels $\{\mathbf{q}\}$ are determined by flowmap $\mathbf{O}$. Sampled depth is $\{d\}$. We illustrate: (a) Due to the noise, corresponded pixel $\mathbf{q}$ does not comply projective geometry, *i.e.*, $\mathbf{q}$ resides outside the epipolar line $\mathbf{l}_\mathbf{p}$. In Eqn. 6, we approximate the scale determined by pixel $\mathbf{q}$ with two pixels $\mathbf{q}^x$ and $\mathbf{q}^y$, residing horizontally and vertically on epipolar line $\mathbf{l}_\mathbf{p}$. (b) One normalized pose $\overline{\mathbf{P}}^r$ is initiated by five-point algorithm. Next, with Eqn. 7, we acquire a pixel-wise scale set $\mathbf{s}^r$. After producing the B-dim histogram of scale set $\mathbf{s}^r$, the optimal scale s^r is determined by majority voting.

in indoor videos. For the r_{th} randomly sampled normalized pose $\overline{\mathbf{P}}^r$, given $\{\mathbf{p}\}$, $\{\mathbf{q}\}$, $\{\mathbf{o}\}$ and $\{d\}$, the original epipolar constraint $h_e(\overline{\mathbf{P}}^r, \{\mathbf{o}\})$ and the additional projection consistency constraint $h_c(\overline{\mathbf{P}}^r, s^r, \{\mathbf{p}\}, \{\mathbf{q}\}, \{d\})$ are:

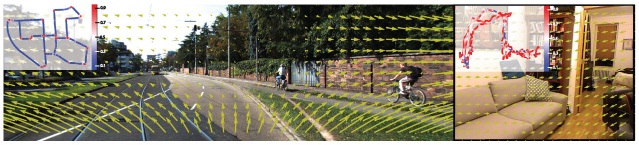
$$\begin{cases} h_e(\overline{\mathbf{P}}^r, \{\mathbf{o}\}) = \sum_{k=1}^{N_r} \left(\mathbf{q}_k^\intercal \mathbf{K}^{-\top} \mathbf{E} \mathbf{K}^\intercal \mathbf{p}_k < k_e\right) & \text{(10a)} \\ h_c(\overline{\mathbf{P}}^r, s^r, \{\mathbf{p}\}, \{\mathbf{q}\}, \{d\}) = & \\ \quad \sum_{k=1}^{N_r} \left(\|f(d_k, p(\overline{\mathbf{P}}^r, s^r)) - \mathbf{q}_k\|^2 < k_c\right). & \text{(10b)} \end{cases}$$

Here $\mathbf{E}$ is an essential matrix, expressed by the matrix form of the cross product $[]_\times$ as $\mathbf{E} = \mathbf{R}[\mathbf{t}]_\times$. The final consensus check number is a weighted summation of the two as $h(\overline{\mathbf{P}}^r) = h_e(\cdot) + \lambda \cdot h_c(\cdot)$.

The optimal normalized pose $\overline{\mathbf{P}}^\dagger$ and scale $s^\dagger$ is selected with the highest consensus number. The RANSAC stop criteria are updated with the new constraint $h(\cdot)$.

Key-frame Search. In Fig. 5, scene depth becomes irrelevant with scene motion under an extreme pure rotation movement. Without the loss of generality, more 3D information is revealed from two-view triangulation as the camera translation *a.k.a.*, baseline, increases. For video captured by a moving platform or a service robot, *e.g.*, KITTI dataset, there typically exists sufficient camera translation between consecutive frames. However, the camera rotation frequently dominates the movement for the video taken by a hand-held camera, *e.g.*, NYUv2 and ScanNet dataset. We alleviate the issue by actively seeking sufficient camera translation. Automatically, as in Fig. 3, we repeat the flow initialization step and pose estimation step with the next frame if the estimated scale $s^\dagger < k_s$, where k_s is a predefined minimum translation.

