

# Uncertainty-Driven Dense Two-View Structure from Motion

Weirong Chen, Suryansh Kumar<sup>†</sup>, *Member, IEEE*, Fisher Yu, *Member, IEEE*

**Abstract**—This work introduces an effective and practical solution to the dense two-view structure from motion (SfM) problem. One vital question addressed is how to mindfully use per-pixel optical flow correspondence between two frames for accurate pose estimation—as perfect per-pixel correspondence between two images is difficult, if not impossible, to establish. With the carefully estimated camera pose and predicted per-pixel optical flow correspondences, a dense depth of the scene is computed. Later, an iterative refinement procedure is introduced to further improve optical flow matching confidence, camera pose, and depth, exploiting their inherent dependency in rigid SfM. The fundamental idea presented is to benefit from per-pixel uncertainty in the optical flow estimation and provide robustness to the dense SfM system via an online refinement. Concretely, we introduce our uncertainty-driven Dense Two-View SfM pipeline (DTV-SfM), consisting of an uncertainty-aware dense optical flow estimation approach that provides per-pixel correspondence with their confidence score of matching; a weighted dense bundle adjustment formulation that depends on optical flow uncertainty and bidirectional optical flow consistency to refine both pose and depth; a depth estimation network that considers its consistency with the estimated poses and optical flow respecting epipolar constraint. Extensive experiments show that the proposed approach achieves remarkable depth accuracy and state-of-the-art camera pose results superseding SuperPoint and SuperGlue accuracy when tested on benchmark datasets such as DeMoN, YFCC100M, and ScanNet. Code and more materials are available at <http://vis.xyz/pub/dtv-sfm>.

**Index Terms**—Dense Structure from Motion, Uncertainty Prediction, Optical Flow, Weighted Bundle Adjustment.

## I. INTRODUCTION

The dense two-view structure from motion (SfM) problem generally aims at recovering the relative camera motion between two frames and the per-pixel 3D position of a rigid scene. A reliable solution to this problem can be helpful in several visual automation, and robotics applications [1] [2], since it forms the basic building block of many large-scale 3D reconstruction pipelines [3] [4]. Unfortunately, as of today, the credible SfM systems are confined to sparse 3D reconstruction [4]. Yet, modern applications in mixed reality [5], robot vision [6] [7], motion-capture [8] [9] and others wish for a reliable dense 3D acquisition and camera pose estimation from images.

One of the leading reasons for the absence of a dense two-view SfM algorithm in practice, contrary to the sparse one,

is that it is relatively simple to analyze a few pixel correspondences, validate the matching accuracy, and be scalable. In other words, sparse pixel correspondence can be considered conditionally independent, making it statistically convenient to justify its reliability in camera pose and sparse 3D reconstruction [10] [11]. As a result, it is easy to use for multi-view cases. On the contrary, modeling per-pixel correspondences between two images for solving dense SfM are often challenging [12]. Practically, modeling per-pixel matching and assessing its consistency with camera pose and scene depth is non-trivial. It can seriously influence the camera pose and depth estimation accuracy if not modeled aptly by the algorithm. Also, we must be careful about the pixel correspondence selection for pose estimation, as five distinct correct correspondences are theoretically sufficient for fast and accurate camera pose estimation [13]. Additionally, it is relatively challenging—both computationally and algorithmically—to extend dense two-view SfM for multi-view cases. Yet, the dense two-view case shall serve as a foundational pipeline for multi-view cases hence, important for research.

Meanwhile, recent deep-learning approaches can address some of the limitations of dense SfM pipelines by learning the camera motion and scene geometry priors in a supervised setting. In this regard, some recent works focus on improving the pixel correspondence quality with deep neural networks (DNNs). Their key motivation is to learn suitable feature representations and reliable matching from large-scale datasets [14] [15] [16] [17]. With neural networks being the universal function approximators [18], another trend is to overlook the geometric two-view SfM pipeline [19] and directly regress the relative camera pose through the neural networks, which are usually trained jointly with another depth network in a supervised [20] [21] or self-supervised setting [22] [23] [24]. While these learning-based methods show encouraging results, most are not careful about the pixel correspondence reliability for pose estimation and how the predicted poses can affect the overall depth estimation. Even state-of-the-art deep learning-based methods mostly use well-known supervised learning pipelines and ignore the measure of correctness of the predicted pose and its impact on depth estimates [21] [25]. These matters, or even the confidence of dense correspondence prediction, have often not been considered fully in deep learning-based dense two-view SfM problems [25]. To mitigate the aforementioned challenges and resolve the prevailing issues in solving this problem, we propose a simple, accurate, and systematic learning-based dense two-view SfM approach.

The proposed approach benefits from both the classical multi-view geometry theory and learning-based confidence

Manuscript received: September, 21, 2022; Revised December, 20, 2022; Accepted January, 19, 2023. This paper was recommended for publication by Editor Cesar Cadena Lerma upon evaluation of the Associate Editor and Reviewers' comments.

All the authors are with VIS Group at ETH Zürich, 8092 Zürich, Switzerland (email: wrchen530@gmail.com; k.sur46@gmail.com; i@yf.io).

<sup>†</sup>Corresponding Author: Suryansh Kumar.

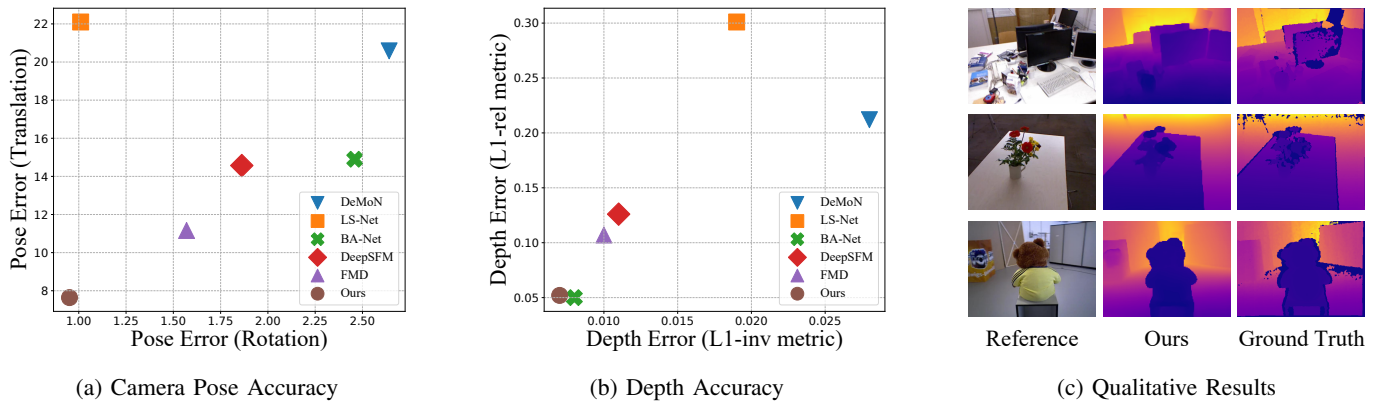


Fig. 1: Comparison with state-of-the-art learning-based dense two-view SfM methods [20] [21] [26] [27] [28] on RGBD dataset [29]. (a)-(b) Our approach shows significant boosts in the camera pose accuracy results; at the same time, we have high-quality depth estimation results. (c) Qualitative depth estimation results compared to the respective ground truth depth map.

modeling. For camera pose estimation with known intrinsics, we introduce an uncertainty-aware pose estimation module that considers matching confidence and robust outlier filtering for camera pose recovery at test time. On one hand, with the advancement of learning-based uncertainty modeling, it is easy and convenient to model the matching uncertainty in the network design itself rather than optimize it separately. On the other hand, conventional methods like RANSAC [10] and other consensus set maximization methods [30] [31] can perform outlier rejection and robust pose recovery, especially for the large baseline case where optical flow estimation is unreliable. As we demonstrate statistically later in the paper, these two methods complement each other and provide state-of-the-art camera pose estimation results when tested on a diverse benchmark dataset (see Fig.1 (a)).

