STRUCTURE AND MOTION FROM DEPTH AND CORRESPONDENCE MODELS

By

Shengjie Zhu

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2025

ABSTRACT

Recovering structure and motion from videos is a well-studied comprehensive 3D vision task that involves (1) image calibration, (2) two-view pose initialization, and (3) multi-view Structure-from-Motion (SfM). Prior arts are optimization-based methods built over sparse image correspondence inputs. This thesis develops systematic approaches to enhance classic solutions with deep learning models. We introduce EdgeDepth and PMatch for dense monocular depthmaps and dense binocular correspondence map estimations. Since classic approaches typically rely on sparse and accurate inputs, they are less suitable for the dense yet high-variance predictions from dense depth and correspondence models. As a solution, we propose to optimize through the robust inlier-counting-based scoring function, which is widely applied in RANdom SAmpling Consensus (RANSAC). Our system is structured as follows: (1) For image calibration, we introduce WildCamera. The system utilizes a RANSAC algorithm applied to a dense incidence field regressed by a deep model. It calibrates in-the-wild monocular images without checkerboard. (2) In two-view pose estimation, we introduce LightedDepth. It estimates the optimal pose by aligning the depth map with the correspondence map, maximizing the projective inliers. (3) The strategy is extended to a Hough Transform in RSfM for multi-view SfM over a local 3 to 9 frame system. (4) We generalize the RSfM discrete inlier counting scoring function to a smoothed scoring function via marginalizing thresholds for general SfM task. To this end, we formulate a comprehensive system that recovers structure and motion from two-view / local multi-view / large-scale multi-view images with dense monocular depthmap and binocular correspondence maps. Compared to prior arts, our methods show comprehensive improvement on two-view, small-scale, and large-scale multi-view systems.

This thesis is dedicated to my wife Lisheng, whose unwavering support and encouragement have been my greatest source of strength throughout this journey.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Xiaoming Liu, for his unwavering support, guidance, and mentorship throughout my Ph.D. journey. His high expectations and relentless pursuit of excellence have pushed me beyond my limits, enabling me to grow as a researcher and thinker. His willingness to engage in deep technical discussions, provide constructive feedback, and challenge me to think critically has been invaluable in shaping my research skills. Through his mentorship, I have learned the importance of perseverance, curiosity, and precision in scientific inquiry. Beyond academics, his encouragement and belief in my potential have been a constant source of motivation. I am truly honored to have had the opportunity to learn from him, and his mentorship will continue to inspire me in my future endeavors.

I would also like to express my great pleasure to have Prof. Anil Jain, Prof. Daniel Morris, and Prof. Vishnu Boddeti in my Ph.D. guidance committee for their valuable guidance and feedback.

I would like to thank Dr. Ahmed Abdelkader, Dr. Vincent Chu, and Mark Matthews for their mentorship and support during my internship at Google. Your support was instrumental in shaping the final chapter of this thesis. I would also like to thank Dr. Ning Zhou, Dr. Haotian Xu, Dr. Jingyi Zhang, and Dr. Rui Hou for their mentorship during my internship at Amazon. It has been a truly inspiring experience, broadening my perspective on research and innovation. The insights I gained from the team have been a lasting source of motivation, driving my later research. Further, I am deeply grateful to Prof. Fernando for providing me with the opportunity to present at CMU.

CVLab is a loving place. I would like to thank all my labmates, Dr. Xi Yin, Dr. Amin Jourabloo, Dr. Garrick Brazil, Dr. Luan Tran, Dr. Feng Liu, Dr. Yaojie Liu, Dr. Andrew Hou, Dr. Abhinav Kumar, Dr. Vishal Asnani, Xiao Guo, Minchul Kim, Joel Stehouwer, Bangjie Yin, Hieu Nguyen, Masa Hu, Ziyuan Zhang, Girish Chandar Ganesan, Yiyang Su, Jie Zhu, and Zhiyuan Ren, for making my Ph.D. journey both productive and enjoyable.

Finally, I would like to extend my heartfelt gratitude to my wife and my parents for their unconditional support. No words can fully capture the depth of my love for you.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Estimating structure and motion from 2D images set is a fundamental task with diverse applications in 3D reconstruction [31], robotics [132], and autonomous driving [310]. This task extracts 3D point clouds, camera extrinsics, and camera intrinsics from RGB images, requiring a comprehensive vision system that includes camera calibration, two-view pose estimation, and multi-view Structure-from-Motion (SfM).

Classic Structure-from-Motion (SfM) methods [209, 287, 85] rely on sparse image correspondence inputs extracted using feature detectors such as SIFT [150], SURF [15], and learned descriptors like SuperPoint [63, 204]. These methods construct a sparse yet highly accurate 3D point cloud by triangulating matched keypoints across multiple views. Given a well-initialized system, a robust Bundle Adjustment (BA) algorithm jointly optimizes 3D point positions, camera intrinsics, and camera poses by minimizing reprojection and photometric errors. However, classic SfM approaches typically assume that the input image collection exhibits well-textured regions, sufficient parallax between views, and a high degree of visual overlap—conditions that may not always hold in real-world scenarios.

Recent advancements in deep learning have enabled the development of monocular depth estimators [19] that generate dense depth maps from single RGB images without requiring camera motion. The rise of transformer-based foundation models [259] has further accelerated research efforts toward creating large-scale, highly generalizable monocular depth estimation models. These models [295] are trained using a combination of large-scale labeled datasets and unlabeled image collections, enhancing their robustness and adaptability. Notably, monocular depth models output depth maps or point clouds in metric space, in contrast to traditional SfM systems, which produce up-to-scale point clouds. Despite its growing capabilities, monocular depth estimation has limitations. A major drawback is that the generated point clouds are significantly noisier compared to those produced by SfM. Additionally, accurately quantifying the noise level in these depth predictions remains an open research question. Since classic SfM algorithms rely on sparse and highly accurate

points, the high variance in monocular depth maps makes them less suitable for direct integration into traditional SfM pipelines.

A similar trend is observed in image correspondence estimation models [331]. Traditionally, image correspondence relied on handcrafted feature descriptors such as SIFT [151] and ORB [175]. In contrast, learning-based methods utilize labeled data to automatically learn image matching features during training. These models utilize more powerful computational resources, such as GPUs. As a result, learning-based approaches have achieved significantly higher accuracy compared to their handcrafted counterparts. Recently, several studies have extended sparse image correspondence estimation to dense correspondence estimation [69]. The transition to a dense output format allows learning-based models to incorporate additional global priors. Experimental results in two-view pose estimation have shown that dense correspondence can improve pose estimation accuracy. However, similar to monocular depth models, dense correspondence estimators face challenges in integrating with classic SfM methods, as these systems are designed to operate on sparse point clouds.

Camera pose estimation has become increasingly important with the growing number of applications that rely on precise spatial localization. For instance, autonomous vehicles, drones, and other robotic systems depend on accurate pose estimation for navigation. Additionally, emerging 3D image generation methods [176], which synthesize coherent 3D models from multiview images, require well-registered input images. Neural rendering techniques [170], with significant potential in AR/VR applications, also assume multi-view images with known camera poses.

Pioneering work in pose estimation has explored the integration of deep learning with camera pose estimation. One line of research focuses on absolute camera pose regression, where a deep neural network takes a single image or an image pair as input and directly regresses the absolute camera pose in world coordinates [24] or the relative pose between the two images [216]. Another approach is scene coordinate regression, where the model predicts a 3D point cloud either in a global world coordinate system [21] or relative to the input images [273]. However, there is still insufficient evidence that these learning-based methods consistently outperform traditional

geometric approaches. In this thesis, we propose a novel framework that effectively combines deep learning with camera pose estimation, leveraging the strengths of both paradigms. Our approach utilizes deep networks for dense, pixel-wise predictions guided by spatial geometric priors, *i.e.*, the dense depthmaps and correspondence maps. Then, we employ a post-optimization scheme to refine the low-degree-of-freedom (DoF) camera poses based on the dense yet noisy predictions.

In this dissertation, we present a comprehensive system for estimating multi-view camera intrinsics and extrinsics by leveraging network outputs, specifically dense image correspondence maps and depth maps. Our approach begins with the dense depth estimator EdgeDepth [327] and the dense image correspondence estimator PMatch [331]. We then introduce a two-view pose initialization method, LightedDepth [330], followed by RSfM [332], a multi-view pose estimation algorithm designed to refine the two-view initialized results within a small multi-view system. Finally, we present MfS, an extension of RSfM that enhances performance across both small-scale and large-scale multi-view pose estimation scenarios. We start with the inputs to our system, *i.e.*, the Monocular Depth Estimator and Binocular Correspondence Estimator.

In Chapter 2, we present a monocular depth estimarot EdgeDepth [327]. EdgeDepth explores the mutual benefits between self-supervised monocular depth estimation and semantic segmentation, two fundamental tasks in computer vision. Unlike previous methods that implicitly model their relationship, we introduce an explicit border consistency constraint, ensuring alignment between segmentation and depth edges. We leverage a novel morphing algorithm to iteratively refine depth predictions, making them more consistent with segmentation boundaries. Additionally, we identify and mitigate bleeding artifacts commonly found in stereo-based self-supervised depth estimation using a stereo occlusion masking technique, further enhancing depth quality near object edges. Our approach achieves state-of-the-art performance on self-supervised monocular depth estimation, for the first time matching supervised methods in absolute relative error on the KITTI dataset.

In Chapter 3, we introduce PMatch [331], a novel Paired Masked Image Modeling (pMIM) framework designed for dense geometric matching. Traditional monocular pretraining tasks, such as image classification and masked image modeling (MIM), fail to optimize the cross-frame match-

ing module, limiting their effectiveness in geometric correspondence estimation. To overcome this, we reformulate MIM from reconstructing a single masked image to reconstructing a pair of masked images, enabling more effective pretraining of the transformer-based matching module. Additionally, we propose a cross-frame global matching module (CFGM) that enhances robustness in textureless regions by incorporating positional embeddings and a homography loss, which regularizes correspondences on planar surfaces. Through these innovations, PMatch achieves state-of-the-art performance in dense geometric matching, outperforming both sparse and dense methods on diverse benchmark datasets.

Given the input depth maps and correspondence maps, we outline our pose estimation system, beginning with a monocular intrinsic calibration method. This is followed by a two-view pose initialization approach. Finally, we introduce RSfM for small-scale multi-view pose estimation and MfS for large-scale multi-view pose estimation.

In Chapter 4, we introduce WildCamera [329], a 4 Degree-of-Freedom (DoF) camera calibration method tailored for in-the-wild images. Our approach is motivated by the intrinsic relationship between monocular depth maps and surface normal maps, where the optimal intrinsic parameters should align the depth map consistently with the normal map. However, traditional depth-normal-based calibration methods suffer from numerical instability due to their dependence on accurate depth gradients. To address this, we propose an alternative representation—the incidence field, a novel 3D monocular prior that models the incidence rays between observed 3D points and their corresponding 2D projections on the imaging plane. Unlike conventional depth and normal maps, the incidence field remains invariant to image cropping and resizing, enhancing its generalization to in-the-wild images. We develop a deep neural network to estimate the incidence field and introduce a non-learning RANSAC-based optimization algorithm to recover intrinsic parameters from the estimated field. Our method achieves state-of-the-art performance on synthetic and real-world datasets, offering a robust solution for monocular camera calibration and enabling diverse downstream applications, including image manipulation detection, uncalibrated two-view pose estimation, and improved 3D sensing.

In Chapter 5, we present LightedDepth [330], a novel two-view SfM algorithm centered around a two-view metric space pose initialization approach. Given two input images, we extract dense monocular depth maps and image correspondences. Our method proceeds in three key stages: (1) We estimate a normalized up-to-scale camera pose from the correspondences. (2) We determine the metric space translation scale using a majority-voting algorithm, which incorporates a robust, non-differentiable inlier-counting-based scoring function to enhance reliability. This strategy effectively accommodates depth map noise by leveraging its density. (3) Finally, we complete two-view SfM by estimating the two-view structure as video depth, formulated as a logged residual regression over the monocular depth input. Through this decomposition, LightedDepth achieves superior performance in video depth estimation, demonstrating robustness in scenarios with limited inference view angles while maintaining computational efficiency.

This thesis explores advancements in self-supervised depth estimation by integrating local Structure-from-Motion (SfM). Traditional self-supervised depth estimation relies on photometric loss across immediate neighboring frames, often neglecting geometric consistency. To bridge this gap, we propose a local SfM approach with a novel Bundle-RANSAC-Adjustment algorithm that optimizes camera poses and depth adjustments across multiple frames. Experimental results demonstrate that with only a few frames, our method significantly improves depth accuracy and consistency, outperforming state-of-the-art supervised models. In sparse-view pose estimation, our approach achieves certified global optimality and surpasses existing methods in both rotational and translational accuracy. Additionally, it enhances correspondence estimation, confirming its robustness and applicability. These results establish that self-supervision within limited frames not only benefits supervised models but also sets new standards in pose and depth estimation, advancing applications in AR/VR, autonomous driving, and 3D reconstruction.

In Chapter 6, we propose RSfM [332], which extends LightedDepth [330]'s majority voting from two-view SfM to a local multi-view SfM with 3 to 9 frames. To address the non-differentiable inlier-counts scoring function, we introduce a Hough Transform to convert it to a differentiable manifold space. However, this transformation assumes all frames are mutually visible, limiting its

scalability. Despite this, RSfM shows improved pose accuracy over classic SfM by utilizing 3D priors within dense monocular depth maps, whereas classic methods [209] rely on triangulation, which is less effective with limited camera views (3 to 5 frames). Experiments demonstrate that self-supervision with only 5 frames already enhances the performance of state-of-the-art supervised models across datasets like ScanNet and KITTI360, achieving improvements in pose accuracy, depth consistency, and correspondence estimation.

In Chapter 7, we present Motion-from-Structure (MfS). We generalize the inlier counting strategy adopted in RSfM [332] to large-scale SfM systems. This method leverages the dense structural information from monocular depth priors to directly estimate camera motion without the need for per-pixel depth adjustments or model fine-tuning. Central to MfS is a reformulated bundle adjustment framework that distinguishes inliers and outliers through a robust scoring function. Unlike traditional methods that rely on a single inlier threshold, MfS generalizes this by computing an Area-Under-Curve (AUC) over multiple thresholds, effectively modeling the residual distribution as a continuous cumulative distribution function (CDF). This approach not only mitigates sensitivity to hyper-parameters but also offers a smooth and differentiable optimization landscape. Experiments on diverse datasets, including the sparse-set ETH3D and the large-scale dense-set ScanNet, demonstrate MfS's ability to achieve state-of-the-art performance in multi-view pose estimation and camera re-localization. Notably, MfS consistently outperforms classical methods by robustly handling noisy depth maps, achieving high accuracy even in challenging scenarios with limited texture or motion parallax. Furthermore, the method's scalable and plug-and-play design allows it to integrate seamlessly with arbitrary monocular depth estimation models, promoting efficient large-scale SfM without compromising accuracy.