Scale Update. The camera scale $s^\dagger$ will be updated with the finetuned video depthmap $\mathbf{D}^\dagger$ using Eqn. 8 and Eqn. 9 if odometry is desired.



(a) KITTI Flow (b) NYUv2 Flow

Figure 5. Outdoor video motion patterns differ from indoor. Marked in yellow arrows, we visualize an indoor and outdoor scene motion. In (a), a translation dominates the scene motion. In (b), a rotation dominates the scene motion. Comparing (a) and (b), as rotation accumulates, the flow becomes irrelevant to scene depth, making image clues less usable for depth. Further, it degenerates the nonlinear projection transformation to the linear affine transformation, undermining the epipolar constraint based five-point algorithm. We thus introduce the additional projection constraint h_c in Eqn. 10. Further, we actively seek keyframes until sufficient translation movement is detected. We plot the entire odometry on the corner of (a) and (b). As the color changes from blue to red, more scene motion is from the rotation movement.

3.2. Video Depth Estimation

To this end, we have optimized Eqn. 3a. To optimize Eqn. 3b in inference, we adopt a cost volume based network, taking in an initial monocular depthmap $\mathbf{D}^*$, predicted pose $\mathbf{P}^\dagger = p(\overline{\mathbf{P}}^\dagger, s^\dagger)$ and a frame pair $\mathbf{I}_m/\mathbf{I}_n$ (see Fig. 3). We consider video depth estimation a log space residual learning over its monocular depth initialization $\mathbf{D}^*$. The meaning of residual is two-fold.

Construct Cost Volume $\mathcal{V}_D$. We sample residual depth candidates $\mathcal{D}$ of size $k_\mathcal{D}$ around initial monocular depthmap $\mathbf{D}^*$ with predefined interval Δd as:

$$\mathcal{D} = \left\{\mathbf{D}_i \parallel \mathbf{D}_i = \exp(\Delta d_i) \cdot \mathbf{D}^*\right\}_{i=1}^{k_\mathcal{D}}. \tag{11}$$

We then sample feature map $\mathbf{F}_n$ according to $\mathcal{D}$ and predicted pose $\mathbf{P}$ as:

$$\mathcal{F}_d^* = \left\{\mathbf{F}_i^* \parallel \mathbf{F}_i^* = g(f(\mathbf{D}_i, \mathbf{P}), \mathbf{F}_n)\right\}_{i=1}^{k_\mathcal{D}}. \tag{12}$$

$\mathcal{V}_D$ is then constructed by stacking $\mathcal{F}_d^*$ and the repetition of input feature $\mathbf{F}_n$, illustrated in Fig. 6.

Estimate Residual Depth. The cost volume is decoded by ResDepth network $\mathbb{D}$, yielding a log space residual depthmap $\Delta\mathbf{D}$ for monocular initial $\mathbf{D}^*$, preparing the final video depthmap $\mathbf{D}$ as:

$$\mathbf{D}^\dagger = \mathbf{D}^* \cdot \exp(\mathbf{\Delta D}) = \mathbf{D}^* \cdot \exp(\mathbb{D}(\mathcal{V}_D)). \tag{13}$$

Supervision Signal. Following [27], we use a scale-invariant loss, to supervise the training of the depth network,

$$D(w) = \frac{1}{n}\sum_{i=1}^{n} w_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} w_i\right)^2 + (1-\mu)\left(\frac{1}{n}\sum_{i=1}^{n} w_i\right)^2, \tag{14}$$

where $w_i = \log d_i - \log \tilde{d}_i$, n is the number of pixels and $\tilde{d}_i$ is groundtruth depth.