For depth estimation, existing methods typically build a cost volume to aggregate multi-view features given the camera pose and later apply multiple 3D convolutions to obtain the depth [21] [25]. On the contrary, our approach utilizes monocular and cross-view image feature cues and their consistency with the estimated poses and per-pixel matching, respecting the well-founded epipolar constraint. Furthermore, most of the existing deep-learning methods, if not all, compile their results once the camera pose and depth of the scene are recovered [25] [26] [28]. Nonetheless, to enhance the performance, in this work, we propose taking the learning-based dense SfM pipeline a step further. Notably, our work shows that we can improve camera pose and depth results by exploiting the dependency of optical flow, depth, and camera pose in rigid SfM. This dependency loop of per-pixel correspondence confidence score, pose, and depth is encapsulated in an iterative optimization and solved until convergence. To put it simply, we compute per-pixel correspondences with the predicted pose and depth by projecting the 3D points onto the other image. We call it “induced optical flow”. The induced optical flow provides additional evidence about the dense correspondences between frames. We update the earlier predicted optical flow and flow confidence by checking the consistency between the induced and predicted optical flow. The updated flow and flow matching confidence provides better cues for our

weighted bundle adjustment formulation—refer to Sec.III-B, which helps refine the camera pose and depth estimates.

**Contributions.** To summarize, our key contributions are:

- This paper proposes an accurate and reliable dense two-view SfM system. The proposed method utilizes the fundamentals of rigid SfM with mindful statistical modeling of per-pixel optical flow matching leading to state-of-the-art camera pose estimates and favorable scene depth recovery.
- It is demonstrated that excellent camera pose can be recovered by using uncertainty-aware dense optical flow correspondence in addition to the popular statistical outlier rejection approach. Hence, we demonstrate that flow based uncertainty-driven weighted bundle adjustment (WBA) with outlier filtering and bidirectional flow consistency is a simple yet effective tool for camera pose recovery.
- If not all, most methods conclude their results once camera pose and depth are estimated. To this end, an iterative update scheme is introduced to improve the estimated camera pose and depth further. This later stage update is done by refining the dense optical flow and flow confidence via induced optical flow consistency.

## II. RELATED WORK

Classical two-view SfM has been extensively studied over the past few decades. While a robust, practical, and reliable two-view SfM system is still limited to a sparse set of image points, a consistent effort to extend it to a dense SfM system can be observed, especially with the advancement in deep neural network architectures [21] [32]. Thus, we survey the existing works by dividing them into classical and deep learning-based methods for better understanding.

**Classical two-view SfM.** These approaches follow the well-founded projective geometry rules and formulation developed in the early 1980s-1990s [19]. A standard pipeline for two-view SfM is (a) for each image compute the features and descriptors; (b) find the descriptor matching between the two images and filter the wrong matches using outlier rejection methods [10] [30] [31]; (c) estimate camera-pose and 3D structure using those matches [13]. Such a pipeline has developed over the years and put to use in many real-world applications,

mainly due to the advancement in feature descriptor [11] [33], optical flow matches [14] [34], improved optimization techniques [3] and scalable data structures [4]. Since the classical two-view SfM pipeline depends on well-behaved imaging conditions, its performance can degrade drastically for texture-less, specular, and non-diffuse objects in the scene. Recent developments in deep neural networks can help resolve such a limitation with the classical pipeline, which brings us to our next discussion on two-view SfM methods.

**Deep learning for SfM.** For better insight, we briefly discuss the key deep learning methods under two sub-categories.

(i) *Self-supervised methods.* These methods estimate the camera pose and depth without using ground truth supervision at train time. Here, the common approach is to use the view synthesis-based image consistency loss function to train the model, *i.e.*, one should be able to reconstruct the next image from the previous image, given the correct camera parameters, pose, and depth map [35]. Self-supervised methods generally adopt a monocular depth estimation network and a pose regression network that can be trained with the monocular image sequence alone [12] [22] [23] [24]. However, self-supervised methods rely heavily on imaging priors, the correctness of the predicted pose and therefore, often have difficulty handling challenging environments.

(ii) *Supervised methods.* Contrary to self-supervised approaches, these methods rely on ground truth scene depth and camera poses at train time. By now, it is widely accepted that the key to estimating better camera pose and depth largely depends on per-pixel correspondence accuracy. Following this, several works focus on improving correspondences through learning-based methods [15] [16] [17] [36] [37] [38]. However, it is difficult for a deep-learning framework to directly adopt the classical SfM idea of using a robust outlier rejection algorithm for robust camera pose estimation. Consequently, pose regression networks have been proposed in the past, which are fully differentiable by design and can be trained end-to-end. DeMoN [20] is one of the early methods that use a multi-scale encoder-decoder network to regress camera pose and depth map and iteratively refine them using dense optical flow estimates. Likewise, DeepSfM [21] transforms the camera pose and depth into two 3D volumes and performs iterative updates within the cost volume space. Some recent works also try to encode the optimization steps into the neural network as differentiable solvers. LS-Net [26] uses a deep neural network to predict the camera pose and depth by minimizing the photometric error, whereas BA-Net [27] uses the Levenberg-Marquardt algorithm for bundle adjustment as a differentiable layer with learnable damping factor and optimizes with the feature-metric error. More recently, DROID-SLAM [6] proposes an end-to-end system by integrating flow, confidence, and geometric optimization via an iterative GRU update module to solve monocular dense SLAM tasks.

### III. METHOD

**Problem Statement.** Given a consecutive pair of monocular images  $\mathbf{X} = (\mathbf{I}^r, \mathbf{I}^s)$  and the intrinsic camera calibration matrix  $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ , our goal is to estimate the relative camera pose

$\mathbf{T} \in SE(3)$  between those frames and a depth map  $\mathbf{D}^r \in \mathbb{R}^{H \times W}$  corresponding to  $\mathbf{I}^r$ . Here,  $H$  and  $W$  symbolize the image's height and width. We denote  $\mathbf{I}^r \in \mathbb{R}^{H \times W \times 3}$  and  $\mathbf{I}^s \in \mathbb{R}^{H \times W \times 3}$  as the reference image and source image, respectively.