1.1 Contributions of the Thesis

This thesis presents significant advancements in the field of Structure-from-Motion (SfM) and camera pose estimation, addressing challenges related to dense depth and correspondence estimation, camera calibration, and multi-view pose estimation. The primary contributions are:

1. Chapter 2: The Edge of Depth: Explicit Constraints between Segmentation and Depth

- Introduced EdgeDepth, a novel self-supervised monocular depth estimation framework that explicitly enforces border consistency between depth and semantic segmentation maps. This approach improves depth accuracy near object boundaries by employing a morphing algorithm and stereo occlusion masking to mitigate common artifacts.
- Achieved state-of-the-art performance on self-supervised depth estimation benchmarks, matching supervised methods on the KITTI dataset in terms of absolute relative error.

2. Chapter 3: PMatch: Paired Masked Image Modeling for Dense Geometric Matching

    - Developed PMatch, a transformer-based framework utilizing Paired Masked Image Modeling (pMIM) for robust dense geometric matching. This method enhances correspondence estimation in textureless regions using a cross-frame global matching module and homography loss.
    - Demonstrated superior performance over existing sparse and dense matching methods across diverse benchmark datasets.

3. Chapter 4: Tame a Wild Camera: In-the-Wild Monocular Camera Calibration

    - Proposed WildCamera, a 4 DoF camera calibration technique leveraging the novel concept of an incidence field. This approach ensures robustness to image cropping and resizing, enhancing its generalization to in-the-wild datasets.
    - Designed a deep learning model to estimate incidence fields and integrated a RANSAC-based optimization method for reliable intrinsic parameter recovery.
    - Achieved state-of-the-art calibration performance on both synthetic and real-world datasets, enabling diverse downstream applications.

4. Chapter 5: LightedDepth: Video Depth Estimation in light of Limited Inference View Angles

    - Presented LightedDepth, a two-view SfM algorithm that accurately estimates metric space poses by integrating dense depth and correspondence inputs. This method intro-

duces a robust majority-voting mechanism to determine translation scales and refines depth predictions through residual regression.

- Demonstrated robustness and superior performance in challenging scenarios with limited inference angles while maintaining computational efficiency.

5. Chapter 6: RSfM: Revisit Self-supervised Depth Estimation with Local Structure-from-Motion

    - Introduced RSfM, extending the LightedDepth framework to local multi-view settings (3–9 frames). This method innovates by converting non-differentiable inlier counts into a differentiable manifold space using the Hough Transform, enhancing pose accuracy in scenarios with limited mutual visibility.
    - Verified its effectiveness through experiments, showing improvements in pose accuracy, depth consistency, and correspondence estimation across benchmark datasets like ScanNet and KITTI360.

6. Chapter 7: Motion-from-Structure: Leveraging Monocular Depth Priors for Multi-View Tasks

    - Developed Motion-from-Structure (MfS), which generalizes the inlier counting strategy to large-scale SfM. MfS introduces a robust scoring function based on an Area-Under-Curve (AUC) framework, improving optimization smoothness and reducing sensitivity to hyper-parameters.
    - Demonstrated state-of-the-art performance in large-scale multi-view pose estimation and camera re-localization, particularly excelling in challenging scenarios involving noisy depth maps and limited texture or motion parallax.

Together, these contributions enhance structure and motion estimation from RGB image collections by bridging dense learning-based approaches with traditional geometric methods, leading to more accurate and scalable solutions.

CHAPTER 2

THE EDGE OF DEPTH: EXPLICIT CONSTRAINTS BETWEEN SEGMENTATION AND DEPTH

In this work we study the mutual benefits of two common computer vision tasks, self-supervised depth estimation and semantic segmentation from images. For example, to help unsupervised monocular depth estimation, constraints from semantic segmentation has been explored implicitly such as sharing and transforming features. In contrast, we propose to explicitly measure the border consistency between segmentation and depth and minimize it in a greedy manner by iteratively supervising the network towards a locally optimal solution. Partially this is motivated by our observation that semantic segmentation even trained with limited ground truth (200 images of KITTI) can offer more accurate border than that of any (monocular or stereo) image-based depth estimation. Through extensive experiments, our proposed approach advances the state of the art on unsupervised monocular depth estimation in the KITTI.

2.1 Introduction

Estimating depth is a fundamental problem in computer vision with notable applications in self-driving [29] and virtual/augmented reality. To solve the challenge, a diverse set of sensors has been utilized ranging from monocular camera [87], multi-view cameras [46], and depth completion from LiDAR [114]. Although the monocular system is the least expensive, it is the most challenging due to scale ambiguity. The current highest performing monocular methods [296, 97, 163, 135, 79] are reliant on *supervised* training, thus consuming large amounts of labelled depth data. Recently, *self-supervised* methods with photometric supervision have made significant progress by leveraging unlabeled stereo images [82, 87] or monocular videos [325, 260, 305] to approach comparable performance as the supervised methods.

Yet, self-supervised depth inference techniques suffer from high ambiguity and sensitivity in low-texture regions, reflective surfaces, and the presence of occlusion, likely leading to a sub-optimal solution. To reduce these effects, many works seek to incorporate constraints from external modalities. For example, prior works have explored leveraging diverse modalities such as optical

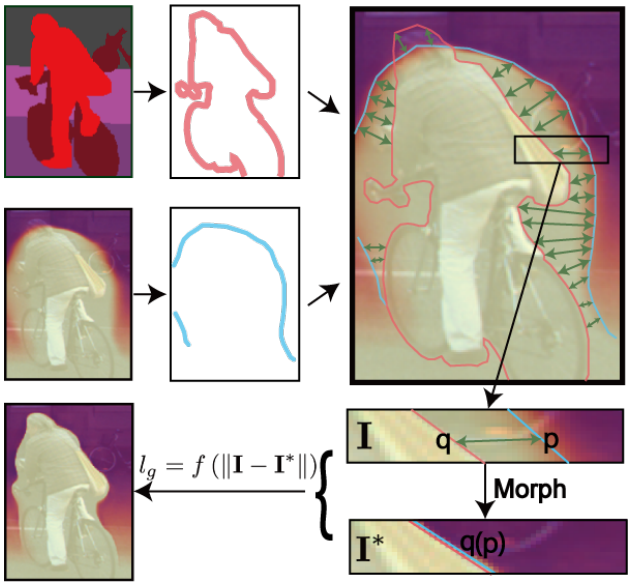


Figure 2.1 We explicitly regularize the depth border to be consistent with segmentation border. A "better" depth $\mathbf{I}^*$ is created through morphing according to distilled point pairs **pq**. By penalizing its difference with the original prediction $\mathbf{I}$ at each training step, we gradually achieve a more consistent border. The morph happens over every distilled pairs but only one pair illustrated, due to limited space.

flow [305], surface normal [297], and semantic segmentation [40, 269, 173, 320]. Optical flow can be naturally linked to depth via ego-motion and object motion, while surface normal can be re-defined as direction of the depth gradient in 3D. Comparatively, semantic segmentation is unique in that, though highly relevant, it is difficult to form definite relationship with depth.

In response, prior works tend to model the relation of semantic segmentation and depth *implicitly* [40, 269, 173, 320]. For instance, [40, 269] show that jointly training a shared network with semantic segmentation and depth can help lean both modalities. [320] learns a transformation between semantic segmentation and depth feature spaces. Despite empirically positive results, such techniques lack clear and detailed explanation for their improvement. Moreover, prior work has yet to explore the relationship from one of the most obvious aspects — the shared borders between segmentation and depth.

Hence, we aim to *explicitly* constrain monocular self-supervised depth estimation to be more consistent and aligned to its segmentation counterpart. We validate the intuition of segmentation being stronger than depth estimation for estimating object boundaries, even compared to depth from

multi-view camera systems [304], thus demonstrating the importance of leveraging this strength (Tab. 2.3). We use the distance between segmentation and depth's edges as a measurement of their consistency. Since this measurement is not differentiable, we can not directly optimize it as a loss. Rather, it is optimized as a "greedy search", such that we iteratively construct a local optimum *augmented* disparity map under the proposed measurement and penalize its discrepancy with the original prediction. The construction of augmented depth map is done via a modified Beier–Neely morphing algorithm[256]. In this way, the estimated depth map gradually becomes more consistent with the segmentation edges within the scene, as demonstrated in Fig. 7.1.

Since we use predicted semantics labels[333], noise is inevitably inherited. To combat this, we develop several techniques to stabilize training as well as improve performance. We also notice recent stereo-based self-supervised methods ubiquitously possess "bleeding artifacts", which are fading borders around two sides of objects. We trace its cause to occlusions in stereo cameras near object boundaries and resolve by integrating a novel stereo occlusion mask into the loss, further enabling quality edges and subsequently facilitating our morphing technique.

Our contributions can be summarized as follows:

◇ We explicitly define and utilize the border constraint between semantic segmentation and depth estimation, resulting in depth more consistent with segmentation.

◇ We alleviate the bleeding artifacts in prior depth methods [88, 87, 40, 191] via proposed stereo occlusion mask, furthering the depth quality near object boundaries.

◇ We advance the state-of-the-art (SOTA) performance of the self-supervised monocular depth estimation task on the KITTI dataset, which for the first time matches SOTA supervised performance in the absolute relative metric.

2.2 Related work

Self-supervised Depth Estimation Self-supervision has been a pivotal component in depth estimation [325, 260, 305]. Typically, such methods require only a monocular image in inference but are trained with video sequences, stereo images, or both. The key idea is to build pixel correspondences from a predicted depth map among images of different view angles then minimize

a photometric reconstruction loss for all paired pixels. Video-based methods [325, 260, 305] require both depth map estimation and ego-motion. While stereo system [82, 87] requires a pair of images captured simultaneously by cameras with known relative placement, reformulating depth estimation into disparity estimation.

We note the photometric loss is subject to two general issues: (1) When occlusions present, via stereo cameras or dynamic scenes in video, an incorrect pixel correspondence can be made yielding sub-optimal performance. (2) There exists ambiguity in low-texture or color-saturated areas such as sky, road, tree leaves, and windows, thereby receiving a weak supervision signal. We aim to address (1) by proposed stereo occlusion masking, and (2) by leveraging additional explicit supervision from semantic segmentation.

Occlusion Problem Prior works in video-based depth estimation [88, 260, 117, 35] have begun to address the occlusion problem. [88] suppresses occlusions by selecting pixels with a minimum photometric loss in consecutive frames. Other works [260, 117] leverage optical flow to account for object and scene movement. In comparison, occlusion in stereo pairs has not received comparable attention in SOTA methods. Such occlusions often result in bleeding depth artifacts when (self-)supervised with photometric loss. [87] partially relieves the bleeding artifacts via a left-right consistency term. Comparatively, [191, 296] incorporates a regularization onto the depth magnitude to suppress the artifacts.

In our work, we propose an efficient occlusion masking based only on a single estimated disparity map, which significantly improves estimation convergence and qualities around dynamic objects' border (Sec. 2.3.2). Another positive side effect is improved edge maps, which facilitates our proposed semantic-depth edge consistency (Sec. 2.3.1).

Using Additional Modalities To address weak supervision in low-texture regions, prior work has begun incorporating modalities such as surface normal [297], semantic segmentation [194, 40, 269, 173], optical flow [260, 117] and stereo matching proxies [278, 247]. For instance, [297] constrains the estimated depth to be more consistent with predicted surface normals. While [278, 247] leverage proxy disparity labels produced by Semi-Global Matching (SGM) algorithms [107, 108], which

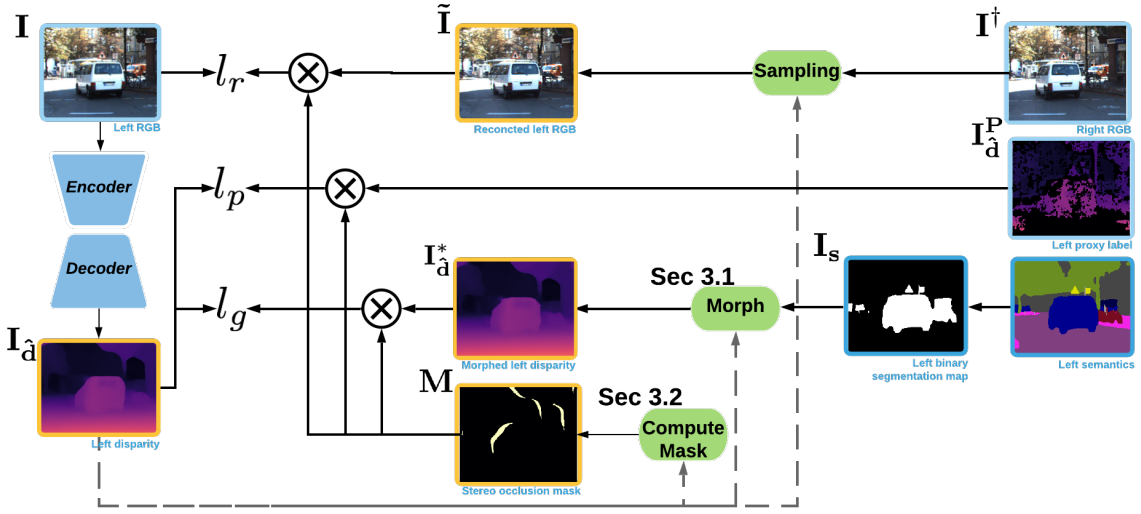


Figure 2.2 **Framework Overview**. The blue box indicates input while yellow box indicates the estimation. The encoder-decoder takes only a left image $\mathbf{I}$, to predict the corresponding disparity $\mathbf{I}_{\hat{\mathbf{d}}}$ which will be converted to depth map $\mathbf{I}_{\mathbf{d}}$. The prediction is supervised via a photometric reconstruction loss l_r, morph loss l_g, and stereo matching proxy loss l_p.

serve as additional psuedo ground truth supervision. In our work, we provide a novel study focusing on constraints from the shared borders between segmentation and depth.

Using Semantic Segmentation for Depth The relationship between depth and semantic segmentation is fundamentally different from the aforementioned modalities. Specifically, semantic segmentation does not inherently hold a definite mathematical relationship with depth. In contrast, surface normal can be interpreted as normalized depth gradient in 3D space; disparity possesses an inverse linear relationship with depth; and optical flow can be decomposed into object movement, ego-motion, and depth estimation. Due to the vague relationship between semantic segmentation and depth, prior work primarily use it in an *implicit* manner.