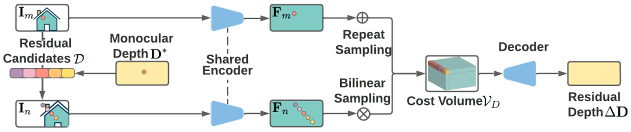

$\mathbf{I}_m$
Residual Candidates $\mathcal{D}$
Monocular Depth $\mathbf{D}^*$
Shared Encoder
$\mathbf{F}_m$
$\mathbf{F}_n$
Repeat Sampling
Bilinear Sampling
Cost Volume $\mathcal{V}_D$
Decoder
Residual Depth $\Delta\mathbf{D}$
$\mathbf{I}_n$

Figure 6. Illustration of video depth estimation. The shared encoder is drawn as one for simplicity in Fig. 3. The encoder and decoder of video depth network $\mathbb{D}$ are plotted. We dynamically sample the residual depth candidates $\mathcal{D}$ in log space centering around the initial depthmap $\mathbf{D}^*$. Then we construct cost volume $\mathcal{V}_D$ with predicted normalized pose $\overline{\mathbf{p}}^\dagger$ and the aligned scale $s^\dagger$. Finally, we predict residual depth $\Delta\mathbf{D}$ in log space through network $\mathbb{D}$.

3.3. Equality of Scale and Video Depth Learning

In Fig. 3, scale is required before video depth estimation. Though scale can be optimized over an initial monodepthmap, augmenting it with a network seems a natural choice. In this section, we show the *equality* of video depth and scale learning and its implication to the choice of scale estimation. Following Eqn. 7, we define the optimal scale s^* as the average of pixel-wise scale $\mathbf{s}$:

$$\log(s^*) = \frac{1}{n}\sum_{i=1}^{n}\log(s_i) = \frac{1}{n}\sum_{i=1}^{n}\big(\log(d_i) + m_i\big). \quad (15)$$

We then show that the learning objective for scale s^* can be approximated as the learning objective for video depth and a noise term contributed by normalized pose $\overline{\mathbf{P}}$ and optical flow $\mathbf{O}$ estimate:

$$\begin{aligned} L_{s^*} = &\| \log(\tilde{s}) - \log(s^*)\| \\ \leq &\frac{1}{n}\sum_{i=1}^{n}\| \log(\tilde{d}_i) - \log(d_i)\| + \|\frac{1}{n}\sum_{i=1}^{n}(\tilde{m}_i - m_i)\|. \end{aligned} \quad (16)$$

Here, $\tilde{s}$ and $\tilde{d}$ are groundtruth scale and depth. Estimating scale, by minimizing L_{s^*}, can be approximately achieved by minimizing its upper-bound in Eqn. 16, thus converting to video depth estimation. This indicates that a deep scale estimator learns the same prior knowledge as a video depth estimator. We empirically support our analysis by showing that the framework in Supp Fig. 1 has no benefit in final depth and scale performance, as in Tab. 5.

4. Experiments

We evaluate depth on KITTI and NYUv2 where both video and monocular depth methods report their results. We conduct indoor pose comparison on ScanNet as NYUv2 does not have pose groundtruth.

Implementation Details For both KITTI and NYUv2 experiments, we train with the Adam optimizer [24] with a learning rate of $1e^{-4}$. The training takes 20 epochs with a batch size of 4. We train 2 days on 2 RTX 2080 Ti GPUs. For the pre-computed initial monocular depthmap, we apply color augmentation to ensure consistent performance between validation and training set. We use BTS [27] during training but test against various mono-depth inputs. For all three monocular methods, BTS [27], AdaBins [2], and NewCRFs [58], we use the author released models. The Monodepth2 [18] is re-trained by us. For flow, we adopt the publicly available model of RAFT [42] trained using the synthetic datasets [33]. On KITTI, we train with a cropped 320×576 resolution. On NYUv2, we train with the original resolution. For both datasets, we test with their full resolution. The residual depth candidates $\mathcal{D}$ with a size of $k_{\mathcal{D}} = 32$. While selecting the random correspondences from flowmap for pose estimation, we do not apply forward-backward consistency [60] as the improvement does not worth its running time. But we exclude the invisible area and object edges in the next views. We use the OpenCV's EPnP [28] algorithm as a replacement if the five-point algorithms fail.