Following the widely accepted SfM pipeline [4], we first seek an accurate per-pixel correspondence between the two frames. As alluded to above, it is a challenging task. Thus, we turn to modern deep-learning methods that have recently demonstrated outstanding results for the per-pixel correspondence problem. Nevertheless, for SfM it is equally important to reason about the correctness of the estimated pixel correspondence. Therefore, we adopt a dense optical flow network with uncertainty modeling, which at test time provides a per-pixel optical flow matching  $\mathbf{Y} \in \mathbb{R}^{H \times W \times 2}$  with a per-pixel confidence score matrix  $\mathbf{C} \in \mathbb{R}^{H \times W}$  of the predicted matching. Our uncertainty-aware optical flow estimation network helps in excellent camera pose recovery (Sec.III-A).

For camera pose estimation, we propose weighted bundle adjustment (WBA), where weights in WBA consider flow uncertainty, robust outlier filtering, and bidirectional flow consistency (Sec.III-B). The recovered camera pose  $\mathbf{T}$  along with  $\mathbf{X}$  are passed to the depth estimation network to predict  $\mathbf{D}^r$ . Unlike other methods [25] [28], our approach allows further improvement of  $\mathbf{T}$  and  $\mathbf{D}^r$  by iteratively updating flow  $\mathbf{Y}$  and flow confidence  $\mathbf{C}$  via induced optical flow due to the current estimate of  $\mathbf{T}$  and  $\mathbf{D}^r$  (Sec.III-C). Finally, an overall loss to train the network is presented in Sec.III-D. Next, we discuss our approach in more detail explaining the neural network design choice used in our overall pipeline (see Fig. 2).

#### A. Per-Pixel Correspondence Estimation

Reliability of per-pixel matching from  $\mathbf{I}^r$  to the corresponding pixel in  $\mathbf{I}^s$  is the key to dense two-view SfM problem. Yet, relative camera pose estimation needs a few accurate pixel matches. Since there is no trivial way to identify the correct per-pixel correspondences directly, it motivates us to explore a neural-network-based data-driven solution to reason about the fidelity of the predicted flow, so that we can *a priori* decide the suitability of the estimated correspondence. To this end, we exploit the dense correspondence method PDC-Net [15] as our optical flow backbone. Using this backbone model, we compute dense matches relating to two frames and the pixel-wise uncertainty map representing the correspondences' reliability. As outlined in Sec.III-B, such an idea can provide a better camera pose estimate than popular methods [39] [40]. We introduce the associated notations with a brief recap of the PDC-Net as follows.

In the paper, we use the term forward flow and backward flow denoted as  $\mathbf{Y}_{r \rightarrow s} \in \mathbb{R}^{H \times W \times 2}$  and  $\mathbf{Y}_{s \rightarrow r} \in \mathbb{R}^{H \times W \times 2}$ , for the optical flow estimated from  $\mathbf{I}^r$  to  $\mathbf{I}^s$  and vice-versa. We represent  $\mathbf{Y}_{r \rightarrow s} = F_{r \rightarrow s}(\mathbf{X}; \theta)$ ,  $\mathbf{Y}_{s \rightarrow r} = F_{s \rightarrow r}(\mathbf{X}; \theta)$ , where  $F(\mathbf{X}; \theta)$  symbolizes the neural network parameterized by  $\theta$ . For notation convenience, we use the symbol  $\mathbf{Y}$  for denoting optical flow in general and will explicitly denote the forward-backward flow symbol if required.

Given  $\mathbf{X}$ , instead of predicting the single flow vector values for each pixel, the goal is to predict the conditional



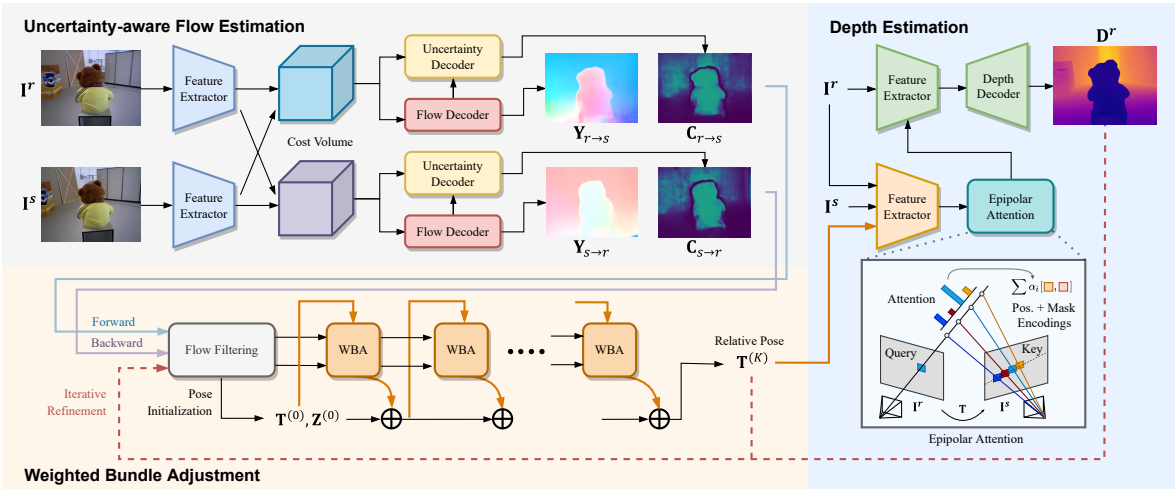


Fig. 2: An overview of our uncertainty-driven dense two-view SfM approach. We first predict the forward-backward dense optical flow correspondences and their confidence maps. Then, we solve the relative pose using the weighted bundle adjustment module with robust outlier rejection and bidirectional flow consistency information. Finally, we recover the dense scene depth using the predicted pose, which can update flow and flow confidence leading to further refinement of the camera pose and depth.

probability density of the optical flow as  $p(\mathbf{Y}|\mathbf{X};\theta)$ . Denoting  $\mathbf{y}_{ij} \in \mathbb{R}^{2 \times 1}$  and  $\phi_{ij} \in \mathbb{R}^{n \times 1}$  as the optical flow and predicted distribution parameter for the  $(i, j)$ <sup>th</sup> pixel, we write  $p(\mathbf{Y}|\mathbf{X};\theta) = p(\mathbf{Y}|\Phi(\mathbf{X};\theta)) = \prod_{ij} p(\mathbf{y}_{ij}|\phi_{ij}(\mathbf{X};\theta))$ . To estimate confidence value  $c \in [0, 1]$  for a pixel  $\mathbf{p}$  from the predicted distribution, the total probability of predicted optical flow within a radius  $R$  of the estimated mean flow  $\mu$  is computed as  $c = P(|y - \mu| < R) = \int_{\{y \in \mathbb{R}^2: |y - \mu| < R\}} p(y|\phi) dy$ . The confidence map  $\mathbf{C} = \{c_{ij}\} \forall (i, j) \in [H, W]$  can be obtained by calculating the confidence value for every pixel.