We classify the uses of segmentation for depth estimation into three categories. Firstly, share weights between semantics and depth branches as in [40, 269]. Secondly, mix semantics and depth features as in [269, 173, 320]. For instance, [269, 173] use a conditional random field to pass information between modalities. Thirdly, [124, 194] opt to model the statistical relationship between segmentation and depth. [124] specifically models the uncertainty of segmentation and

depth to re-weight themselves in the loss function.

Interestingly, no prior work has leveraged the border consistency naturally existed between segmentation and depth. We emphasize that leveraging this observation has two difficulties. First, segmentation and depth only share partial borders. Secondly, formulating a differentiable function to link binarized borders to continuous semantic and depth prediction remains a challenge. Hence, designing novel approaches to address these challenges is our contribution to an explicit segmentation-depth constraint.

2.3 The Proposed Method

We observe recent self-supervised depth estimation methods[278] preserve deteriorated object borders compared to semantic segmentation methods[333] (Tab. 2.3). It motivates us to explicitly use segmentation borders as a constraint in addition to the typical photometric loss. We propose an edge-edge consistence loss l_c (Sec. 2.3.1.1) between depth map and segmentation map. However, as the l_c is not differentiable, we circumvent it by constructing an optimized depth map $\mathbf{I}_\mathbf{d}^*$ and penalizing its difference with original prediction $\mathbf{I_d}$ (Sec. 2.3.3.1). This construction is accomplished via a novel morphing algorithm (Sec. 2.3.1.2). Additionally, we resolve bleeding artifacts (Sec. 2.3.2) for improved border quality and rectify batch normalization layer statistics via a finetuning strategy (Sec. 2.3.3.1). As in Fig. 6.3, our method consumes stereo image pairs and precomputed semantic labels [333] in training, while only requiring a monocular RGB image at inference. It predicts a disparity map $\mathbf{I}_{\hat{\mathbf{d}}}$ and then converted to depth map $\mathbf{I_d}$ given baseline b and focal length f under relationship $\mathbf{I_d} = \frac{f \cdot b}{\mathbf{I}_{\hat{\mathbf{d}}}}$.

2.3.1 Explicit Depth-Segmentation Consistency

To explicitly encourage estimated depth to agree with its segmentation counterpart on their edges, we propose two steps. We first extract matching edges from segmentation $\mathbf{I_s}$ and corresponding depth map $\mathbf{I_d}$ (Sec. 2.3.1.1). Using these pairs, we propose a continuous morphing function to warp all depth values in its inner-bounds (Sec. 2.3.1.2), such that depth edges are aligned to semantic edges while preserving the continuous integrity of the depth map.

2.3.1.1 Edge-Edge Consistency

In order to define the edge-edge consistency, we must firstly extract the edges from both the segmentation map $\mathbf{I_s}$ and depth map $\mathbf{I_d}$. We define $\mathbf{I_s}$ as a binary foreground-background segmentation map, whereas the depth map $\mathbf{I_d}$ consists of continuous depth values. Let us denote an edge $\mathbf{T}$ as the set of pixel $\mathbf{p}$ locations such that:

$$\mathbf{T} = \{\mathbf{p} \mid \left\| \frac{\partial \mathbf{I(p)}}{\partial \mathbf{x}} \right\| > k_1\}, \tag{2.1}$$

where $\frac{\partial \mathbf{I(p)}}{\partial \mathbf{x}}$ is a 2D image gradient at $\mathbf{p}$ and k_1 is a hyperparameter controlling necessary gradient intensity to constitute an edge. In order to highlight clear borders in close-range objects, the depth edge $\mathbf{T_d}$ is extracted from the disparity map $\mathbf{I_{\hat{d}}}$ instead of $\mathbf{I_d}$. Given an arbitrary segmentation edge point $\mathbf{q} \in \mathbf{T_s}$, we denote $\delta(\mathbf{q}, \mathbf{T_d})$ as the distance between $\mathbf{q}$ to its closest point in depth edge $\mathbf{T_d}$:

$$\delta(\mathbf{q}, \mathbf{T_d}) = \min_{\{\mathbf{p} | \mathbf{p} \in \mathbf{T_d}\}} \|\mathbf{p} - \mathbf{q}\| . \tag{2.2}$$

Since the correspondence between segmentation and depth edges do not strictly follow an one-one mapping, we limit it to a predefined local range. We denote the valid set $\mathbf{\Gamma}$ of segmentation edge points $\mathbf{q} \in \mathbf{T_s}$ such that:

$$\mathbf{\Gamma}(\mathbf{T_s} \mid \mathbf{T_d}) = \{\mathbf{q} \mid \forall \mathbf{q} \in \mathbf{T_s},\ \delta(\mathbf{q}, \mathbf{T_d}) < k_2\}, \tag{2.3}$$

where k_2 is a hyperparamter controlling the maximum distance allowed for association. For notation simplicity, we denote $\mathbf{\Gamma_s^d} = \mathbf{\Gamma}(\mathbf{T_s} \mid \mathbf{T_d})$. Then the consistency l_c between the segmentation $\mathbf{T_s}$ and depth $\mathbf{T_d}$ edges is as:

$$l_c(\mathbf{\Gamma}(\mathbf{T_s} \mid \mathbf{T_d}),\ \mathbf{T_d}) = \frac{1}{\left\|\mathbf{\Gamma_s^d}\right\|} \sum_{\mathbf{q} \in \mathbf{\Gamma_s^d}} \delta(\mathbf{q}, \mathbf{T_d}). \tag{2.4}$$

Due to the discretization used in extracting edges from $\mathbf{I_s}$ and $\mathbf{I_d}$, it is difficult to directly optimize $l_c(\mathbf{\Gamma_s^d},\ \mathbf{T_d})$. Thus, we propose a continuous morph function (ϕ and g in Sec. 2.3.1.2) to produce an augmented depth $\mathbf{I_d^*}$, with a corresponding depth edge $\mathbf{T_d^*}$ that minimizes:

$$l_c(\mathbf{\Gamma}(\mathbf{T_s} \mid \mathbf{T_d}),\ \mathbf{T_d^*}). \tag{2.5}$$

Note that the l_c loss is asymmetric. Since the segmentation edge is more reliable, we prefer to use $l_c(\mathbf{\Gamma_s^d},\ \mathbf{T_d^*})$ rather than its inverse mapping direction of $l_c(\mathbf{\Gamma_d^s},\ \mathbf{T_s^*})$.

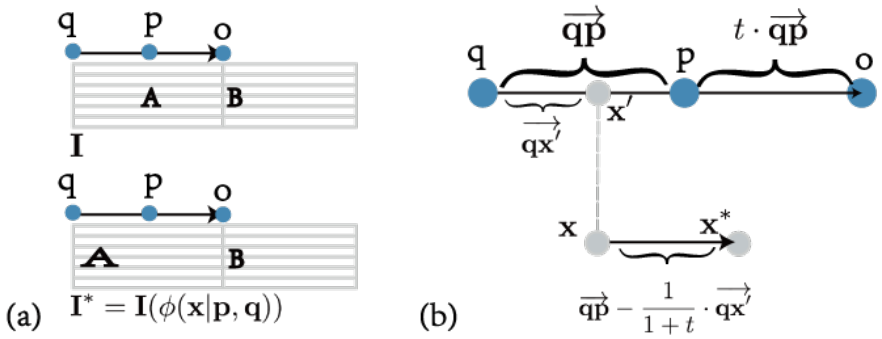


Figure 2.3 The **morph function** $\phi(\cdot)$ morphs a pixel $\mathbf{x}$ to pixel $\mathbf{x}^*$, via Eq. 2.7 and 2.8. (a) A source image $\mathbf{I}$ is morphed to $\mathbf{I}^*$ by applying $\phi(\mathbf{x}|\mathbf{q},\mathbf{p})$ to every pixel $\mathbf{x} \in \mathbf{I}^*$ with the closest pair of segmentation $\mathbf{q}$ and depth $\mathbf{p}$ edge points. (b) we show each term's geometric relationship. The morph warps $\mathbf{x}$ around $\overrightarrow{\mathbf{qo}}$ to $\mathbf{x}^*$ around $\overrightarrow{\mathbf{po}}$. Point $\mathbf{o}$ is controlled by term t in the extended line of $\overrightarrow{\mathbf{qp}}$.

2.3.1.2 Depth Morphing

In the definition of consistence measurement l_c in Eq. (2.5), we acquire a set of associations between segmentation and depth border points. We denote this set as $\mathbf{\Omega}$:

$$\mathbf{\Omega} = \left\{ \mathbf{p} \mid \underset{\{\mathbf{p}|\mathbf{p}\in\mathbf{T_d}\}}{\operatorname{argmin}} \|\mathbf{p}-\mathbf{q}\|, \mathbf{q} \in \mathbf{\Gamma}_\mathbf{s}^\mathbf{d} \right\}. \tag{2.6}$$

Associations in $\mathbf{\Omega}$ imply depth edge $\mathbf{p}$ should be adjusted towards segmentation edge $\mathbf{q}$ to minimize consistence measurement l_c. This motivates us to design a local morph function $\phi(\cdot)$ which maps an arbitrary point $\mathbf{x}$ near a segmentation point $\mathbf{q} \in \mathbf{\Gamma}_\mathbf{s}^\mathbf{d}$ and associated depth point $\mathbf{p} \in \mathbf{\Omega}$ to:

$$\mathbf{x}^* = \phi(\mathbf{x} \mid \mathbf{q},\mathbf{p}) = \mathbf{x} + \overrightarrow{\mathbf{qp}} - \frac{1}{1+t} \cdot \overrightarrow{\mathbf{qx}'}, \tag{2.7}$$

where hyperparameter t controls sample space illustrated in Fig. 2.3, and $\mathbf{x}'$ denotes the point projection of $\mathbf{x}$ onto $\overrightarrow{\mathbf{qp}}$:

$$\mathbf{x}' = \mathbf{q} + (\overrightarrow{\mathbf{qx}} \cdot \hat{\mathbf{qp}}) \cdot \hat{\mathbf{qp}}, \tag{2.8}$$

where $\hat{\mathbf{qp}}$ is the unit vector of the associated edge points. We illustrate a detailed example of $\phi(\cdot)$ in Fig. 2.3.

To promote smooth and continuous morphing, we further define a more robust morph function $g(\cdot)$, applied to every pixel $\mathbf{x} \in \mathbf{I}_\mathbf{d}^*$ as a distance-weighted summation of all morphs $\phi(\cdot)$ for each

associated pair $(\mathbf{q}, \mathbf{p}) \in (\mathbf{\Gamma}_{\mathbf{s}}^{\mathbf{d}}, \mathbf{\Omega})$:

$$g(\mathbf{x} \mid \mathbf{q}, \mathbf{p}) = \sum_{i=0}^{i=|\mathbf{\Omega}|} \frac{w(d_i)}{\sum_{j=0}^{j=|\mathbf{\Omega}|} w(d_j)} \cdot h(d_i) \cdot \phi(\mathbf{x} \mid \mathbf{p}_i, \mathbf{q}_i), \tag{2.9}$$

where d_i is the distance between $\mathbf{x}_i$ and edge segments $\overrightarrow{\mathbf{q}_i\mathbf{p}_i}$. $h(\cdot)$ and $w(\cdot)$ are distance-based weighting functions: $w(d_i) = (\frac{1}{m_3+d_i})^{m_4}$, and $h(d_i) = \text{Sigmoid}(-m_1 \cdot (d_i - m_2))$, where m_1, m_2, m_3, m_4 are predefined hyperparameters. $w(\cdot)$ is a relative weight compromising morphing among multiple pairs, while $h(\cdot)$ acts as an absolute weight ensuring each pair only affects local area. Implementation wise, $h(\cdot)$ makes pairs beyond ~7 pixels negligible, facilitating $g(\mathbf{x} \mid \mathbf{q}, \mathbf{p})$ linear computational complexity.

In summary, $g(\mathbf{x} \mid \mathbf{q}, \mathbf{p})$ can be viewed as a more general Beier–Neely [256] morph, due to inclusion of $h(\cdot)$. We align depth map better to segmentation via applying $g(\cdot)$ morph to pixels of its disparity map $\mathbf{x} \in \mathbf{I}_{\hat{\mathbf{d}}}^{*}$, creating a segmentation-augmented disparity map $\mathbf{I}_{\hat{\mathbf{d}}}^{*}$:

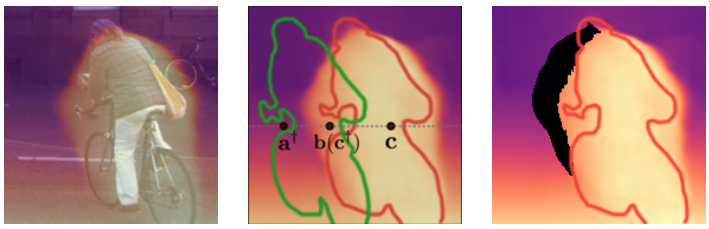
$$\begin{aligned} &\mathbf{I}_{\hat{\mathbf{d}}}^{*}(\mathbf{x}) = \mathbf{I}_{\hat{\mathbf{d}}}(g(\mathbf{x} \mid \mathbf{q}, \mathbf{p})) \\ \vdash \quad &\forall (\mathbf{p}, \mathbf{q}) \in (\mathbf{\Omega}, \Gamma), \ \mathbf{p} = \phi(\mathbf{q}). \end{aligned} \tag{2.10}$$

Next we may transform the edge-to-edge consistency term l_c into the minimization of difference between $\mathbf{I}_{\hat{\mathbf{d}}}$ and the segmentation-augmented $\mathbf{I}_{\hat{\mathbf{d}}}^{*}$, as detailed in Sec. 2.3.3.1. A concise proof of $\mathbf{I}_{\mathbf{d}}^{*}$ as local minimum of l_c under certain condition is in the supplementary material (**Suppl.**).

2.3.2 Stereo Occlusion Mask

Bleeding artifacts are a common difficulty in self-supervised stereo methods [88, 87, 40, 191]. Specifically, bleeding artifacts refer to instances where the estimated depth on surrounding foreground objects wrongly expands outward to the background region. However, few works provide detailed analysis of its cause. We illustrate the effect and an overview of our stereo occlusion mask in Fig. 2.4.