4.1. Monocular Video Depth and Pose Estimation

KITTI Depth KITTI is a widely adopted benchmark for outdoor scenes with stereo, LiDAR, and GPS/IMU available. For fair comparison, we train with Eigen split [9], evaluated on semi-dense groundtruth [45] under Garg crop [15] capped at 80 meters. Tab. 1 reports results in standard 7 metrics [9], with baselines from both single-view and multi-view methods. We outperform all of them by a substantial margin. Particularly, compared to 2-view methods [20, 50], our method significantly reduces 66.7% and 77.3% errors on the $a1$ metric ($\delta < 1.25$). Additionally, we are the first 2-view work to outperform the 5-view SoTA performance [41], achieving a substantial improvement of 60.9% ($= \frac{0.991-0.977}{1-0.977}$) on $a1$ metric. Further, we reduce 70.5% $a1$ metric error compared to our mono-depth initialization BTS. Fig. 7 shows our improvement qualitatively. Finally, our performance gain over prior SoTA does not attribute to monocular initialization. In Tab. 1, our result still substantially outperforms DeepV2D with a lightweight MonoDepth2 monocular initialization.

NYUv2 Depth NYUv2 dataset [35] has RGB and depth image pairs in indoor environments. Our experiment follows the standard train/test split [9]. As NYUv2 is captured by a handheld camera, rotation frequently dominates camera motion across frames, which is undesirable for video depth estimation (see Fig. 5). Despite all the hurdles, our 2-view performance grouped with NewCRFs [58] still substantially outperforms 8-view DeepV2D, reducing 34.1% error on $a1$ metric. Compared to its 2-view performance, the improvement goes up to 46.3%.

Further, our method shows great generalization ability under different mono-initialization. Though trained with BTS, when tested with BTS, AdaBins, and NewCRFs, we reduce error on $a1$ metric by 15.7%, 20.6%, and 16.7%, respectively. However, this performance gain is less than

Method	Venue	Frame	Labels	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
DORN [14]	CVPR'18	1	D	0.069	0.300	2.857	0.112	0.945	0.998	0.996
BTS [27]	Arxiv'18	1	D	0.059	0.245	2.756	0.096	0.956	0.993	**0.998**
AdaBins [2]	CVPR'21	1	D	0.058	0.190	2.360	0.088	0.964	0.995	**0.999**
NeWCRFs [58]	CVPR'22	1	D	**0.052**	**0.155**	**2.129**	**0.079**	**0.974**	**0.997**	**0.999**
Ours + BTS [27]	CVPR'23	2	D+F	**0.037**	0.110	1.809	**0.059**	0.987	**0.998**	**0.999**
Ours + AdaBins [2]		2	D+F	0.045	0.108	1.817	0.064	0.987	**0.998**	**0.999**
Ours + NeWCRFs [58]		2	D+F	0.041	**0.107**	**1.748**	**0.059**	**0.989**	**0.998**	**0.999**
BA-Net [40]	ICLR'19	5	D+P	0.083	0.025	3.640	0.134	-	-	-
SfMR [50]	CVPR'21	2	D+F+P	0.055	0.224	2.273	0.091	0.956	0.984	0.993
DeepMLE [8]	Arxiv'22	2	D+F+P	0.060	0.203	2.257	0.089	0.967	**0.995**	**0.999**
DRO [20]	Arxiv'21	2	D+P	0.047	0.199	2.629	0.082	0.970	0.994	0.998
MaGNet [1]	CVPR'22	3	D	0.051	**0.160**	2.077	0.079	0.974	**0.995**	**0.999**
DeepV2D [41]	ICLR'20	2	D+P	0.064	0.350	2.964	0.120	0.946	0.982	0.991
		5	D+P	**0.037**	0.174	2.005	0.074	0.977	0.993	0.997
DeepV2cD [22]	ICPRAI'22	5	D+P	**0.037**	0.167	**1.984**	**0.073**	**0.978**	0.994	-
Ours + MonoDepth2 [18]	CVPR'23	2	D+F	0.032	0.106	1.889	0.057	0.986	**0.998**	**0.999**
Ours + BTS [27]		2	D+F	0.029	0.098	1.729	0.053	0.989	**0.998**	**0.999**
Ours + AdaBins [2]		2	D+F	0.030	0.089	1.655	0.052	0.989	**0.998**	**0.999**
Ours + NeWCRFs [58]		2	D+F	**0.028**	**0.087**	**1.597**	**0.049**	**0.991**	**0.998**	**0.999**