### B. Weighted Bundle Adjustment

With the dense correspondences, the next step is to estimate the relative camera pose  $\mathbf{T}$ . However, as the estimated correspondences are generally noisy and of varying quality, classical SfM pipelines mostly rely on off-the-shelf RANSAC algorithm [16] or other robust inlier maximization approaches [30] [31]. Without losing the robustness from outlier rejection, we further assess the quality of correspondence matches in the optical flow network design so that we can reason about correspondence inliers at test time. Such a hybrid idea is not only effective for camera pose estimation but also helps refine the correspondence’s measure itself. To realize such an idea, we introduce a weighted bundle adjustment module (WBA).

The proposed WBA optimizes for camera pose and depth such that the induced optical flow computed from the current estimation of camera pose and depth should be consistent with the network’s predicted flow. Despite such an idea has some notion of similarity with [6] that obtains weights from convolutional layers without probabilistic modeling, we consider the uncertainty obtained from a mixture of probability models with additional robust outlier rejection, bidirectional flow consistency, careful pose initialization, and independent depth estimation, which significantly improves the accuracy of pose and depth estimates—refer to Sec. IV.

Let  $\mathbf{p}_{ij} \in \mathbb{R}^{2 \times 1}$  and  $\bar{\mathbf{p}}_{ij} \in \mathbb{R}^{3 \times 1}$  denote  $(i, j)$ <sup>th</sup> image pixel coordinate and its homogeneous coordinate. We compute  $\mathbf{p}^s$ ,

*i.e.*, pixel in the source image corresponding to  $\mathbf{p}^r$  in the reference image with predicted flow  $\mathbf{Y}_{r \rightarrow s}$  as

$$\mathbf{p}_{\text{flow}}^s = \mathbf{Y}_{r \rightarrow s}(\mathbf{p}^r) + \mathbf{p}^r. \quad (1)$$

As stated, our aim of using WBA is to mindfully solve for camera pose; therefore, we propose exploiting the consistency between the induced flow and predicted flow. By induced flow, we mean the optical flow derived from the camera pose and the intermediate depth estimates. Assume  $\mathbf{T} = [\mathbf{R}|\mathbf{t}]$  as the relative camera pose from  $\mathbf{I}^r$  to  $\mathbf{I}^s$ , where  $\mathbf{R} \in SO(3)$ , and  $\mathbf{t} \in \mathbb{R}^{3 \times 1}$  represents the translation vector. Let  $\mathbf{Z} = (\mathbf{Z}^r, \mathbf{Z}^s)$ ,  $\mathbf{Z}^r \in \mathbb{R}^{H \times W}$ ,  $\mathbf{Z}^s \in \mathbb{R}^{H \times W}$  be the intermediate depth representations used in WBA corresponding to  $\mathbf{I}^r$  and  $\mathbf{I}^s$ , respectively. Now, we are ready to define the forward induced flow by back-projecting the pixel in  $\mathbf{I}^r$  to 3D space and projecting it back to  $\mathbf{I}^s$  using the  $\mathbf{T}$  and  $\mathbf{Z}^r$ . We compute the induced-flow corresponding pixel in the source image as

$$\bar{\mathbf{p}}_{\text{ind\_flow}}^s(\mathbf{T}, \mathbf{Z}^r) = \mathbf{K}(\mathbf{R} \cdot \mathbf{Z}^r(\mathbf{p}^r)) \cdot \mathbf{K}^{-1} \bar{\mathbf{p}}^r + \mathbf{t}. \quad (2)$$

Combining Eq.(1) and Eq.(2) gives us the reprojection error constraint, *i.e.*,  $\sum_{ij} \|\mathbf{p}_{\text{flow},ij}^s - \bar{\mathbf{p}}_{\text{ind\_flow},ij}^s(\mathbf{T}, \mathbf{Z}^r)\|^2$  which helps refine the intermediate depth and pose. Meanwhile, we also have prior information pertaining to the quality of optical flow, *i.e.*, the confidence map  $\mathbf{C}$ , which we have estimated using our uncertainty-aware flow estimation network (Sec III-A). While  $\mathbf{C}$  does provide self-contained evidence of the correspondence quality, we further boost the robustness by combining it with the robust outlier filtering from RANSAC. Therefore, we introduce an additional binary mask  $\mathbf{M} = \{m_{ij}\} \forall (i, j) \in [H, W]$  that includes high-value score  $\mathbf{C}$  as well as RANSAC inliers. Accordingly, we compute the binary mask as

$$m_{ij} = [c_{ij} \geq \gamma \text{ and } \mathbf{y}_{ij} \in A] \quad (3)$$

where  $\gamma$  is the confidence threshold,  $[ \cdot ]$  is the Iverson bracket, and  $A$  symbolizes the set of RANSAC inliers computed from  $\mathbf{Y}$ . Using the defined binary mask and  $c_{ij} \in \mathbf{C}$ , we define the

weights for our WBA as  $w_{ij} = m_{ij}c_{ij}$ . This leads us to our WBA formulation defined as

$$E_{r \rightarrow s}(\mathbf{T}, \mathbf{Z}^r) = \sum_{ij} w_{r \rightarrow s, ij} \|\mathbf{p}_{\text{flow}, ij}^s - \mathbf{p}_{\text{ind\_flow}, ij}^s(\mathbf{T}, \mathbf{Z}^r)\|^2. \quad (4)$$

Eq.(4) is due to forward induced optical flow. Similarly, we can compute backward induced optical flow and have one more constraint  $E_{s \rightarrow r}(\mathbf{T}^{-1}, \mathbf{Z}^s)$  for WBA. Combining both WBA constraints, we write the overall WBA objective as

$$\mathbf{T}, \mathbf{Z}^r, \mathbf{Z}^s = \arg \min_{\mathbf{T}, \mathbf{Z}^r, \mathbf{Z}^s} E_{r \rightarrow s}(\mathbf{T}, \mathbf{Z}^r) + E_{s \rightarrow r}(\mathbf{T}^{-1}, \mathbf{Z}^s). \quad (5)$$

We use the Gauss-Newton method [41] to optimize Eq.(5) for  $K$  iterations. The initial pose  $\mathbf{T}^{(0)}$  is obtained from the same RANSAC pose estimation step in the previous outlier filtering, and the initial depth map  $\mathbf{Z}^{(0)}$  is set to all value one. At the  $k^{\text{th}}$  iteration, we compute the camera pose update  $\Delta \xi^{(k)} \in \mathfrak{se}(3)$  (lie-algebra corresponding to  $\mathbf{T}$ ), and the depth update  $\Delta \mathbf{Z}^{(k)}$  for two depth maps. By vectorizing the camera pose and depth parameters, the updates are efficiently computed using the Schur decomposition [6]. The camera pose and depth are then updated via retraction on  $SE(3)$  and addition, respectively over the iterations as follows

$$\mathbf{T}^{(k+1)} = \exp(\Delta \xi^{(k)}) \circ \mathbf{T}^{(k)}, \quad \mathbf{Z}^{(k+1)} = \Delta \mathbf{Z}^{(k)} + \mathbf{Z}^{(k)}. \quad (6)$$

### C. Depth Estimation and Iterative Refinement

By now, we have detailed how we are predicting the uncertainty-aware dense optical flow and utilizing it for accurate pose estimation. In fact, those are critical to dense SfM nonetheless; the final goal is to recover an accurate depth map of the scene. To that end, we utilize the backbone architecture of MVS2D [42] with EfficientNet [43]. Given the estimated pose  $\mathbf{T}$  and  $\mathbf{X}$ , the proposed depth network predicts the reference image depth as  $\mathbf{D}^r = G(\mathbf{T}, \mathbf{X}; \theta_D)$ . However, if we switch the order of the reference and source image, *i.e.*,  $\mathbf{X}' = (\mathbf{I}^s, \mathbf{I}^r)$ , we can predict the source image depth map as  $\mathbf{D}^s = G(\mathbf{T}^{-1}, \mathbf{X}'; \theta_D)$ .