Let us define a point $\mathbf{b} \in \mathbf{I}_{\mathbf{d}}$ near the boundary of an object and corresponding point $\mathbf{b}^{\dagger} \in \mathbf{I}_{\mathbf{d}}^{\dagger}$ in the right stereo view. When point $\mathbf{b}^{\dagger}$ is occluded by a foreground point $\mathbf{c}^{\dagger}$ in the right stereo, a photometric loss will seek a similar non-occluded point in the right stereo, *e.g.*, the objects' left boundary $\mathbf{a}^{\dagger}$, since no exact solution may exist for occluded pixels. Therefore, the



(a) (b) (c)

Figure 2.4 (a) Overlays disparity estimation over the input image showing typical **bleeding artifacts**. (b) We denote the **red** object contour from the left view $\mathbf{I}$ and **green** object contour from the right view $\mathbf{I}^\dagger$. Background point $\mathbf{b}$ is visible in the left view, yet its corresponding right point $\mathbf{b}^\dagger$ is occluded by an object point $\mathbf{c}^\dagger$. Thus, this point is incorrectly supervised by photometric loss l_r to look for the nearest background pixel (*e.g.*, $\mathbf{a}^\dagger$) leading to a bleeding artifact in (a). (c) We depict occluded region detected via Eq. 2.11.

disparity value at point b will be $\hat{d}^*_{\mathbf{b}} = \left\|\overrightarrow{\mathbf{a}^\dagger \mathrm{b}}\right\| = x_{\mathrm{b}} - x_{\mathbf{a}^\dagger}$, where x is the horizontal location. Since background is assumed farther away than foreground points, generally a false supervision has the quality such that the occluded background disparity will be significantly larger than its (unknown) ground truth value. As $\mathbf{b}$ approaches $\mathbf{a}^\dagger$ the effect is lessened, creating a fading effect.

To alleviate the bleeding artifacts, we form an occlusion indicator matrix $\mathbf{M}$ such that $\mathbf{M}(x, y) = 1$ if the pixel location (x, y) has possible occlusions in the stereo view. For instance, in the left stereo image $\mathbf{M}$ is defined as:

$$\mathbf{M}(x,y) = \begin{cases} 1 & \min\limits_{i \in (0, W-x]} \left(\mathbf{I}_{\hat{\mathbf{d}}}(x+i, y) - \mathbf{I}_{\hat{\mathbf{d}}}(x, y) - i\right) \geq k_3 \\ 0 & \text{otherwise,} \end{cases} \tag{2.11}$$

where W denotes predefined search width and k_3 is a threshold controlling thickness of the mask.

The disparity value in the left image represents the horizontal left distance of each pixel to be moved. As the occlusion is due to pixels in its right, we intuitively perform our search in one direction. Additionally, we can view occlusion as when neighbouring pixels on its right move too much left and cover itself. In this way, occlusion can be detected as $\min\limits_{i \in (0, W-x]} \left(\mathbf{I}_{\hat{\mathbf{d}}}(x+i, y) - \mathbf{I}_{\hat{\mathbf{d}}}(x, y) - i\right) \geq 0$. Considering bleeding artifacts in Fig. 2.4, we use k_3 to counter large incorrect disparity values of occluded background pixels. The regions indicated by $\mathbf{M}$ are then masked when computing a reconstruction loss (Sec. 2.3.3.1).

Cita.	Method	PP	Data	H × W	Size (Mb)	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
[296]	Yang *et al.*	✓	D†S	256 × 512	-	0.097	0.734	4.442	0.187	0.888	0.958	0.980
[97]	Guo *et al.*		D*DS	256 × 512	79.5	0.097	0.653	4.170	0.170	0.889	**0.967**	**0.986**
[163]	Luo *et al.*		D*DS	192 × 640 crop	1,562	0.094	0.626	4.252	0.177	0.891	0.965	0.984
[135]	Kuznietsov *et al.*		DS	187 × 621	324.8	0.113	0.741	4.621	0.189	0.862	0.960	**0.986**
[79]	Fu *et al.*		D	385 × 513 crop	399.7	0.099	0.593	**3.714**	**0.161**	0.897	0.966	**0.986**
[137]	Lee *et al.*		D	352 × 1,216	563.4	**0.091**	**0.555**	4.033	0.174	**0.904**	**0.967**	0.984
[87]	Godard *et al.*	✓	S	256 × 512	382.5	0.138	1.186	5.650	0.234	0.813	0.930	0.969
[167]	Mehta *et al.*		S	256 × 512	-	0.128	1.019	5.403	0.227	0.827	0.935	0.971
[193]	Poggi *et al.*	✓	S	256 × 512	954.3	0.126	0.961	5.205	0.220	0.835	0.941	0.974
[312]	Zhan *et al.*	✗	MS	160 × 608	-	0.135	1.132	5.585	0.229	0.820	0.933	0.971
[161]	Luo *et al.*		MS	256 × 832	160	0.128	0.935	5.011	0.209	0.831	0.945	0.979
[191]	Pillai *et al.*	✓	S	384 × 1,024	-	0.112	0.875	4.958	0.207	0.852	0.947	0.977
[247]	Tosi *et al.*	✓	S	256 × 512 crop	511.0	0.111	0.867	4.714	0.199	0.864	0.954	0.979
[40]	Chen *et al.*	✓	SC	256 × 512	-	0.118	0.905	5.096	0.211	0.839	0.945	0.977
[88]	Godard *et al.*	✓	MS	320 × 1,024	59.4	0.104	0.775	4.562	0.191	0.878	0.959	0.981
[278]	Watson *et al.* (ResNet18)	✓	S	320 × 1,024	59.4	0.099	0.723	4.445	0.187	0.886	0.962	0.981
	Ours (ResNet18)	✓	SC†	320 × 1,024	59.4	0.097	0.675	4.350	0.180	0.890	0.964	0.983
[278]	Watson *et al.* (ResNet50)	✓	S	320 × 1,024	138.6	0.096	0.710	4.393	0.185	0.890	0.962	0.981
	Ours (ResNet50)	✓	SC†	320 × 1,024	138.6	**0.091**	**0.646**	**4.244**	**0.177**	**0.898**	**0.966**	**0.983**

Table 2.1 **Depth Estimation Performance**, on KITTI Stereo 2015 dataset eigen splits [71] capped at 80 meters. The Data column denotes: D for ground truth depth, D† for SLAM auxiliary data, D* for synthetic depth labels, S for stereo pairs, M for monocular video, C for segmentation labels, C† for predicted segmentation labels. PP denotes post-processing. Size refers to the model size in Mb, which could be different depend on implementation language.

2.3.3 Network and Loss Functions

Our network is comprised of an encoder-decoder, identical to the baseline [278]. It takes in a monocular RGB image and predicts corresponding disparity map which is later converted to depth map under known camera parameters.

2.3.3.1 Loss Functions

The overall loss function is comprised of three terms:

$$l = l_r(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) + \lambda_2 l_g(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) + \lambda_1 l_p(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})), \tag{2.12}$$

where l_r denotes a photometric reconstruction loss, l_g a morphing loss, l_p a stereo proxy loss [278], and $\mathbf{x}$ are the non-occluded pixel locations, *i.e.*, $\{\mathbf{x} \mid \mathbf{M}(\mathbf{x}) = 0\}$. λ_1 and λ_2 are the weights of terms. We emphasize that exclusion will not prevent learning of object borders. *E.g.,* in Fig. 2.4(c), although the pixel **b** in cyclist's left border is occluded, the network can still learn to estimate depth from a visible and highly similar pixel $\mathbf{a}^\dagger$ in the stereo counterpart, as both left and right view images are respectively fed into the encoder in training, similar to prior self-supervised works [278, 88].

Following [88], we define the l_r reconstruction loss as:

$$l_r\left(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})\right) = \alpha \frac{1 - \text{SSIM}\left(\mathbf{I}(\mathbf{x}), \tilde{\mathbf{I}}(\mathbf{x})\right)}{2} + (1-\alpha)|\mathbf{I}(\mathbf{x}) - \tilde{\mathbf{I}}(\mathbf{x})|, \tag{2.13}$$

Method	Area	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Watson *et al.* [278]	O	0.085	0.507	3.684	0.159	0.909
	W	0.096	0.712	4.403	0.185	0.890
	N	0.202	2.819	8.980	0.342	0.702
Ours (ResNet50)	O	0.081	0.466	3.553	0.152	0.916
	W	0.091	0.646	4.244	0.177	0.898
	N	0.192	2.526	8.679	0.324	0.712

Table 2.2 **Edge vs. Off-edge Performance**. We evaluate the depth performance for O-off edge, W-whole image, N-near edge.

which consists of a pixel-wise mix of SSIM [276] and L_1 loss between an input left image $\mathbf{I}$ versus the reconstructed left image $\tilde{\mathbf{I}}$, which is re-sampled according to predicted disparity $\mathbf{I}_{\hat{\mathbf{d}}}$. The α is a weighting hyperparameter as in [87, 278].

We minimize the distance between depth and segmentation edges by steering the disparity $\mathbf{I}_{\hat{\mathbf{d}}}$ to approach the semantic-augmented disparity $\mathbf{I}^*_{\hat{\mathbf{d}}}$ (Eq. 2.10) in a logistic loss:

$$l_g(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) = \mathbf{w}(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) \cdot \log(1 + |\mathbf{I}^*_{\hat{\mathbf{d}}}(\mathbf{x}) - \mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})|), \tag{2.14}$$

where $\mathbf{w}(\cdot)$ is a function to downweight image regions with low variance. It is observed that the magnitude of the photometric loss (Eq. 2.13) varies significantly between textureless and rich texture image regions, whereas the morph loss (Eq. 2.14) is primarily dominated by the border consistency. Moreover, the morph is itself dependent on an *estimated* semantic psuedo ground truth $\mathbf{I}_\mathbf{s}$ [333] which may include noise. In consequence, we only apply the loss when the photometric loss is comparatively improved. Hence, we define the weighting function $\mathbf{w}(\cdot)$ as:

$$\mathbf{w}(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) = \begin{cases} \text{Var}(\mathbf{I})(\mathbf{x}) & \text{If } l_r(\mathbf{I}^*_{\hat{\mathbf{d}}}(\mathbf{x})) < l_r(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) \\ 0 & \text{otherwise,} \end{cases} \tag{2.15}$$

where $\text{Var}(\mathbf{I})$ computes pixel-wise RGB image variance in a 3×3 local window. Note that when a noisy semantic estimation $\mathbf{I}_\mathbf{s}$ causes l_r to degrade, the pixel location is ignored.

Following [278], we incorporate a stereo proxy loss l_p which we find helpful in neutralizing noise in estimated semantics labels, defined similarly to Eq. 2.14 as:

$$l_p(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) = \begin{cases} \log(1 + |\mathbf{I}^\mathbf{p}_{\hat{\mathbf{d}}} - \mathbf{I}_{\hat{\mathbf{d}}}|) & \text{If } l_r(\mathbf{I}^\mathbf{p}_{\hat{\mathbf{d}}}(\mathbf{x})) < l_r(\mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x})) \\ 0 & \text{otherwise,} \end{cases} \tag{2.16}$$

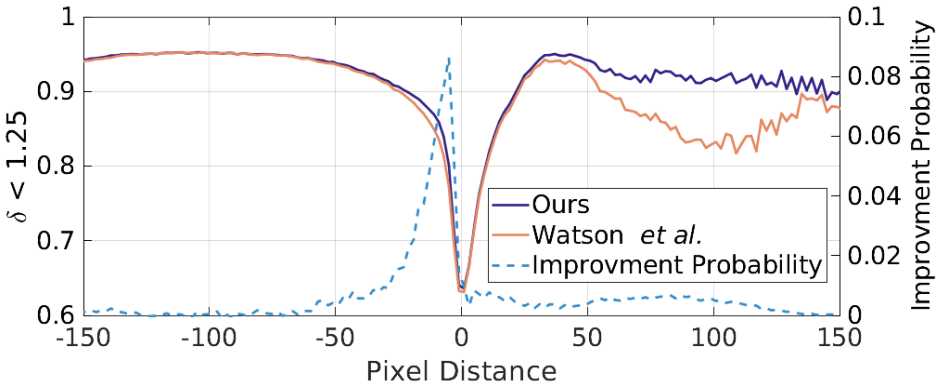


Figure 2.5 Left axis: Metric $\delta < 1.25$ as a function of distance off segmentation edges in background $(-x)$ and foreground $(+x)$. compared to [278]. Right axis: improvement distribution against distance. Our gain mainly comes from near-edge background area but not restricted to it.

where $\mathbf{I}^{\mathbf{p}}_{\hat{\mathbf{d}}}$ denotes the stereo matched proxy label generated by the Semi-Global Matching (SGM) [107, 108] technique.

Finetuning Loss: We further finetune the model to regularize the batch normalization [115] statistics to be more consistent to an identity transformation. As such, the prediction becomes less sensitive to the exponential moving average, following inspiration from [226] denoted as: $l_{\text{bn}} = \left\| \mathbf{I}_{\hat{\mathbf{d}}}(\mathbf{x}) - \mathbf{I}'_{\hat{\mathbf{d}}}(\mathbf{x}) \right\|^2$, where $\mathbf{I}_{\hat{\mathbf{d}}}$ and $\mathbf{I}'_{\hat{\mathbf{d}}}$ denote predicted disparity with and without batch normalization, respectively.

2.3.3.2 Implementation Details

We use PyTorch [187] for training, and preprocessing techniques of [88]. To produce the stereo proxy labels, We follow [278]. Semantic segmentation is precomputed via [333], in an ensemble way with default settings at a resolution of $320 \times 1{,}024$. Using semantics definition in Cityscapes [48], we set object, vehicle, and human categories as foreground, and the rest as background. This allows us to convert a semantic segmentation mask to a binary segmentation mask $\mathbf{I}_{\mathbf{s}}$. We use a learning rate of $1e^{-4}$ and train the joint loss (Eq. 2.12) for 20 epochs, starting with ImageNet pretrained weights. After convergence, we apply l_{bn} loss for 3 epochs at a learning rate of $1e^{-5}$. We set $t = \lambda_1 = 1$, $\lambda_2 = 5$, $k_1 = 0.11$, $k_2 = 20$, $k_3 = 0.05$, $m_1 = 17$, $m_2 = 0.7$, $m_3 = 1.6$, $m_4 = 1.9$, and $\alpha = 0.85$.

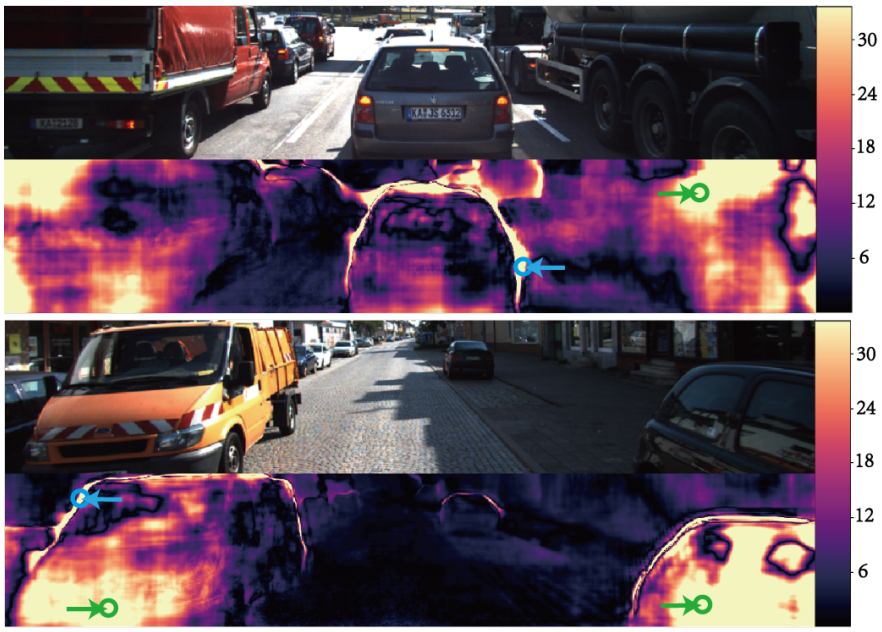


Figure 2.6 Input image and the disagreement of estimated disparity between our method and [278]. Our method impacts both borders (←) and inside (→) of objects.