Table 1. **KITTI Monocular Video Depth Evaluation** on Eigen split [9] with Garg crop [15] capped at 80 meters using semi-dense groundtruth [45]. The lower half table applies median scaling [62] to the predicted depths to compare with SfM methods. [Key: **Best**, **Second Best** except our work, Frame=the number of frames used in inference, Labels=required supervision in training, D=semi-dense depthmap, P=IMU pose, F=synthetic optical flow datasets [4, 33]]

Method	Venue	Frame	Abs Rel	Sc Inv	RMSE	log10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
DORN [14]	CVPR'18	1	0.115	-	0.509	-	0.828	0.965	0.992
BTS [27]	Arxiv'18	1	0.108	0.115	0.404	0.047	0.885	0.978	0.994
AdaBins [2]	CVPR'21	1	0.103	0.106	0.370	0.044	0.903	0.983	0.997
NewCRFs [58]	CVPR'22	1	**0.095**	**0.090**	**0.334**	**0.041**	**0.922**	**0.992**	**0.998**
Ours + BTS [27]	CVPR'23	2	0.102	0.098	0.356	0.044	0.903	0.984	0.997
Ours + AdaBins [2]		2	0.095	0.089	0.326	0.040	0.923	0.990	0.998
Ours + NewCRFs [58]		2	**0.090**	**0.080**	**0.306**	**0.038**	**0.935**	**0.995**	**0.999**
DfUSMC [21]	CVPR'16	Multi	0.447	0.456	1.793	0.169	0.487	0.697	0.814
DeMoN [46]	CVPR'17	2	0.144	0.179	0.775	0.061	0.805	0.951	0.985
DeepV2D [41]	ICLR'20	2	0.094	0.133	0.521	0.403	0.905	0.975	0.992
		9	**0.061**	**0.094**	**0.403**	**0.026**	**0.956**	**0.989**	**0.996**
Ours + BTS [27]	CVPR'23	2	0.070	0.098	0.280	0.030	0.948	0.991	0.998
Ours + AdaBins [2]		2	0.064	0.089	0.255	0.027	0.961	0.994	0.999
Ours + NewCRFs [58]		2	**0.057**	**0.080**	**0.230**	**0.025**	**0.971**	**0.996**	**0.999**

Table 2. **NYUv2 Monocular Video Depth Evaluation.** Results in the lower half table apply median scaling in evaluation. Results of DeMoN [46] is from [41]. Results of 2-view DeepV2D [41] are evaluated with the published code and pretrained model. [Key: **Best**, **Second Best**, Frame=the number of frames in inference]

Seq	Err	BetterGen* [60]	LTMVO* [64]	DfVWild* [19]	MLF-VO [23]	SfMR [50]	LSR* † [47]	Ours
09	t_{err}	6.03	3.49	3.10	3.90	1.70	**1.19**	**1.08** ± 0.07
	r_{err}	0.44	1.03	-	1.41	0.48	**0.30**	**0.28** ± 0.02
10	t_{err}	4.66	5.81	5.40	4.88	1.49	**1.34**	**1.29** ± 0.04
	r_{err}	0.62	1.82	-	1.38	0.55	**0.37**	**0.36** ± 0.02