Additionally, we know there is an inherent dependency between camera pose, optical flow, and depth. As mentioned before (Sec. I), most previous methods overlook exploiting this dependency to refine pose and depth. On the contrary, we propose to exploit the relation between them over iteration to improve our camera pose and depth. The key idea is to utilize the current pose and depth estimates to update flow confidence. Given  $\mathbf{T}$  and  $\mathbf{D} = (\mathbf{D}^r, \mathbf{D}^s)$ , we compute a new induced flow using Eq.(2), which provides additional evidence about the dense correspondences. The weights are penalized according to the distance between the predicted flow  $\mathbf{Y}$  and the new induced flow<sup>1</sup>. We adopt the RBF kernel with standard deviation  $\sigma$  to penalize large distances. Taking forward flow as an example, the penalization factor can be computed as

$$\Delta w_{r \rightarrow s, ij}^{\text{iter}} := \exp\left(-\frac{\|\mathbf{p}_{\text{flow}, ij}^s - \mathbf{p}_{\text{ind\_flow}, ij}^s(\mathbf{T}, \mathbf{D}^r)\|^2}{2\sigma^2}\right) \quad (7)$$

where the weights are updated as  $w_{r \rightarrow s, ij} := \Delta w_{r \rightarrow s, ij} \cdot w_{r \rightarrow s, ij}$ . We also update the predicted flow through a simple mixup as

$\mathbf{Y}_{r \rightarrow s, ij} := \frac{1}{2}\mathbf{Y}_{r \rightarrow s, ij} + \frac{1}{2}(\mathbf{p}_{\text{ind\_flow}, ij}^s(\mathbf{T}, \mathbf{D}^r) - \mathbf{p}_{ij}^r)$ . We re-run the WBA with the updated flow and weight to obtain a better pose and further improve the depth.

### D. Loss Functions

The uncertainty-aware flow estimation network is trained using the negative log-likelihood loss, while the depth estimation network is trained using standard L1 loss.

$$\mathcal{L}_{\text{flow}} = -\log p(\mathbf{Y}^{gt} | \Phi(\mathbf{X}; \theta_F)); \quad \mathcal{L}_{\text{depth}} = \sum_{ij} |\mathbf{D}_{ij}^{gt} - \mathbf{D}_{ij}|. \quad (8)$$

We trained the two networks separately using ground truth pose  $\mathbf{T}^{gt}$  and depth  $\mathbf{D}^{gt}$  at train time. The induced ground truth flow  $\mathbf{Y}^{gt}$  is estimated from  $\mathbf{T}^{gt}$  and  $\mathbf{D}^{gt}$  on-the-fly.

## IV. EXPERIMENTS, RESULTS, AND ABLATIONS

### A. Datasets and Evaluation Metrics

Following the previous methods [21] [25] [44], we evaluated our proposed pipeline on four standard two-view SfM benchmark datasets, namely **YFCC100M** [45], **ScanNet** [46], **DeMoN** [20], and **KITTI VO** [47].

(i) **YFCC100M**. It is a large-scale dataset containing 100 million media objects with metadata collected from the internet. We follow the standard set-up of [48] and evaluate camera pose on four outdoor test scenes. Each scene contains 1000 image pairs with ground truth pose generated by [49].

(ii) **ScanNet**. ScanNet is a large-scale RGB-D video dataset with annotated camera poses and ground truth depth maps from 1613 scans of indoor scenes. We follow the standard set-up of [39] and evaluate our camera pose estimation method on the 1500 testing pairs.

(iii) **DeMoN**. DeMoN is a two-view SfM benchmark containing images from various scenes. The training set consists of images from three sources, *i.e.*, RGBD [50], SUN3D [51], and Scene11 [20]. RGBD is a SLAM dataset with high-quality RGB-D data with 16786 training and 160 testing pairs. Sun3D is a diverse indoor dataset with noisy camera pose and depth with 79577 training and 160 testing pairs. Scenes11 is a synthetic dataset of the simulated virtual scenes with random-positioned objects, containing 71820 training and 256 testing pairs. We use the same train split from [42] for training depth.

(iv) **KITTI VO**. KITTI visual odometry dataset contains image sequences with ground truth trajectories collected from real-world driving scenes. We evaluate our method on seq.09 with 1591 images and seq.10 with 1201 images.

**Evaluation Metrics**. For YFCC100M and ScanNet, we evaluate the pose accuracy using the cumulative pose error curve (AUC) and mean Average Precision (mAP) of the pose error at different thresholds ( $5^\circ$ ,  $10^\circ$ ,  $20^\circ$ ) [44]. The pose error is computed as the maximum angular distance for rotation and translation vectors between ground truth and prediction. For DeMoN, we use the standard metrics from [21]. Concretely, we assess the angular distance between rotation and translation vectors while the depth performance using scale-invariant error, L1 relative error, and L1 inverse error [20]. For KITTI VO, we follow [52] and report the absolute trajectory RMSE error of the entire trajectory with scale alignment.

<sup>1</sup>Alternatively, iterative reweighted least squares can also be used.

TABLE I: Camera pose estimation on YFCC100M [45]. The statistics show that our method achieves better results compared to the SOTA sparse (top) and dense (bottom) methods.

Method	AUC $\uparrow$			mAP $\uparrow$		
	@5°	@10°	@20°	@5°	@10°	@20°
SIFT [11] + ratio test	24.09	40.71	58.14	45.12	55.81	67.20
SIFT [11] + OANet [48]	29.15	48.12	65.08	55.06	64.97	74.83
SIFT [11] + SuperGlue [39]	30.49	51.29	69.72	59.25	70.38	80.44
SuperPoint [40] (SP)	-	-	-	30.50	50.83	67.85
SP [40] + OANet [48]	26.82	45.04	62.17	50.94	61.41	71.77
SP [40] + SuperGlue [39]	38.72	59.13	75.81	67.75	77.41	85.70
LoFTR-DT [17]	<b>42.21</b>	<b>62.07</b>	<b>77.22</b>	<b>72.27</b>	<b>79.99</b>	<b>86.95</b>
D2D [36]	-	-	-	55.58	66.79	-
RANSAC-Flow [16]	-	-	-	64.88	73.31	81.56
PDC-Net (D) [15]	32.21	52.61	70.13	60.52	70.91	80.03
PDC-Net (H) [15]	34.88	55.17	71.72	63.90	73.00	81.22
PDC-Net+ (D) [44]	34.76	55.37	72.55	63.93	73.81	82.74
PDC-Net+ (H) [44]	<b>37.51</b>	<b>58.08</b>	<b>74.50</b>	<b>67.35</b>	<b>76.56</b>	<b>84.56</b>
Ours	<b>45.48</b>	<b>63.90</b>	<b>78.05</b>	<b>73.85</b>	<b>80.80</b>	<b>87.16</b>

TABLE II: Camera pose estimation on ScanNet [46]. The statistics show that our method achieves better results compared to the SOTA sparse (top) and dense (bottom) methods. The symbol † represents networks that do not use ScanNet trainset.