2.4 Experiments

We first present the comprehensive comparison on the KITTI benchmark, then analyze our results, and finally ablate various design choices of the proposed method.

KITTI Dataset: We compare our method against SOTA works on KITTI Stereo 2015 dataset, a comprehensive urban autonomous driving dataset providing stereo images with aligned LiDAR data. We utilize the eigen splits, evaluated with the standard seven KITTI metrics [71] with the crop of Garg [82] and a standard distance cap of 80 meters [87]. Readers can refer to [71] for explanation of used metrics.

Depth Estimation Performance: We show a comprehensive comparison of our method to the SOTA in Tab. 2.1. Our framework outperforms prior methods on each of the seven metrics. For a fair comparison, we utilize the same network structure as [88, 278]. We consider that approaching the performance of supervised methods is an important goal of self-supervised techniques. Notably, our method is *the first self-supervised method matching SOTA supervised performance*, as seen in the absolute relative metric in Tab. 2.1. Additionally, We emphasize our method improves on the $\delta < 1.25$ from 0.890 to 0.898, thereby reducing the gap between supervised and unsupervised

Category	Method	Morph	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Unsupervised	Watson *et al.*[278]	✗	0.097	0.734	4.454	0.187	0.889	0.961	0.981
		✓	0.096 ↓	0.700 ↓	4.401 ↓	0.184 ↓	0.891 ↑	0.963 ↑	0.982 ↑
Supervised	Lee *et al.* [137]	✗	0.088	0.490	3.677	0.168	0.913	0.969	0.984
		✓	0.088	0.488 ↓	3.666 ↓	0.168	0.913	0.970 ↑	0.985 ↑
Stereo	Yin *et al.*[304]	✗	0.049	0.366	3.283	0.153	0.948	0.971	0.983
		✓	0.049	0.365 ↓	3.254 ↓	0.152 ↓	0.948	0.971	0.983

Table 2.3 Comparison of algorithms if coupled with an segmentation network during inference. Given the segmentation predicted at inference, we apply morph defined in Sec. 2.3.1.2 to depth prediction. The improved metric is marked in green.

methods by relative ~60% ($= 1 - \frac{0.904-0.898}{0.904-0.890}$). We further demonstrate a consistent performance gain with two variants of ResNet (Tab. 2.1), demonstrating our method's robustness to the backbone architecture capacity.

We emphasize our contributions are orthogonal to most methods including stereo and monocular training. For instance, we use noisy segmentation *predictions*, which can be further enhanced by pairing with stronger segmentation or via segmentation annotations. Moreover, recall that we do not use the monocular training strategy of [88] or additional stereo data such as Cityscapes, and utilize a substantially smaller network (*e.g.*, 138.6 vs. 563.4 MB [137]), thereby leaving more room for future enhancements.

Depth Performance Analysis: Our method aims to explicitly constrain the estimated depth edges to become similar to segmentation counterparts. Yet, we observe that the improvements to the depth estimation, while being emphasised near edges, are distributed in *more* spatial regions. To understand this effect, we look at three perspectives.

Firstly, we demonstrate that depth performance is the most challenging near edges using the $\delta < 1.25$ metric. We consider a point $\mathbf{x}$ to be near an edge point $\mathbf{p}$ if below averaged edge consistence l_c, that is $\mid \mathbf{x} - \mathbf{p} \mid \leq 3$. We demonstrate the depth performance of off-edge, whole image, and near edge regions in Tab. 2.2. Although our method has superior performance on whole, *each* method degrades near an edge ($\downarrow$ ~0.18 on δ from W to N), reaffirming the challenge of depth around object boundaries.

Secondly, we compare metric $\delta < 1.25$ against baseline [278] in the left axes of Fig. 2.5. We observe improvement from background around object borders (px~ -5) and from foreground inside

Loss	Morph	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	✗	0.102	0.754	4.499	0.187	0.884	0.962	0.982
Baseline + **M**	✗	0.101	0.762	4.489	0.186	0.887	0.962	0.982
Baseline + **M** + l_g	✗	0.099	0.736	4.462	0.185	0.889	0.963	0.982
	✓	0.098	0.714	4.421	0.183	0.890	0.964	0.982
Baseline + **M** + l_g + Finetune	✗	0.098	0.692	4.393	0.182	0.889	0.963	**0.983**
	✓	**0.097**	**0.674**	**4.354**	**0.180**	**0.891**	**0.964**	**0.983**

Table 2.4 Ablation study of the proposed method. ✓ indicates morphing during inference.

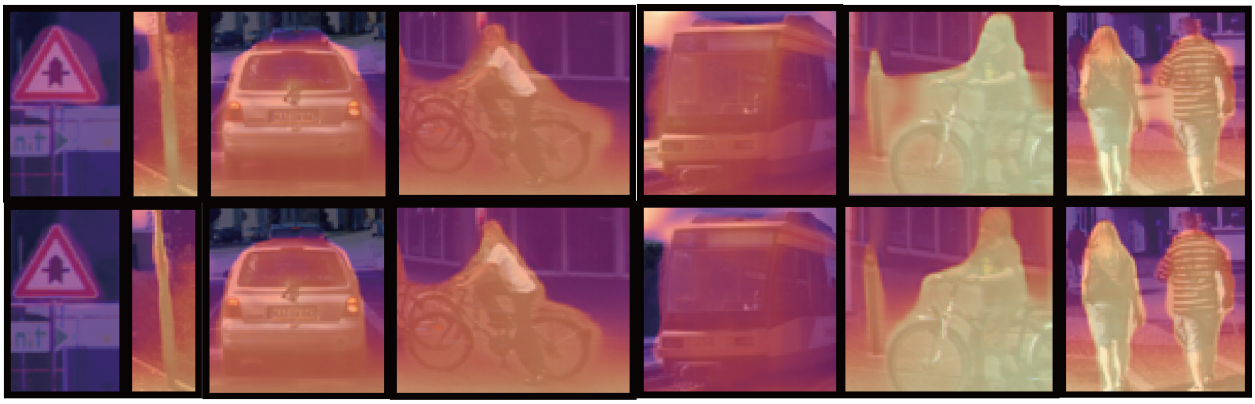


Figure 2.7 Compare the quality of estimated depth around foreground objects between [278] (top) and ours (bottom).

objects (px $\geq$ 30). This is cross-validated in Fig. 2.6 which visualizes the disagreements between ours and baseline [278]. Our method impacts near the borders ($\leftarrow$) as well as inside of objects ($\rightarrow$) in Fig. 2.6.

Thirdly, we view the improvement as a normalized probability distribution, as illustrated in right axes of Fig. 2.5. It peaks at around -5 px, which agrees with the visuals of Fig. 2.7 where originally the depth spills into the background but becomes close to object borders using ours. Still, the improvement is consistently positive and generalized to entire distance range. Such findings reaffirm that our improvement is both *near and beyond* the edges in a general manner.

Depth Border Quality: We examine the quality of depth borders compared to the baseline [278], as in Fig. 2.7. The depth borders of our proposed method is significantly more aligned to object boundaries. We further show that for SOTA methods, even without training our models, applying our morphing step at inference leads to performance gain, when coupled with a segmentation network [333] (trained with only 200 domain images). As in Tab. 2.3, this trend holds for unsupervised, supervised, and multi-view depth inference systems, implying that typical depth

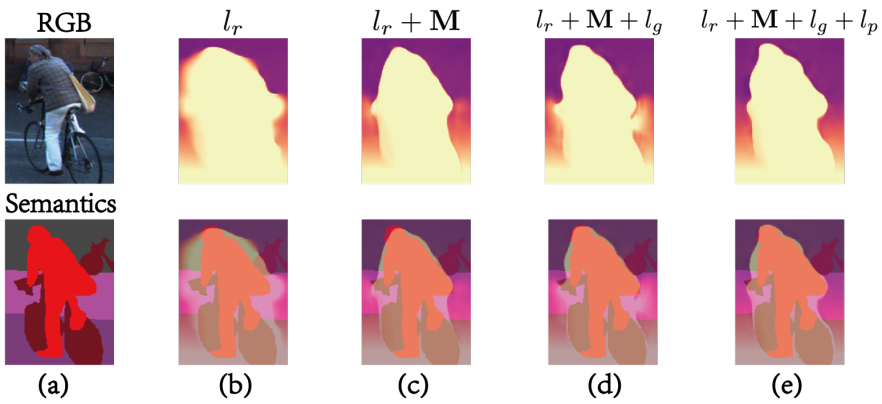


Figure 2.8 (a) input image and segmentation, (b-e) estimated depth (top) and with overlaid segmentation (bottom) for various ablation settings, as defined in Tab. 7.5.

Model	Finetune	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Godard *et al*. [88]	✗	0.104	0.775	4.562	0.191	0.878
	✓	0.103	0.731	4.531	0.188	0.878
Watson *et al*. [278]	✗	0.096	0.710	4.393	0.185	0.890
	✓	0.094	0.676	4.317	0.180	0.892

Table 2.5 Improvement after finetuning of different models.

methods can struggle with borders, where our morphing can augment. However, we find that the inverse relationship using depth edges to morph segmentation is harmful to border quality.

Stereo Occlusion Mask: To examine the effect of our proposed stereo occlusion masking (Sec. 2.3.2), we ablate its effects (Tab. 7.5). The stereo occlusion mask $\mathbf{M}$ improves the absolute relative error (0.102 $\rightarrow$ 0.101) and $\delta < 1.25$ (0.884 $\rightarrow$ 0.887). Upon applying stereo occlusion mask during training, we observe the bleeding artifacts are significantly controlled as in Fig. 2.8 and in **Suppl.** Fig. 3. Hence, the resultant borders are stronger, further supporting the proposed consistency term l_c and morphing operation.

Morph Stabilization: We utilize estimated segmentation [333] to define the segmentation-depth edge morph. Such estimations inherently introduce noise and destablization in training for which we propose a $\mathbf{w}(\mathbf{x})$ weight to provide less attention to low image variance and ignore any regions which degrades photometric loss (Sec. 2.3.3.1). Additionally, we ablate the specific help from stereo proxy labels in stabilizing training in Fig. 2.8 (d) & (e) and **Suppl.** Fig. 3.

Finetuning Strategy: To better understand the effect of our finetuning strategy (Sec. 2.3.3.1) on performance, we ablate using [88, 278] and our method, as shown in Tab. 7.5 and 2.5.

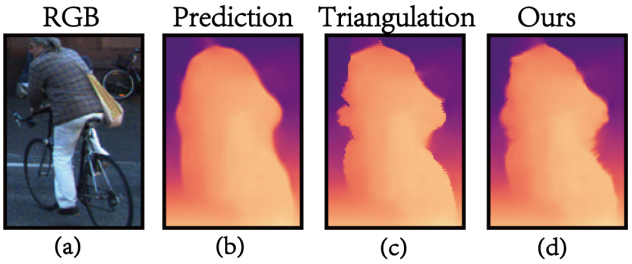


Figure 2.9 Comparison of depth of initial baseline (b), triangularization (c), and proposed morph (d).

Method	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Ours (Triangularization)	0.697	4.379	0.180	0.895
Ours (Proposed)	0.686	4.368	0.180	0.895

Table 2.6 Our morphing strategy versus triangularization.

Each ablated method achieves better performance after applying the finetuning, suggesting the technique is general.

Morphing Strategy: We explore the sensitivity of our morph operation (Sec. 2.3.1), by comparing its effectiveness against using triangularization to distill point pair relationships. We accomplish this by first forming a grid over the image using anchors. Then define corresponding triangularization pairs between the segmentation edge points paired with two anchors. Lastly, we compute an affine transformation between the two triangularizations. We analyze the technique vs. our proposed morphing strategy qualitatively in Fig. 2.9 and quantitatively in Tab. 2.6. Although the methods have subtle distinctions, the triangularization morph is generally inferior, as highlighted by the RMSE metrics in Tab. 2.6. Further, the triangularization morphing forms boundary errors with acute angles which introduce more noise in the supervision signal, as exemplified in Fig. 2.9.

2.5 Conclusions

We present a depth estimation framework designed to explicitly consider the mutual benefits between two neighboring computer vision tasks of self-supervised depth estimation and semantic segmentation. Prior works have primarily considered this relationship implicitly. In contrast, we propose a morphing operation between the borders of the predicted segmentation and depth, then use this morphed result as an additional supervising signal. To help the edge-edge consistency quality, we identify the source problem of bleeding artifacts near object boundaries then propose

a stereo occlusion masking to alleviate it. Lastly, we propose a simple but effective finetuning strategy to further boost generalization performance. Collectively, our method advances the state of the art on self-supervised depth estimation, matching the capacity of supervised methods, and significantly improves the border quality of estimated depths.

CHAPTER 3

PMATCH: PAIRED MASKED IMAGE MODELING FOR DENSE GEOMETRIC MATCHING

Dense geometric matching determines the dense pixel-wise correspondence between a source and support image corresponding to the same 3D structure. Prior works employ an encoder of transformer blocks to correlate the two-frame features. However, existing monocular pretraining tasks, *e.g.*, image classification, and masked image modeling (MIM), can not pretrain the cross-frame module, yielding less optimal performance. To resolve this, we reformulate the MIM from reconstructing a single masked image to reconstructing a pair of masked images, enabling the pretraining of transformer module. Additionally, we incorporate a decoder into pretraining for improved upsampling results. Further, to be robust to the textureless area, we propose a novel cross-frame global matching module (CFGM). Since the most textureless area is planar surfaces, we propose a homography loss to further regularize its learning. Combined together, we achieve the State-of-The-Art (SoTA) performance on geometric matching.

3.1 Introduction

When a 3D structure is viewed in both a source and a support image, for a pixel (or keypoint) in the source image, the task of geometric matching identifies its corresponding pixel in the support image. This task is a cornerstone for many downstream vision applications, *e.g.* homography estimation [65], structure-from-motion [209], visual odometry estimation [72] and visual camera localization [28].