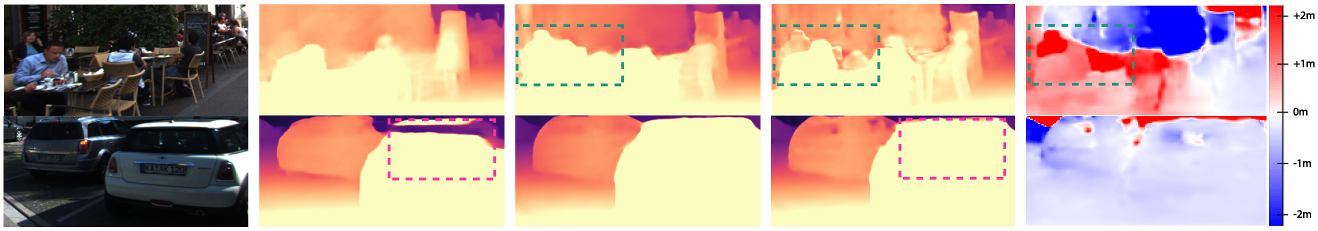
Seq	Err	DeepV2d [41]	Ours
00	t_{err}	**3.80**	**1.19** ± 0.04
	r_{err}	**1.66**	**0.39** ± 0.02
05	t_{err}	**3.25**	**1.36** ± 0.05
	r_{err}	**1.34**	**0.40** ± 0.03

Table 3. **KITTI Odometry Evaluation**. Results in the right of the table are trained on Eigen split [9] and tested on odometry sequence 00 and 05. Performance is reported with 5 random runs. Self-supervised methods are marked with *. † uses test time parameter fine-tuning (PFT) [47]. [Key: **Best**, **Second Best**]

in KITTI (15.7% to 70.5%), indicating our method shines more on videos with sufficient translation.

KITTI Pose KITTI Odometry includes 20 driving videos with 11 having odometry groundtruth. Our experiment includes both self-supervised and supervised methods and reports standard metrics [19]. For methods [19, 23, 47, 50, 60, 64], we follow [19] to train/test on sequences 00-08/09-10. For DeepV2D [41], as trained on Eigen split [9], we test on unseen sequences 00 and 05. As odometry from self-supervised methods lacks real-world scale priors, we align prediction against groundtruth trajectory by applying 7 DoF transformation [60] during inference. In Tab. 3, we outperform SoTA on rotation and translation errors.

ScanNet Pose ScanNet [5] is a large indoor dataset with groundtruth depthmap and camera trajectory. We follow DeepV2D's test protocol, train on NYUv2, and test on 2,000 sequences of ScanNet. We outperform 8 frames DeepV2D-8 except for the metric 'tr. (deg)'. Further, our method achieves solid improvement over 2-view DeepV2D.



(a) RGB Input $\mathbf{I}_t$ (b) DeepV2D [41] (c) Init $\mathbf{D}^*$ (d) Finetuned $\mathbf{D}^\dagger$ (e) Residual (meter)

Figure 7. Subplot (e) shows residual depth $\mathbf{D}^* \cdot (\exp(\Delta\mathbf{D}) - 1)$ in meter. In Green boxes, mono-depthmap gets improved after residual estimation. In Pink boxes, artifacts around moving foreground objects are avoided.

ScanNet	DeMoN [46]	BA-Net [40]	DSO	DeepV2D-2	DeepV2D-8	FivePoint	Ours
Rotation (degree) ↓	3.791	1.009	0.946	0.806	**0.714**	0.671	**0.621** ± 0.007
Translation (degree) ↓	31.626	14.626	19.238	13.259	**12.205**	13.878	**12.840** ± 0.161
Translation (cm) ↓	15.500	2.365	2.165	1.726	**1.514**	1.524	**1.440** ± 0.011

Table 4. **ScanNet Pose Evaluation.** DeMoN, BA-Net, and DSO are trained on ScanNet. DSO is evaluated only on success cases. DeepV2D and ours are trained on NYUv2 and tested on ScanNet. DeepV2D-2/8 are DeepV2D taking 2 or 8 frames. FivePoint is the baseline five-point algorithm with RANSAC. Our result is reported with 5 random runs. [Key: **Best**, **Second Best**]