Method	AUC $\uparrow$			mAP $\uparrow$		
	@5°	@10°	@20°	@5°	@10°	@20°
ORB [2] + GMS [53]	5.21	13.65	25.36	-	-	-
D2-Net [54] + NN	5.25	14.53	27.96	-	-	-
ContextDesc [55] + ratio test	6.64	15.01	25.75	-	-	-
SP [40] + OANet [48]	11.76	26.90	43.85	-	-	-
SP [40] + SuperGlue [39]	16.16	33.81	51.84	-	-	-
LoFTR-OT [17] †	16.88	33.62	50.62	-	-	-
LoFTR-OT [17]	21.51	40.39	<b>57.96</b>	-	-	-
LoFTR-DT [17]	<b>22.06</b>	<b>40.80</b>	<b>57.62</b>	-	-	-
PDC-Net † (D) [15]	17.70	35.02	51.75	39.93	50.17	60.87
PDC-Net † (H) [15]	18.70	36.97	53.98	42.87	53.07	63.25
PDC-Net+ † (D) [44]	19.02	36.90	54.25	42.93	53.13	63.95
PDC-Net+ † (H) [44]	<b>20.25</b>	<b>39.37</b>	<b>57.13</b>	<b>45.66</b>	<b>56.67</b>	<b>67.07</b>
Ours †	<b>24.07</b>	<b>43.58</b>	<b>60.35</b>	<b>51.13</b>	<b>61.00</b>	<b>70.23</b>

## B. Implementation Details

We implemented our approach in Python 3.9 with PyTorch 1.11. The confidence threshold for weighted bundle adjustment is set to  $\gamma = 0.1$  [44]. For RANSAC pose initialization, we adopt the OpenCV implementation with threshold 1.0 and confidence 0.99. For camera pose estimation on YFCC100M-ScanNet and KITTI VO, we use the provided models of PDC-Net+ (H) [44] and PDC-Net (D) [15] pre-trained on Megadepth, respectively. For camera pose estimation on DeMoN dataset, we use the PDC-Net (D) [15] pre-trained on Megadepth and fine-tune on DeMoN train set for 50 epochs using Adam optimizer with  $lr = 5e^{-5}$ . On DeMoN, for training the depth estimation network, we use the DeMoN train set and train our model for 50 epochs using Adam optimizer with  $lr = 8e^{-4}$ , which takes 48 hours on 4 NVIDIA TITAN RTX GPUs. To maintain the stability of fine-tuned flow confidence for pose estimation, we use the additional  $4 \times 4$  local grid filtering with confidence quantile in each local grid.

## C. Results

(i) **Results on YFCC100M and ScanNet.** Results for indoor and outdoor scenes are shown in Table I and Table II, respectively. Our method uses the pre-trained optical flow backbone model, which is denoted as the ‘PDC-Net+ (H)’

[44] in the tables. Table I-II statistical results indicate that our method outperforms the classic sparse keypoint-based two-view SfM pipeline by a significant margin, which uses SIFT descriptors [11] on both YFCC100M and ScanNet benchmark dataset. Our method further supersedes the popular state-of-the-art learning-based sparse methods, *i.e.*, SuperPoint [40] + SuperGlue [39] and LoFTR [17], which have been commonly used by many recent 3D reconstruction and localization systems [4] [56]. Compared to learning-based dense methods, our method outperforms the previous state-of-the-art ‘PDC-Net+(H)’ for YFCC100M and ‘LoFTR-DT’ for ScanNet. Noticeably, using the pre-trained optical flow backbone with our weighted bundle adjustment pose estimation shows consistent improvement on all metrics.

(ii) **Results on DeMoN.** Unlike YFCC100M and ScanNet dataset, DeMoN provides both depth and camera pose ground truth for evaluation. Table III shows the quantitative comparison results of our method with other competing approaches. Results for both pose and depth in Table III are shown side-by-side for better understanding. It can be inferred from the table that our method outperforms all other methods in camera pose performance accuracy. Note that we follow [25] and rely on pre-trained models as outlined in the implementation, contrary to the DeepSfM [21] and FMD [28] that train their pose network directly on the DeMoN pose set.

Unlike [21] which trains depth networks using their network’s predicted pose to better fit the ground-truth depth, we train the depth estimation network using ground truth pose alone. This helps assess the robustness of the depth estimation network against the predicted pose at test time. Contrary to [21], our depth-network training methodology is close to the conventional way of training a deep neural network.

(iii) **Results on KITTI VO.** Further, we evaluate our pose accuracy on KITTI VO, as shown in Table IV. We align all trajectories to the ground truth before computing the RMSE. Our method achieves better results than other learning-based methods without fine-tuning on the KITTI VO dataset.

## D. Ablation Study

(i) **Optical flow modeling and its effect on camera pose estimation.** The proposed camera pose estimation method uses weighted bundle adjustment to aptly encapsulate per-pixel correspondence uncertainty, the bi-directional flow consistency, and the robust outlier rejection. Previous methods, for example, PDC-Net [44] directly uses RANSAC [16] with the forward optical flow and fixed confidence threshold to estimate camera pose, and hence overlooks to exploit the uncertainty information and bi-directional flow consistency mindfully. As shown in Table V, using the introduced weighted bundle adjustment that integrates both optical flow prediction confidence and outlier filtering achieves significant improvements for both single-direction and bi-direction cases. Further, the use of bi-directional optical flow estimates outperforms single-directional flow by a clear margin, indicating the effectiveness of bi-directional optical flow consistency.

(ii) **Use of iterative refinement.** Here we show that we can iteratively improve camera pose and depth by exploiting the

TABLE III: Depth and pose estimation results on DeMoN [20]. We can observe that our method shows better results for pose estimation on all three datasets. In contrast to [21] that relies on the additional pose and depth initialization for depth supervision, our depth network only uses ground-truth pose to train and achieves remarkable results.