There exist both sparse and dense methods for geometric matching. The sparse methods [67, 199, 255, 160, 63, 201, 152, 234, 234] only yield correspondence on sparse or semi-dense locations while the dense methods [252, 250, 68] estimate pixel-wise correspondence. They primarily differ in that the sparse methods embed a keypoint detection or a global matching on discrete coordinates, which underlyingly assumes a unique mapping between source and support frames. Yet, the existence of textureless surfaces introduces multiple similar local patches, disabling keypoint detection or causing ambiguous matching results. Dense methods, though facing similar challenges at the

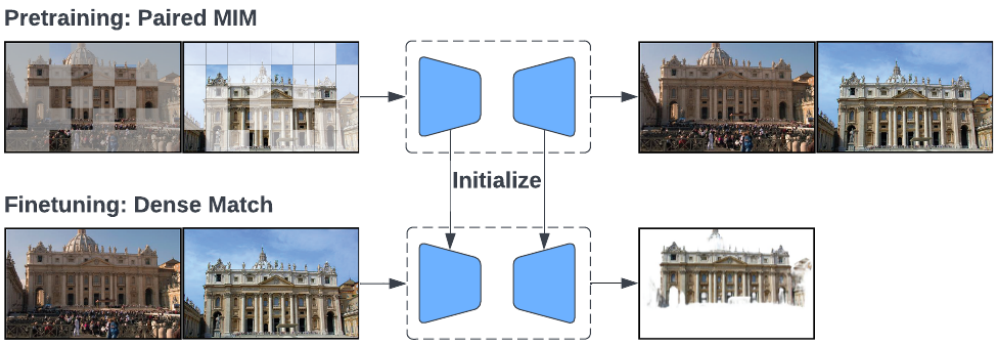


Figure 3.1 Most vision tasks start with a pretrained network. In geometric matching, the unique network components processing two-view features cannot benefit from the monocular pretraining task, *e.g.*, image classification, and masked image modeling (MIM). As in the figure, this work enables the pretraining of a matching model via reformulating MIM from reconstructing a single masked image to reconstructing a pair of masked images.

coarse level, alleviate it with the additional fine-level local context and smoothness constraint. Until recently, the dense methods demonstrate a comparable or better geometric matching performance over the sparse methods [252, 250, 68].

A relevant task to dense geometric matching is the optical flow estimation [241]. Both tasks estimate dense correspondences, whereas the optical flow is applied over consecutive frames with the constant brightness assumption.

In geometric matching [234, 38], apart from the encoder encodes source and support frames into feature maps, there exist transformer blocks which correlate two-frame features, *e.g.*, the LoFTR module [234]. Since these network components consume two-frame inputs, the monocular pretraining task, *e.g.*, the image classification and masked image modeling (MIM) defined on ImageNet dataset, is unable to benefit the network. This limits both the geometric matching performance and its generalization capability.

To address this, we reformulate the MIM from single masked image reconstruction to paired masked images reconstruction, *i.e.*, pMIM. Paired MIM benefits the geometric matching as both tasks rely on the cross-frame module to correlate two frames inputs for prediction.

With a pretrained encoder, the decoder in dense geometric matching is still randomly initialized. Following the idea of pretraining encoder, we extend pMIM pretraining to the decoder. As part functionality of decoder is to upsample the coarse-scale initial prediction to the same resolution as

input, we also task the decoder in pMIM to upsample the coarse-scale reconstruction to its original resolution. Correspondingly, we consist the decoder as stacks of the depth-wise convolution except for the last prediction head. With the depth-wise decoder, when transferring from pMIM to geometric matching, we duplicate the decoder along the channel dimension to finish the initialization. To this end, there exists only a small number of components in the decoder randomly initialized, we pretrain the rest network components using synthetic image pair augmentation [250].

To further improve the dense geometric matching performance, we propose a cross-frame global matching module (CFGM). In CFGM, we first compute the correlation volume. We model the correspondences of coarse scale pixels as a summation over the discrete coordinates in the support frame, weighted by the softmaxed correlation vector. However, this modeling fails when multiple similar local patches exit. As a solution, we impose positional embeddings to the discrete coordinates and decode with a deep architecture to avoid ambiguity. Meanwhile, we notice that the textureless surfaces are mostly planar structures described by a low-dimensional 8 degree-of-freedom (DoF) homography matrix. We thus design a homography loss to augment the learning of the low DoF planar prior.

We summarize our contributions as follows:

• We introduce the paired masked image modeling pretext task, pretraining both the encoder and decoder of a dense geometric matching network.

• We propose a novel cross-frame global matching module that is robust to textureless local patches. Since the most textureless patches are planar structures, we augment their learning with a homography loss.

• We outperform dense and sparse geometric matching methods on diverse datasets.

3.2 Related works

3.2.1 Pretraining and Finetuning

Pretraining and finetuning is an effective paradigm in vision tasks. Supervised image classification has been one of the most widely adopted pretraining methods. An encoder [104, 225, 112], *e.g.*, ResNet [104], together with a few fully connected (FC) layers is trained for image classifica-

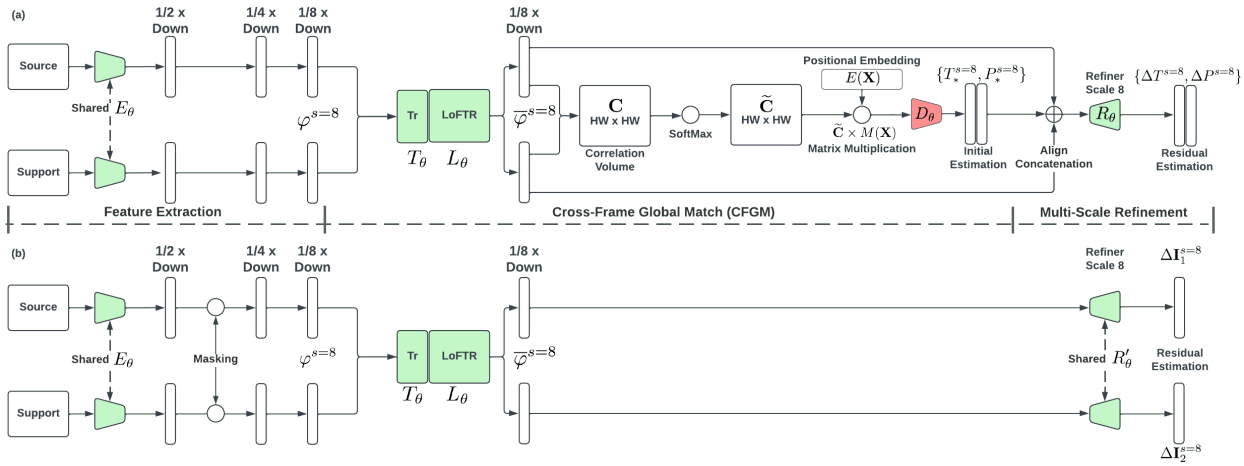


Figure 3.2 **Methodology Overview.** In (a), we illustrate the proposed dense geometric matching network. After extracting the multi-scale feature with the encoder E_θ, we extend the LoFTR module with (1) Transformer blocks T_θ and (2) positional embeddings with an appended decoder D_θ to remove the ambiguity when multiple local patches exist. In (b), we show the proposed paired MIM pretext task. We apply image masking at the scale $s = 2$, and recover the masked images with the transformer blocks. In (a), network D_θ (in red) is not included in pMIM pretraining. In dense matching, R_θ takes in the stack of source and the aligned support frame feature. In the pretext task, R'_θ only takes in the source frame feature. Thus, R'_θ is a sub-graph of R_θ. We detail how to initialize R_θ using R'_θ in Fig. 3.3. The residual refinement at other scales repeats the process at scale $s = 8$ but consumes feature embeddings of other scales, skipped for simplicity.

tion using a large-scale dataset, *e.g.*, ImageNet [59]. After converging, the encoder is used as the initialization in the downstream vision tasks.

Apart from supervised classification tasks, there are self-supervised methods producing discriminative feature representation. Inspired by BYOL [92], DINO [34] introduces a self-supervised mean-teacher knowledge distillation task. It encourages the prediction consistency between a student and teacher model where the teacher is an exponential moving average of the student model. The pretrained ViT model embeds explicit information of semantic segmentation, which is not observed in a supervised counterpart. Other self-supervised pretraining methods include color transformation [44], geometric transformation [44], Jigsaw Puzzle [171], feature frame prediction [185], *etc*.

Among the self-supervised learning tasks, masked image modeling (MIM) [261, 290, 8, 324, 294, 103] achieves SoTA finetuning performance on ImageNet [59]. The task introduces Masked Language Modeling used in NLP domain to vision, reconstructing an image from its masked input.

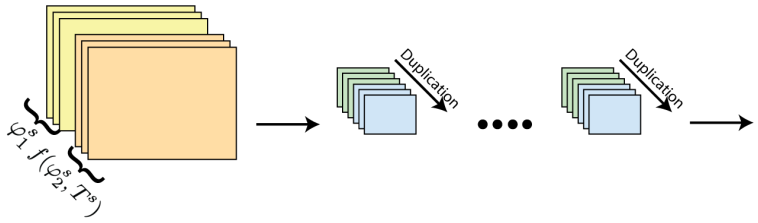


Figure 3.3 **Resolution of the Discrepancy between R_θ and R'_θ.** We adopt stacks of the depth-wise convolution in the refinement module, *i.e.*, each convolution kernel only works with one channel of the input feature maps. This makes refiner R'_θ in pretexting a sub-graph of refiner R_θ in finetuning. While transferring from the pretexting task to finetuning task, the input feature map concatenates an extra aligned support frame feature $f(\varphi_2^s, T^s)$. As the bilinear sampling f imposes minimal distribution change, we duplicate the kernel weight along the channel dimension.

While iGPT [39], ViT [64], and BEiT [8] adopt sophisticated paradigm in modeling, MAE [103] and SimMIM [292] show that directly regressing the masked continuous RGB pixels can achieve competitive results. Typically, they focus on pretraining the encoder, adopting an asymmetric design where only a shallow decoder head is appended.

In this paper, we reformulate MIM from reconstructing a single image to the paired images, reducing the domain gap between the pretexting task and the downstream geometric matching. As a result, we extend the benefit of MIM pretraining to the task of dense geometric matching.

3.2.2 Sparse Geometric Matching

There are detector-based and detector-free sparse geometric matching methods. Classic works are detector based, and employ the nearest neighbor (NN) match using the hand-crafted feature on detected keypoints, *e.g.*, SIFT [160], SURF [14], and ORB [202]. Both keypoint detection and feature extraction are improved by data-driven deep models [63, 67, 184, 199, 302, 63]. Later, [204, 201, 255] propose to replace the naive NN match by graph neural network based differentiable matching.

While the detector based methods operate on keypoints, the detector free methods, *e.g.* LoFTR [234] and ASpanFormer [38] operate all-to-all matching on coarse-scale discrete grid locations. Still, their matching depends on the correlation between features, yielding ambiguous results when multiple local patches exist. We improve LoFTR from two perspectives. First, we extend the LoFTR module to the proposed cross-frame global matching module to benefit from the MIM pretexting

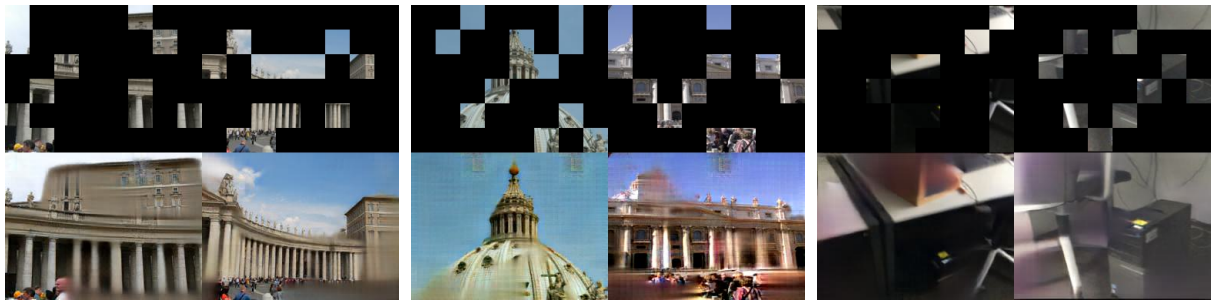


Figure 3.4 **Visual Quality of the paired MIM pretext task.** Visualized cases are from the MegaDepth and the ScanNet dataset.

task. Second, we alleviate the ambiguity caused by similar local patches by imposing positional embeddings over the low-dimensional 2D coordinates. A decoder is then employed to resolve the ambiguity.

3.2.3 Dense Geometric Matching

DGC-Net [168] regresses dense correspondences from a global correlation volume at a limited resolution. GLU-Net [249] increases the resolution with a global-local correlation layer. GOCor [248] further improves GLU-Net [249] by replacing the correlation layer with online optimization. Other methods, such as RANSAC Flow [217], iteratively recover a homography transformation to reduce the visual difference between the source and support images.

Though dense methods estimate more correspondences than sparse methods, it is less favored for geometric matching. Until recently, PDC Net+ [250] and DKM [68] close the gap between dense and sparse methods. Both methods model the dense match as probability functions. PDC Net+ adopts a mixture Laplacian distribution while DKM models with the Gaussian Process (GP). Furthermore, they estimate a confidence score to remove false positive results. We follow [250, 68] in the confidence estimation. However, instead of applying probabilistic regression, we keep the correlation based explicit matching process. This saves the computation of the inverse matrix required in the GP Regression of DKM. Also, we apply a unique architecture design to benefit from the MIM pretexting task.

Figure 3.5 **Visual Quality of the Reconstruction.** We visualize 4 reconstructed images using estimated dense correspondences. In each group, from left to right is the source image, support image, and the reconstructed image. The areas of low confidence are filled with white color. In ScanNet where the confidence groundtruth is not available, we use forward-backward flow consistency mask as a replacement.

3.3 Method

In this section, we first introduce the proposed dense geometric matching method. Then we discuss how to pretext the network via the paired masked image modeling. Fig. 6.3 depicts our framework in finetuning and pretexting stages.

3.3.1 Dense Geometric Matching

Dense geometric matching computes the dense correspondences between the source image $\mathbf{I}_1$ and support image $\mathbf{I}_2$. Under the estimated correspondences T, source image $\mathbf{I}_1$ can be recovered from support image $\mathbf{I}_2$ by applying bilinear sampling at T. Since the dense correspondences between $\mathbf{I}_1$ and $\mathbf{I}_2$ is not guaranteed to exist at each pixel location, we follow [68] in estimating confidence P to indicate the fidelity of the prediction.