	ResDepth	PoesEstimation	ScaleNet	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	Seq-00 t_{err}
KITTI		✓		0.070	0.275	2.405	0.093	0.959	1.55
	✓	✓		0.038	**0.110**	**1.821**	0.060	**0.987**	1.55
	✓	✓	✓	**0.037**	0.117	1.841	**0.059**	0.986	**1.24**

Table 5. **Ablation on Outdoor Video Depth Estimation.** [Key: 'ResDepth'= Residual depth learning (Sec. 3.2). 'PoseEstimation'= Proposed Pose Estimation Method (Sec. 3.1). 'ScaleNet'=Further refine pose scale with an additional ScaleNet (detailed in Supplementary).]

	FivePoint	PoesEstimation	KeySearch	Abs Rel	Sc Inv	RMSE	log10	$\delta < 1.25$
NYUv2	✓			0.063	0.087	0.248	0.027	0.964
		✓		0.061	0.083	0.239	0.026	0.968
		✓	✓	**0.057**	**0.080**	**0.230**	**0.025**	**0.971**

Table 6. **Ablation on Indoor Video Depth Estimation.** [Key: 'FivePoint'=Baseline Five-point algorithm with RANSAC. 'PoseEstimation'=Proposed Pose Estimation Method (Sec. 3.1). 'KeySearch'=Keyframe search. Bold marks the best score.]

4.2. Ablation Study

The Equality between Scale and Video Depth Learning In Tab. 5 row 2 & 3, we ablate pose & depth performance if augment pose scale learning with an additional ScaleNet (detailed in Supplementary). Clearly, the added ScaleNet learns additional scale prior, reducing t_{err} from 1.55 to 1.24. However, the improved pose scale does not benefit video depth due to the equality between their learning objective. Further, this benefit diminishes after updating the scale with video depthmap (1.19 from Tab. 3 and 1.24 from Tab. 5). This is expected, as the LiDAR depth possesses less noise than IMU and GPS pose. Thus we empirically demonstrate the equality between scale and video depth learning.

Residual Depth Estimation Estimating video depth as logged residual improves cost volume sampling efficiency, supported by our improvement over SfMR [50] in Tab. 1 and the performance gap in row 1 and 2 of Tab. 5. Meanwhile, it avoids artifacts in moving objects, as in Fig. 7.

Pose Estimation and Key-frame Search Compared to using baseline five-point algorithm over flow estimate [50,60], our proposed method benefits both pose and depth performance, as shown in Tabs. 4 and 6. Also, ensuring sufficient camera translation shows noticeable improvement, as shown in Tab. 6.

Computational Efficiency We compare the running time to DeepV2D [41] on an RTX 2080 Ti GPU, for 192 × 1088 images. In Fig. 3, our inference has $1 + 2$ steps: initialization of flow [42] and mono-depth [27], pose estimation, and video depth estimation. Each takes $0.124 + 0.063$, 0.253, 0.058s respectively, in total 0.498s. In comparison, 5-view DeepV2D takes 1.619s.

5. Conclusions

Video depth estimation in prior works is solved as a simplified SfM problem. But video depth has fewer view angles in video depth estimation. Thus, we decompose it into two sub-tasks that are robust to deficient views, *i.e.*, normalized pose, and residual depth estimation. We connect the two tasks with a scale alignment algorithm. The proposed framework improves both pose and video depth.

Limitations Our method depends on multiple modality initializations. A joint model is preferred.