Method	RGBD					Scenes11					Sun3D				
	Depth			Pose		Depth			Pose		Depth			Pose	
	L1-inv	Sc-inv	L1-rel	Rot	Trans	L1-inv	Sc-inv	L1-rel	Rot	Trans	L1-inv	Sc-inv	L1-rel	Rot	Trans
Base-SIFT	0.050	0.577	0.703	12.010	56.021	0.051	0.900	1.027	6.179	56.650	0.029	0.290	0.286	7.702	41.825
Base-Matlab	-	-	-	12.813	49.612	-	-	-	0.917	14.639	-	-	-	5.920	32.298
DeMoN [20]	0.028	0.130	0.212	2.641	20.585	0.019	0.315	0.248	0.809	8.918	0.019	0.114	0.172	1.801	18.811
LS-Net [26]	0.019	0.090	0.301	1.010	22.100	0.010	0.410	0.210	4.653	8.210	0.015	0.189	0.650	1.521	14.347
BA-Net [27]	0.008	0.087	<b>0.050</b>	2.459	14.900	0.080	0.210	0.130	3.499	10.370	0.015	0.110	0.060	1.729	13.260
FMD [28]	0.010	0.158	0.107	1.570	11.163	0.015	0.268	0.179	0.615	5.331	<b>0.009</b>	0.105	0.076	1.494	12.049
DeepSfM [21]	0.011	<b>0.071</b>	0.126	1.862	14.570	0.007	<b>0.112</b>	0.064	0.403	5.828	0.013	0.093	0.072	1.704	13.107
Deep2View [25]	-	-	-	-	-	<b>0.005</b>	<b>0.097</b>	0.058	0.276	2.041	0.010	<b>0.081</b>	<b>0.057</b>	1.391	10.757
Ours	<b>0.007</b>	0.086	0.052	<b>0.951</b>	<b>7.640</b>	0.007	0.115	<b>0.053</b>	<b>0.107</b>	<b>0.772</b>	0.011	0.092	0.062	<b>1.165</b>	<b>9.731</b>

TABLE IV: RMSE (m) results comparison with competing methods on KITTI VO Seq.09 and Seq.10 [47].

Method	SfMLearner [22]	CC [12]	DMVO [57]	LVMVO [52]	Ours
Seq.09	24.31	29.00	27.08	11.30	<b>6.65</b>
Seq.10	20.87	13.77	24.44	11.80	<b>5.58</b>

TABLE V: Effect of the proposed pose estimation method on ScanNet [46]. ‘F’, ‘F-B’ symbolizes forward flow and forward-backward flow, respectively.

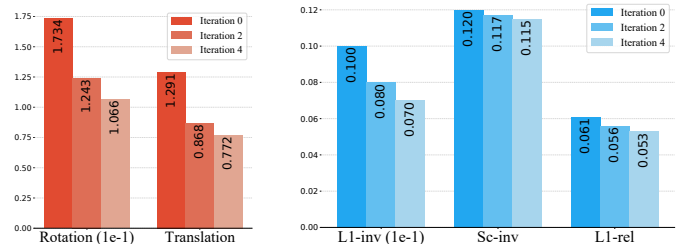
Method	Flow	Mask	AUC $\uparrow$		
			@5 $^\circ$	@10 $^\circ$	@20 $^\circ$
RANSAC	F	Conf.	19.47	38.62	56.80
Ours	F	RANSAC	19.06	37.01	54.55
Ours	F	Conf.	20.11	37.67	55.32
Ours	F	Conf.+RANSAC	<b>21.82</b>	<b>40.14</b>	<b>57.59</b>
Ours	F-B	-	20.85	39.46	56.88
Ours	F-B	RANSAC	21.31	40.00	56.88
Ours	F-B	Conf.	22.62	40.93	58.02
Ours	F-B	Conf.+RANSAC	<b>24.07</b>	<b>43.58</b>	<b>60.35</b>

dependency loop of optical-flow correspondence, camera pose, and depth. We can induce the flow from the predicted camera pose and depth, which provides additional evidence of the flow uncertainty. In Figure 3, we report the camera pose and depth error results for iterative refinement on Scenes11. We can observe that camera pose and depth error decrease over iterations, indicating that the joint information obtained from pose and depth estimation can provide helpful cues for updating the flow and reweighting the flow confidence. After four iterations, the rotation error drops 38%, and the translation drops 40%. The pose update also converges along with the depth error as the reweighting process does not obtain new information from induced flow consistency.

(iii) **Effect of fine-tuning on pose estimation.** We show the influence of fine-tuning on our pose estimation method in Table VI. For PDC-Net [15], we run RANSAC on predicted

TABLE VI: Effect of fine-tuning on DeMoN [20]. The statistics show our approach’s effectiveness for both w/ and w/o fine-tuning.

Method	RGBD		Scenes11		Sun3D	
	Rot	Trans	Rot	Trans	Rot	Trans
PDC-Net (pre-trained)	1.311	10.424	0.721	5.745	1.382	13.331
Ours (pre-trained)	0.987	9.754	0.429	2.684	1.256	11.079
PDC-Net (fine-tuned)	1.108	8.803	0.321	2.210	1.292	12.197
Ours (fine-tuned)	<b>0.951</b>	<b>7.640</b>	<b>0.107</b>	<b>0.772</b>	<b>1.165</b>	<b>9.731</b>



(a) Pose Error (b) Depth Error  
Fig. 3: Iterative refinement on Scenes11 [20]. It shows that pose and depth errors can be largely reduced via iterative refinement.

TABLE VII: Runtime evaluation.

Flow Est.	RANSAC	WBA	Depth Est.	Total
0.13s	0.14s	0.21s	0.17s	0.65s

optical flow to recover camera poses. Our method shows clear improvement on both pre-trained and fine-tuned models.

(iv) **Runtime Evaluation.** Table VII provides our method’s runtime analysis on a single Nvidia GTX 1080 Ti GPU. Due to our modular design on the two-view SfM problem, we also have the flexibility to choose between accuracy and efficiency depending on the requirement.

## V. CONCLUSION

From our extensive experimental analysis and rigorous ablation study, we conclude that our approach, which explicitly models the correctness of per-pixel correspondence matching in the neural network design itself with classical take for pose estimation, outperforms recent methods’ results. Moreover, our approach for performing online refinement via weighted bundle adjustment help further improve camera pose and depth estimation accuracy. In this regard, it is observed that the proposed iterative refinement is indeed vital to exploit the dependency of optical flow, depth, and camera pose in SfM, which is generally overlooked in the previous deep learning-based dense SfM approaches. That said, our approach assumes known intrinsic camera parameters, and we hope to extend the pipeline to fully uncalibrated SfM in our follow-up work.

## ACKNOWLEDGMENTS

This work was supported by an ETH RobotX research grant funded through the ETH Zürich Foundation.