Feature Extraction. As shown in Fig. 6.3, we adopt a multi-scale ResNet-based [104] feature extractor E_θ. Taking the source frame $\mathbf{I}_1$ as an example, we produce the multiscale feature embeddings as:

$$\{\varphi_1^{s=2}, \varphi_1^{s=4}, \varphi_1^{s=8}\} = E_\theta(\mathbf{I}_1). \tag{3.1}$$

For the input image $\mathbf{I}_1$ of resolution $H \times W$, the scale s indicates a feature map of resolution $H/s \times W/s$.

Cross-Frame Global Matching The cross-frame global matching module (CFGM) is designed to accomplish coarse-scale geometric matching. To benefit from the MIM pretext task, we first

process the scale $s = 8$ feature map $\varphi_1^{s=8}$ with the transformer block [123]:

$$\{\overline{\varphi}_1^{s=8\prime}, \overline{\varphi}_2^{s=8\prime}\} = T_\theta(\varphi_1^{s=8}, \varphi_2^{s=8}). \tag{3.2}$$

In the pretraining stage, the masked feature map is recovered by the appended transformer blocks. Then, we follow LoFTR [234] in using linear transformer blocks to correlate the source and support frame feature:

$$\{\overline{\varphi}_1^{s=8}, \overline{\varphi}_2^{s=8}\} = L_\theta(\varphi_1^{s=8\prime}, \varphi_2^{s=8\prime}). \tag{3.3}$$

To compute the global matching results, we first compute the 4D correlation volume $\mathbf{C}\left(\overline{\varphi}_1^{s=8}, \overline{\varphi}_2^{s=8}\right) \in \mathbb{R}^{H/8\times W/8\times H/8\times W/8}$, where:

$$C_{ijkl} = \sum_h \frac{1}{\gamma}\left(\overline{\varphi}_1^{s=8}\right)_{ijh} \cdot \left(\overline{\varphi}_2^{s=8}\right)_{klh}, \tag{3.4}$$

where γ is a temperature scalar. The coarse matches are computed as a summation over pixel locations $\mathbf{X} \in \mathbb{R}^{(H/8)(W/8)\times 2}$ weighted by the softmaxed correlation volume. That is, after the correlation volume $\mathbf{C}$ being reshaped to $\mathbf{C} \in \mathbb{R}^{(H/8)(W/8)\times(H/8)(W/8)}$, we apply the softmax:

$$\widetilde{C_{ij}} = \text{softmax}(C_{ij}). \tag{3.5}$$

Here, element C_{ij} is a size $(H/8)(W/8)\times 1$ vector. We conclude the coarse global matching results as:

$$T_*^{s=8} = \widetilde{\mathbf{C}} \times \mathbf{X}. \tag{3.6}$$

Note, Eqn. 3.6 will cause ambiguous results when multiple similar textureless local patches exist, *i.e.*, multiple peak values in softmaxed correlation vector $\widetilde{C_{ij}}$. To resolve this, we modify Eqn. 3.6 with:

$$T_*^{s=8}, P_*^{s=8} = D_\theta\left(\widetilde{\mathbf{C}} \times M(\mathbf{X})\right), \tag{3.7}$$

where $M(\mathbf{X})$ is cosine positional embeddings with learnable tokens [234, 68], projecting the 2D pixel locations to a high dimensional space to avoid ambiguity when multiple similar patches exist. The decoder D_θ decodes $T_*^{s=8}$, initial correspondences estimation at scale $s = 8$, and $P_*^{s=8}$, initial confidence estimation.

Methods	Venue	Dense Match PCK ↑			Run-
		@1 px	@3 px	@5 px	time (ms)
RANSAC-FLow [217]	ECCV'20	53.47	83.45	86.81	3,596
PDC-Net [314]	CVPR'21	71.81	89.36	91.18	1,017
PDC-Net+ [250]	Arxiv'21	**74.51**	**90.69**	**92.10**	1,017
LIFE [113]	Arxiv'21	39.98	76.14	83.14	**78**
GLU-Net-GOCor [248]	NeurIPS'20	57.77	78.61	82.24	**71**
PDC-Net [314]	CVPR'21	68.95	84.07	85.72	88
PDC-Net+ [250]	Arxiv'21	72.41	86.70	88.12	88
PMatch (Ours)	CVPR'23	**79.83**	**95.18**	**96.52**	124

Table 3.1 **MegaDepth Dense Geometric Matching.** The running time of all methods is measured at the resolution 480×480. The upper and lower groups are methods running multiple or single times. [Key: Red color marks Best, Blue color marks the Second Best]

Multi-Scale Refinement We follow [68] in using the multi-scale refinement module:

$$\Delta T^s, \Delta P^s = R_\theta(\varphi_1^s, f(\varphi_2^s, T^s)), \tag{3.8}$$

where function $f(\cdot)$ indicates the bilinear interpolation to align the support frame feature using the current estimated correspondences T^s, shown in Fig. 6.3. To accommodate the transfer between pretexting and finetuning stage, we apply depth-wise convolution [68] in R_θ. We detail the discussion in Fig. 3.3 and Sec.3.3.2. The correspondences and confidence on the next scale are initialized with the bilinear upsampling.

3.3.2 Paired MIM Pretraining

Paired Masked Image Modeling (MIM) MIM is extensively adopted in image classification task [103, 292]. An image classification network can be further improved after MIM pretexting. As shown in Fig. 7.1 and 3.4, the network reconstructs the input from randomly masked feature embeddings at a specific scale. In this work, we investigate the benefit of pretraining both the encoder and decoder under MIM. Compared to only pretraining the encoder, pretraining the whole network further reduces the domain gap between pretexting and finetuning tasks.

Masking Strategy We follow SimMIM [292] in using randomly selected 32×32 mask patches with a predefined masking ratio r_1 and r_2 for source and support frames. For source view, given the feature embeddings $\varphi_1^{s=2}$ output by the extractor E_θ at scale $s = 2$, we apply the randomly generated mask $\mathbf{w}$ to mask out the feature embeddings, *i.e.*:

$$\varphi_1^{s=2'} = \varphi_1^{s=2} * (1 - \mathbf{w}) + \mathbf{x} * \mathbf{w}, \tag{3.9}$$

Category	Methods	Venue	Pose Estimation AUC ↑		
			@5°	@10°	@20°
Sparse W/ Detector	SuperGlue [204]	CVPR'19	42.2	61.2	75.9
	SGMNet [144]	Pattern'20	40.5	59.0	72.6
Sparse Wo/ Detector	DRC-Net [152]	ICASSP'22	27.0	42.9	58.3
	LoFTR [234]	CVPR'21	52.8	69.2	81.2
	QuadTree [239]	ICLR'22	54.6	70.5	82.2
	MatchFormer [271]	ACCV'22	53.3	69.7	81.8
	ASpanFormer [38]	ECCV'22	55.3	71.5	83.1
Dense	PDC-Net+ [250]	Arxiv'19	43.1	61.9	76.1
	DKM [68]	CVPR'23	**60.5**	**74.9**	**85.1**
	PMatch (Ours)	CVPR'23	**61.4**	**75.7**	**85.7**

Table 3.2 **MegaDepth Two-View Camera Pose Estimation.** We compare three groups of methods following SuperGlue [204] in evaluation. The pose AUC error is reported. Our method shows substantial improvement. [Key: Red color marks Best, Blue color marks the Second Best]

where $\mathbf{x}$ is the learnable mask tokens. Note, our extractor E_θ starts from a 3×3 convolution kernel to avoid leakage of the masked patches.

Prediction Heads Different from SimMIM [292], our prediction heads include most network components of the decoder. We complete the masked feature embeddings with the transformer as:

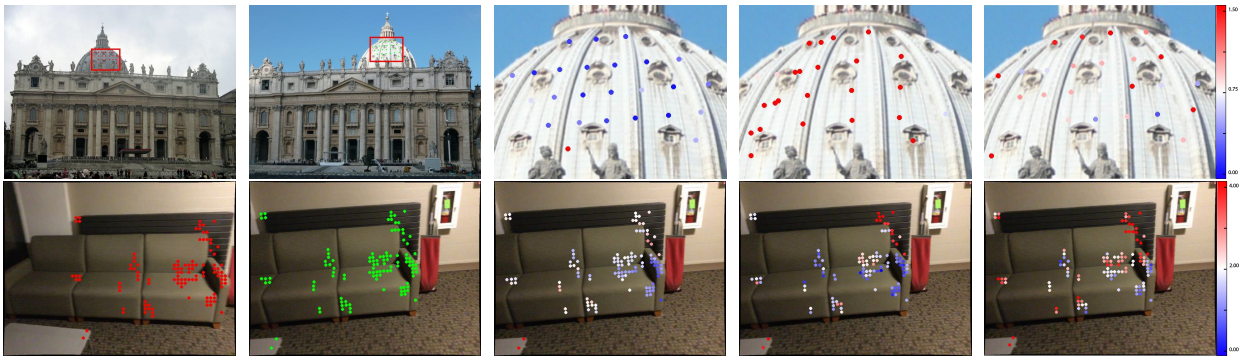
$$\varphi_1^{s=8\prime} = T_\theta(\varphi_1^{s=8}). \tag{3.10}$$

Here, we use the same notation as Eqn. 3.2 since both indicate image features at the scale $s = 8$. Note that the subsequent network component LoFTR is a series of linear transformer blocks [123] which reduce the quadratic computational complexity to linear. However, empirically we find the linear transformer poorly recovers the masked patches. We thus append the transformer blocks.

As shown in Fig. 6.3, after Eqn. 3.10, we feed the completed feature map to CFGM. Note the refiner between the two stages is different. Instead of taking a stacked feature map (Eqn. 3.8), in pretexting we only take in a single feature map:

$$\Delta\mathbf{I}_1^s = R'_\theta(\varphi_1^s), \quad \Delta\mathbf{I}_2^s = R'_\theta(\varphi_2^s). \tag{3.11}$$

To account for the difference between Eqn. 3.8 and Eqn. 3.11, we apply depth-wise convolution, where each convolution kernel operates on one channel of the feature map, shown in Fig. 3.3. Since $f(\varphi_2^s, T^s)$ in Eqn. 3.8 is a resampled support frame feature, it imposes minimal distribution difference to φ_2^s. Then, while transferring from the pretexting task to the downstream task, we only need to duplicate the channel of R_θ to complete the initialization. We follow SimMIM [292]



(a) Source Frame $\mathbf{I}_1$ (b) Supp. Frame $\mathbf{I}_2$ (c) PMatch (Ours) (d) DKM [68] (e) LoFTR [234]

Figure 3.6 **Visual Comparisons.** We conduct the visual comparison against the SoTA dense [68] and sparse [234] methods on the MegaDepth and the ScanNet datasets. The color from **blue** to **red** indicates an increment in the end-point-error (L2 error).

in estimating full resolution residual RGB images in each scale of the decoder. We visualize the reconstructed paired masked images in Fig. 3.4.

Network Components not included in pMIM Since the feature map at $s = 2$ contains little information about masked patches, the pretraining only includes refinement modules at scale $s = 4$ and $s = 8$. Furthermore, the CFGM decoder D_θ and part of R_θ are not included. We pretrain the rest network component with synthetic image pairs [250].

Prediction Objective Set the accumulated reconstruction at each scale s as $\mathbf{I}^s$, we regress the raw pixel value with an l_1 loss:

$$\mathcal{L}_M = \sum_s \frac{1}{N}(|\mathbf{I}_1^s - \mathbf{I}_1|_1 + |\mathbf{I}_2^s - \mathbf{I}_2|_1), \tag{3.12}$$

where N is the number of unmasked pixels.

3.3.3 Dense Geometric Matching Loss

Homography Loss The image correspondences between two planar structures are constrained by a 3×3 homography matrix $\mathbf{H}$ with 8 DoF. Compared to correspondences estimation over arbitrary shapes, the correspondences in planar structures possess a lower rank. Given a surface normal $\mathbf{n}$

computed using the depth gradient [177], the homography of the pixel can be computed as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^\intercal \\ \mathbf{h}_2^\intercal \\ \mathbf{h}_3^\intercal \end{bmatrix} = \mathbf{K}_1 \left(\mathbf{R} + \frac{\mathbf{t}^\top}{d} \mathbf{n} \right) \mathbf{K}_2^{-1}, \tag{3.13}$$

where the $\mathbf{K}_1$ and $\mathbf{K}_2$ are intrinsic matrices of $\mathbf{I}_1$ and $\mathbf{I}_2$, $\mathbf{R}$ and $\mathbf{t}$ are camera rotation and translation, and d is the pixel depth. We randomly sample K anchor points $\{\mathbf{p}_m \mid 1 \leq m \leq K\}$. For each anchor point $\mathbf{p}_m$, we sample K candidate points $\{\mathbf{q}_n^m \mid 1 \leq n \leq K\}$. We determine a co-planar indicator matrix O^+ of size $K \times K$ to suggest all co-planar pairs. We use the normal consistency, point-to-plane distance, and homography consistency to compute the co-planar groundtruth, detailed in Supp. Finally, we apply a gradient-based penalty, penalizing the correspondences difference between the estimation and the groundtruth.

$$\mathcal{L}_h^s = \frac{1}{|O^+|} \sum_{O^+_{\mathbf{p},\mathbf{q}}=1} | \left(T^s_{\mathbf{p}} - T^s_{\mathbf{q}} \right) - \left(\overline{T}^s_{\mathbf{p}} - \overline{T}^s_{\mathbf{q}} \right) |_1. \tag{3.14}$$

Global Matching Loss Following [234], we minimize a binary cross-entropy loss over the correlation volume $\mathbf{C}$ after a dual-softmax operation:

$$\widetilde{C_{ijkl}}' = \text{softmax}(C_{ij}) \cdot \text{softmax}(C_{kl}), \tag{3.15}$$

where C_{ij} and C_{kl} are $(H/8)(W/8) \times 1$ vectors. The loss is defined as:

$$\begin{aligned} \mathcal{L}_g = &- \frac{1}{|\mathcal{M}^+|} \sum_{ijkl \in \mathcal{M}^+} \log \widetilde{C_{ijkl}}' \\ &- \frac{1}{|\mathcal{M}^-|} \sum_{ijkl \in \mathcal{M}^-} \log \left(1 - \widetilde{C_{ijkl}}' \right), \end{aligned} \tag{3.16}$$

where $\mathcal{M}^+$ and $\mathcal{M}^-$ are groundtruth indicator matrix of size $H \times W \times H \times W$ indicating whether a source frame pixel (i, j) pairs with a target frame pixel (k, l).