References

[1] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *CVPR*, 2022. 1, 3, 7

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021. 6, 7

[3] Julius Butime, Iñigo Gutierrez, L Galo Corzo, and C Flores Espronceda. 3d reconstruction methods, a survey. In *VISAPP*, 2006. 1

[4] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 7

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *PAMI*, pages 5828–5839, 2017. 7

[6] Yuchao Dai, Zhidong Zhu, Zhibo Rao, and Bo Li. Mvs2: Deep unsupervised multi-view stereo with multi-view symmetry. In *3DV*, 2019. 3

[7] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE PAMI*, 2007. 3

[8] Marcio L Lima de Oliveira and Marco JG Bekooij. Deepmle: Fusion between a neural network and mle for a single snapshot doa estimation. In *ICASSP*, 2022. 1, 7

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 6, 7

[10] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014. 3

[11] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *ICCV*, 2013. 3

[12] Tuo Feng and Dongbing Gu. Sganvo: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks. *RA-L*, 2019. 3

[13] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 2, 4

[14] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 7

[15] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 6, 7

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[17] Riccardo Gherardi, Michela Farenzena, and Andrea Fusiello. Improving the efficiency of hierarchical structure-and-motion. In *CVPR*, 2010. 1, 3

[18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 6, 7

[19] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *ICCV*, 2019. 7

[20] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, Chengzhou Tang, and Ping Tan. Dro: Deep recurrent optimizer for structure-from-motion. *arXiv preprint arXiv:2103.13201*, 2021. 1, 3, 6, 7

[21] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality depth from uncalibrated small motion clip. In *CVPR*, 2016. 7

[22] Christian Homeyer, Oliver Lange, and Christoph Schnörr. Multi-view monocular depth and uncertainty prediction with deep sfm in dynamic environments. In *ICPRAI*, 2022. 1, 3, 7

[23] Zijie Jiang, Hajime Taira, Naoyuki Miyashita, and Masatoshi Okutomi. Self-supervised ego-motion estimation based on multi-layer fusion of rgb and inferred depth. *ICRA*, 2022. 7

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2015. 6

[25] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, 2007. 3

[26] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *IJRR*, 2013. 1

[27] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 1, 5, 6, 7, 8

[28] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 2009. 6

[29] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *ICPR*, 2006. 2, 4

[30] Xiaoxiao Long, Lingjie Liu, Wei Li, Christian Theobalt, and Wenping Wang. Multi-view depth estimation using epipolar spatio-temporal networks. In *CVPR*, 2021. 3

[31] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981. 1, 3

[32] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 3

[33] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 6, 7

[34] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 2015. 3

[35] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 6

[36] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 3

[37] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. In *CVPR*, 2016. 1, 3

[38] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 3

[39] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular slam: Why filter? In *ICRA*, 2010. 3

[40] Chengzhou Tang and Ping Tan. BA-net: Dense bundle adjustment networks. In *ICLR*, 2019. 2, 3, 7, 8

[41] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *ICLR*, 2020. 1, 2, 3, 6, 7, 8

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 6, 8

[43] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *NeurIPS*, 2021. 3

[44] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 2002. 3

[45] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, 2017. 6, 7

[46] 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 *CVPR*, 2017. 1, 2, 3, 7, 8

[47] Brandon Wagstaff, Valentin Peretroukhin, and Jonathan Kelly. Self-supervised structure-from-motion through tightly-coupled depth and egomotion networks. *arXiv preprint arXiv:2106.04007*, 2021. 7

[48] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Itermvs: Iterative probability estimation for efficient multi-view stereo. 2022. 3

[49] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, 2021. 3

[50] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. *CVPR*, 2021. 1, 3, 6, 7, 8

[51] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 3

[52] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *ECCV*, 2020. 1, 2, 3

[53] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 1999. 3

[54] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2011. 1, 3

[55] Yuxi Xiao, Li Li, Xiaodi Li, and Jian Yao. Deepmle: A robust deep maximum likelihood estimator for two-view structure from motion. *IROS*, 2022. 1

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

[57] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, 2019. 3

[58] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. New crfs: Neural window fully-connected crfs for monocular depth estimation. 2022. 6, 7

[59] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 2020. 1

[60] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *CVPR*, 2020. 2, 6, 7, 8

[61] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *ECCV*, 2018. 3

[62] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 7

[63] Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *CVPR*, 2020. 3

[64] Yuliang Zou, Pan Ji, Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. Learning monocular visual odometry via self-supervised long-term modeling. In *ECCV*, 2020. 7