## REFERENCES

- [1] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *ICRA*. IEEE, 2014, pp. 15–22.
- [2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *TOR*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [3] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, “Building rome in a day,” *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.
- [4] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *CVPR*, 2016, pp. 4104–4113.
- [5] A. Mossel and M. Kroeter, “Streaming and exploration of dynamically changing dense 3d reconstructions in immersive virtual reality,” in *ISMAR*. IEEE, 2016, pp. 43–48.
- [6] Z. Teed and J. Deng, “Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras,” *NeurIPS*, vol. 34, pp. 16558–16569, 2021.
- [7] D. Menini, S. Kumar, M. R. Oswald, E. Sandström, C. Sminchisescu, and L. Van Gool, “A real-time online learning framework for joint 3d reconstruction and semantic segmentation of indoor scenes,” *IEEE RAL*, vol. 7, no. 2, pp. 1332–1339, 2021.
- [8] S. Kumar, Y. Dai, and H. Li, “Multi-body non-rigid structure-from-motion,” in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 148–156.
- [9] S. Kumar and L. Van Gool, “Organic priors in non-rigid structure from motion,” in *ECCV, Part II*. Springer, 2022, pp. 71–88.
- [10] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [11] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, “Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation,” in *CVPR*, 2019, pp. 12240–12249.
- [13] D. Nistér, “An efficient solution to the five-point relative pose problem,” *T-PAMI*, vol. 26, no. 6, pp. 756–770, 2004.
- [14] Z. Teed and J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow,” in *ECCV*. Springer, 2020, pp. 402–419.
- [15] P. Truong, M. Danelljan, L. Van Gool, and R. Timofte, “Learning accurate dense correspondences and when to trust them,” in *CVPR*, 2021, pp. 5714–5724.
- [16] X. Shen, F. Darmon, A. A. Efros, and M. Aubry, “Ransac-flow: generic two-stage image alignment,” in *ECCV*. Springer, 2020, pp. 618–637.
- [17] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “Loftr: Detector-free local feature matching with transformers,” in *CVPR*, 2021, pp. 8922–8931.
- [18] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [19] H. C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.
- [20] B. Ummerhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, “Demon: Depth and motion network for learning monocular stereo,” in *CVPR*, 2017, pp. 5038–5047.
- [21] X. Wei, Y. Zhang, Z. Li, Y. Fu, and X. Xue, “DeepSfm: Structure from motion via deep bundle adjustment,” in *ECCV*. Springer, 2020, pp. 230–247.
- [22] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in *CVPR*, 2017, pp. 1851–1858.
- [23] Z. Yin and J. Shi, “Geonet: Unsupervised learning of dense depth, optical flow and camera pose,” in *CVPR*, 2018, pp. 1983–1992.
- [24] R. Mahjourian, M. Wicke, and A. Angelova, “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints,” in *CVPR*, 2018, pp. 5667–5675.
- [25] J. Wang, Y. Zhong, Y. Dai, S. Birchfield, K. Zhang, N. Smolyanskiy, and H. Li, “Deep two-view structure-from-motion revisited,” in *CVPR*, 2021, pp. 8953–8962.
- [26] R. Clark, M. Bloesch, J. Czarnowski, S. Leutenegger, and A. J. Davison, “Learning to solve nonlinear least squares for monocular stereo,” in *ECCV*. Springer, 2018, pp. 291–306.
- [27] C. Tang and P. Tan, “BA-net: Dense bundle adjustment networks,” in *ICLR*, 2019.
- [28] K. Wang and S. Shen, “Flow-motion and depth network for monocular stereo and beyond,” *RA-L*, vol. 5, no. 2, pp. 3307–3314, 2020.
- [29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *IROS*. IEEE, 2012, pp. 573–580.
- [30] J.-C. Bazin, Y. Seo, R. Hartley, and M. Pollefeys, “Globally optimal inlier set maximization with unknown rotation and focal length,” in *ECCV*. Springer, 2014, pp. 803–817.
- [31] J. Yang, H. Li, and Y. Jia, “Optimal essential matrix estimation via inlier-set maximization,” in *ECCV*. Springer, 2014, pp. 111–126.
- [32] Z. Teed and J. Deng, “Deepv2d: Video to depth with differentiable structure from motion,” in *ICLR*, 2020.
- [33] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” in *ECCV*. Springer, 2006, pp. 404–417.
- [34] S. Kumar, Y. Dai, and H. Li, “Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene,” *TPAMI*, vol. 43, no. 5, pp. 1705–1717, 2019.
- [35] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, “Unsupervised cnn for single view depth estimation: Geometry to the rescue,” in *ECCV*. Springer, 2016, pp. 740–756.
- [36] O. Wiles, S. Ehrhardt, and A. Zisserman, “D2d: Learning to find good correspondences for image matching and manipulation,” 2020.
- [37] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, “Learning feature descriptors using camera pose supervision,” in *ECCV*. Springer, 2020, pp. 757–774.
- [38] Z. Huang, X. Pan, W. Pan, W. Bian, Y. Xu, K. C. Cheung, G. Zhang, and H. Li, “Neuralmarker: A framework for learning general marker correspondence,” *TOG*, vol. 41, no. 6, pp. 1–10, 2022.
- [39] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superglue: Learning feature matching with graph neural networks,” in *CVPR*, 2020, pp. 4938–4947.
- [40] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *CVPR workshops*, 2018, pp. 224–236.
- [41] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment—a modern synthesis,” in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.
- [42] Z. Yang, Z. Ren, Q. Shan, and Q. Huang, “Mvs2d: Efficient multi-view stereo via attention-driven 2d convolutions,” in *CVPR*, 2022, pp. 8574–8584.
- [43] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *ICML*. PMLR, 2019, pp. 6105–6114.
- [44] P. Truong, M. Danelljan, R. Timofte, and L. Van Gool, “Pdc-net+: Enhanced probabilistic dense correspondence network,” *arXiv preprint arXiv:2109.13912*, 2021.
- [45] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “Yfcc100m: The new data in multimedia research,” *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.
- [46] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “ScanNet: Richly-annotated 3d reconstructions of indoor scenes,” in *CVPR*, 2017, pp. 5828–5839.
- [47] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *IJRR*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [48] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, “Learning two-view correspondences and geometry using order-aware network,” in *ICCV*, 2019, pp. 5845–5854.
- [49] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm, “Reconstructing the world\* in six days\*(as captured by the yahoo 100 million image dataset),” in *CVPR*, 2015, pp. 3287–3295.
- [50] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *IROS*. IEEE, 2012, pp. 573–580.
- [51] J. Xiao, A. Owens, and A. Torralba, “Sun3d: A database of big spaces reconstructed using sfm and object labels,” in *ICCV*, 2013, pp. 1625–1632.
- [52] Y. Zou, P. Ji, Q.-H. Tran, J.-B. Huang, and M. Chandraker, “Learning monocular visual odometry via self-supervised long-term modeling,” in *ECCV*. Springer, 2020, pp. 710–727.
- [53] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, “Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence,” in *CVPR*, 2017, pp. 4181–4190.
- [54] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-net: A trainable cnn for joint description and detection of local features,” in *CVPR*, 2019, pp. 8092–8101.
- [55] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, “Contextdesc: Local descriptor augmentation with cross-modality context,” in *CVPR*, 2019, pp. 2527–2536.



- [56] Z. Cui and P. Tan, "Global structure-from-motion by similarity averaging," in *ICCV*, 2015, pp. 864–872.
- [57] T. Shen, Z. Luo, L. Zhou, H. Deng, R. Zhang, T. Fang, and L. Quan, "Beyond photometric loss for self-supervised ego-motion estimation," in *ICRA*. IEEE, 2019, pp. 6359–6365.