Refinement Loss Following [68], we supervise both correspondences and confidence on each scale of the predictions,

$$\mathcal{L}_r^s = \frac{1}{|P^+|} \sum_{ij \in P^+} \left| T^s_{ij} - \overline{T}^s_{ij} \right|_2, \tag{3.17}$$

Category	Methods	Venue	Pose Estimation AUC ↑		
			@5°	@10°	@20°
Sparse	SuperGlue [204]	CVPR'19	16.2	33.8	51.8
W/ Detector	SGMNet [144]	PR'20	15.4	32.1	48.3
Sparse Wo/ Detector	DRC-Net [152]	ICASSP'22	7.7	17.9	30.5
	LoFTR [234]	CVPR'21	22.0	40.8	57.6
	QuadTree [239]	ICLR'22	24.9	44.7	61.8
	MatchFormer [271]	ACCV'22	24.3	43.9	61.4
	ASpanFormer [38]	ECCV'22	**25.6**	46.0	63.3
Dense	PDC-Net+ [250]	Arxiv'19	20.2	39.4	57.1
	DKM [68]	CVPR'23	**29.4**	**50.7**	**68.3**
	PMatch (Ours)	CVPR'23	**29.4**	**50.1**	**67.4**

Table 3.3 **ScanNet Two-View Camera Pose Estimation.** We follow SuperGlue [204] in the testing protocol. The pose AUC error is reported. Our method achieves clear improvement over other baselines. [Key: Red color marks Best, Blue color marks the Second Best]

where P^+_{ij} is a $H \times W$ matrix that indicates whether a valid pair is found at pixel location ij in the source frame. Similarly, the loss of confidence is defined as:

$$\mathcal{L}^s_c = -\frac{1}{|\mathcal{P}^+|}\sum_{ij\in P^+}\log(P_{ij}) - \frac{1}{|\mathcal{P}^-|}\sum_{ij\in P^-}\log(1-P_{ij}). \tag{3.18}$$

Total Loss The total loss is a weighted summation of proposed losses:

$$\mathcal{L} = \frac{1}{4}\sum_s (L^s_r + w_c\mathcal{L}^s_c) + w_g\cdot\mathcal{L}_g + \frac{1}{4}w_h\sum_s\mathcal{L}^s_h. \tag{3.19}$$

The constant 4 comes from the four scales $s = \{1, 2, 4, 8\}$ set in our paper.

3.4 Experiments

We first compare with other SoTA dense matching methods on the MegaDepth dataset. Then, to comprehensively reflect the contributions from both the density and accuracy of geometric matching, we follow [234, 68] in using the two-view relative camera pose estimation performance as the metric. We report on both the outdoor scenario MegaDepth [146] dataset and the indoor scenario ScanNet [52] dataset. We additionally evaluate on the HPatches [7] and the YFCC100m [243] datasets to demonstrate the generalizability of the model.

3.4.1 Implementation Details

Pretext stage From DeMoN [258], BlendedMVS [300], HyperSim [200], ARKitScenes [13], and TartanAir [275] datasets, we collect a pretraining dataset of 1, 281, 167 image pairs, *i.e.*, the same size as ImageNet [59]. Each pair is collected with a fixed frame index interval. In the pretraining

Methods	Venue	Pose Estimation AUC ↑			Pose Estimation mAP ↑		
		@5°	@10°	@20°	@5°	@10°	@20°
RANSAC-Flow [217]	ECCV'20	-	-	-	64.9	73.3	81.6
PDC-Net [252]	CVPR'21	35.7	55.8	72.3	63.9	73.0	81.2
PDC-Net+ [250]	Arxiv'21	37.5	58.1	74.5	**67.4**	**76.6**	**84.6**
OANet [54]	ICCV'19	-	-	-	52.2	-	-
CoAM [282]	CVPR'21	-	-	-	55.6	66.8	-
PDC-Net [252]	CVPR'21	32.2	52.6	70.1	60.5	70.9	80.3
PDC-Net+ [250]	Arxiv'21	34.8	55.4	72.6	63.9	73.8	82.7
ASpanFormer [38]	ECCV'22	**44.5**	**63.8**	**78.4**	-	-	-
PMatch (Ours)	CVPR'23	**45.7**	**65.2**	**79.8**	**75.9**	**83.1**	**89.3**

Table 3.4 **YFCC100m Two-View Camera Pose Estimation.** The upper group runs multiple times, while the lower group runs a single time. We follow [314] in the evaluation and preprocessing, reporting both pose AUC and mAP errors. [Key: Red color marks Best, Blue color marks the Second Best]

dataset, we train the model using a batchsize of 128 under the resolution 192×256. We use the Adam optimizer [127] with a learning rate $2e^{-4}$, running for 250k steps on 2× A100 GPUs. We stack 1 transformer layer. We initialize the masking ratio $r_1 = 75\%$ and $r_2 = 75\%$. The masking operation applies to the ResNet, causing significantly different batch statistics between masked and unmasked inputs. Since the downstream task takes the unmasked image, we linearly reduce the support frame masking ratio r_2 to 0 and use a different batch normalization layer for support view, resolving the batch statistics difference. We also apply the synthetic image pair augmentation introduced in [250].

Finetuning stage Our model trains with a batchsize of 16 at the resolution 544×720. The learning rate is set to $4e^{-4}$, running 250k steps with a warmup of 25k steps. On 4× A100 GPUs, we train for 5 days with the Adam optimizer. We follow [234] in sampling the paired images, weighted by the sequence length and overlap ratio. The softmax temperature γ is 0.1. We set loss weight w_g to 0.7 and w_h to 0.02. We sample 600×600 points for homography loss L_h.

3.4.2 Datasets

MegaDepth MegaDepth [146] collects over 10 thousand images of worldwide landmarks from the Internet. The collected images are processed by COLMAP [209] to produce groundtruth poses and depthmaps. The dataset collects images of significant visual contrast due to lighting conditions, view angles, and imaging devices. This imposes challenges to geometric matching.

ScanNet [52] is a large-scale indoor dataset with 1,613 videos captured by RGB-D cameras. There are challenging textureless indoor scenes for geometric matching.

YFCC100m [243] is a large multi-media dataset. A subset of 72 reconstructions of tourist landmarks is generated with groundtruth poses and depthmap.

Hpatches provides the pair of one source and five support images taken under different view angles and lighting conditions with groundtruth homography transformation.

3.4.3 Dense Geometric Matching

We follow the RANSAC-Flow [217] in training and testing split on the MegaDepth dataset. The PCK scores in Tab. 3.1 refer to the thresholded keypoints accuracy. We divide the baseline methods into single and multiple run methods. Note, the baseline methods PDC Net [252] and PDC Net+ [250] consume the additional synthetic data generated using COCO [149] instance segmentation label. For PCK @1px, we outperform the SoTA single and multiple run methods by an absolute margin of 4.89% and 6.99% respectively. Meanwhile, we are about **8× faster** than SoTA baselines while suppassing SoTA performance.

3.4.4 Two-View Camera Pose Estimation

Evaluation Protocol In the MegaDepth, ScanNet, and Hpatches datasets, we follow the evaluation protocol of [204, 234, 68] in reporting the pose accuracy AUC curve thresholded at 5, 10, and 20 degrees. In the YFCC100m dataset, we follow the protocol of RANSAC-Flow [217], additionally reporting the pose mAP value. The pose estimation is considered an outlier if its maximum degree error of translation or rotation exceeds the threshold. The two-view relative pose is estimated using the five-point algorithm [181] with RANSAC [62] via the OpenCV implementation [27].

Baseline Methods We compare with three groups of the methods, *i.e.*, sparse methods with detector [204, 144], sparse methods without detector [152, 234, 239, 271, 38] and dense methods [250, 68, 217, 252, 54, 282]. For sparse detector based methods, we use SuperPoint [63] as the keypoint detector. For dense methods, we further categorize them into single-run and multiple-run methods. For multiple-run methods, *e.g.*, RANSAC-Flow [217], it repeats the prediction while reducing the visual difference with an estimated homography transformation. Among baselines,

AspanFormer [38] is a recent publicly available sparse detector-free method, improving LofTR with a sophisticated attention mechanism.

Outdoor Dataset We test our method on the outdoor dataset MegaDepth. We follow the training and validation split of [204, 234, 68]. The evaluation split contains $1,500$ paired images randomly selected from the scene 0015 and 0022. As shown in Tab. 3.2, we achieve an absolute improvement of 0.9% over the recent SoTA dense method DKM [68]. Compared to the SoTA sparse method ASpanFormer [38], we maintain an improvement of 6.1%.

Indoor Dataset We test our method on the indoor dataset ScanNet. We follow [68] in training and testing protocol, resizing images to 480×640. The validation split of ScanNet consists of $1,500$ image pairs [204]. In Tab. 7.3, we maintain competitive performance with the SoTA dense method DKM [68] and outperform SoTA sparse method by 1.4%.

Generalization to YFCC100m We use the MegaDepth trained model to test on YFCC100m [243] dataset. We follow the preprocessing steps of [314], evaluated on 4 scenes with a total of $1,000$ images. During the evaluation, we resample the input images of the shorter side to 480. Tab. 3.4 shows that our method can achieve a superior generalization ability, maintaining an improvement of 1.2% over SoTA sparse methods [38].

Generalization to HPatches Following LoFTR [234], we test the MegaDepth dataset trained model on HPatches. In evaluation, the homography matrix is estimated using OpenCV's implementation. We compare correspondences accuracy computed using the groundtruth and estimated homography. The image pairs in HPatches have lighting differences or view differences. The pattern is different from the training dataset MegaDepth. Under the unseen testing scenario, our model generalizes best among baselines.

3.5 Ablation Study

Qualitative Comparison The visual quality of reconstructed images using the predicted correspondences is visualized in Fig. 3.5. We conduct a visual comparison with other SoTA dense and sparse methods in Fig. 3.6. In Row 1, (c), and (d), compared to DKM [68], the proposed CFGM module achieves correct initial correspondences. In Row 1, (c), and (e), compared to LoFTR [234],

Category	Methods	Venue	Pose Estimation AUC ↑		
			@3px	@5px	@10px
Sparse W/ Detector	D2Net [67]	CVPR'19	23.2	35.9	53.6
	R2D2 [199]	NeurIPS'19	50.6	63.9	76.8
	DISK [255]	NeurIPS'20	52.3	64.9	78.9
	SuperGlue	CVPR'19	53.9	68.3	81.7
Sparse Wo/ Detector	NCNet [201]	ECCV'20	48.9	54.2	67.1
	DRC-Net [152]	ICASSP'22	50.6	56.2	68.3
	LoFTR [234]	CVPR'21	65.9	75.6	**84.6**
Dense	DKM [68]	CVPR'23	**71.3**	**80.6**	**88.5**
	PMatch (Ours)	CVPR'23	**71.9**	**80.7**	**88.5**

Table 3.5 **Hpatches Homography Estimation.** We follow [234] in evaluation protocol. We report the corner point AUC error under the estimated homography matrix. [Key: Red color marks Best, Blue color marks the Second Best]

Baseline	CFGM	L_H	pMIM Encoder $(E_\theta, T_\theta, L_\theta)$	pMIM Decoder (R_θ)	Pose Estimation AUC ↑		
					@5°	@10°	@20°
✓					56.1	71.5	83.0
✓	✓				57.5	72.6	83.9
✓	✓	✓			57.9	72.9	84.1
✓	✓	✓	✓		60.6	75.0	85.3
✓	✓	✓	✓	✓	**61.4**	**75.7**	**85.7**

Table 3.6 **Ablation Studies on MegaDepth.** The baseline method is the network in Fig. 6.3 with only a LoFTR module, *i.e.*, without the other components of CFGM. The ablation is conducted under the same training and testing resolution as Tab. 3.2. Bold marks best.

multi-scale dense refinement improves fine-scale correspondence accuracy. In Row 2, (c), (d), and (e), our CFGM and homography loss achieve accurate correspondence estimation on textureless planar surface, *e.g.*, the black wall behind the sofa.

Running Time Evaluated on an RTX 2080 Ti GPU, we run 160 ms for an image of 480×640 while LoFTR [234] runs 116 ms and DKM [68] runs 148 ms. Our model runs similarly compared to the baselines. The running time comparison to other dense methods is in Tab. 3.1.

Benefit of the paired MIM pretraining Shown in Tab. 7.5, with the paired MIM pretext task, the pose accuracy thresholded at 5° improves by $3.5\% = 61.4\% - 57.9\%$. A visual result of the paired MIM task is shown in Fig. 3.4.

CFGM and Homography Loss The benefit of the proposed CFGM module and homography loss L_h is included in Tab. 7.5. They help the network predict more accurate results in textureless planar surfaces.

3.6 Conclusion

This work investigates the benefit of pretraining the encoder and decoder of a dense geometric matching network under the paired MIM task. We solve the discrepancy between the pretraining and finetuning tasks. Also, we contribute an improved geometric matching network by reducing the ambiguity of textureless patches and augmenting the learning of local planar surfaces.

Limitation Our method does not produce robust local descriptors. When registering a keypoint, our method needs to run dense matching over all past frames, imposing latency for time-sensitive applications, *e.g.*, odometry estimation.

CHAPTER 4

TAME A WILD CAMERA: IN-THE-WILD MONOCULAR CAMERA CALIBRATION

3D sensing for monocular in-the-wild images, *e.g.*, depth estimation and 3D object detection, has become increasingly important. However, the unknown intrinsic parameter hinders their development and deployment. Previous methods for the monocular camera calibration rely on specific 3D objects or strong geometry prior, such as using a checkerboard or imposing a Manhattan World assumption. This work solves the problem from the other perspective by exploiting the monocular 3D prior. Our method is assumption-free and calibrates the complete 4 Degree-of-Freedom (DoF) intrinsic parameters. First, we demonstrate intrinsic is solved from two well-studied monocular priors, *i.e.*, monocular depthmap, and surface normal map. However, this solution imposes a low-bias and low-variance requirement for depth estimation. Alternatively, we introduce a novel monocular 3D prior, the incidence field, defined as the incidence rays between points in 3D space and pixels in the 2D imaging plane. The incidence field is a pixel-wise parametrization of the intrinsic invariant to image cropping and resizing. With the estimated incidence field, a robust RANSAC algorithm recovers intrinsic. We demonstrate the effectiveness of our method by showing superior performance on synthetic and zero-shot testing datasets. Beyond calibration, we demonstrate downstream applications in image manipulation detection & restoration, uncalibrated two-view pose estimation, and 3D sensing.

4.1 Introduction

Camera calibration is typically the first step in numerous vision and robotics applications [99, 164] that involve 3D sensing. Classic methods enable accurate camera calibration by imaging a specific 3D structure such as a checkerboard [192]. With the rapid growth of monocular 3D vision, there is an increasing focus on 3D sensing for in-the-wild images, such as monocular depth estimation, 3D object detection, and 3D reconstruction. While techniques of 3D sensing over in-the-wild monocular images developed, camera calibration for such in-the-wild images continues to pose significant challenges.

Classic methods for monocular calibration use strong geometry prior, such as using a checker-

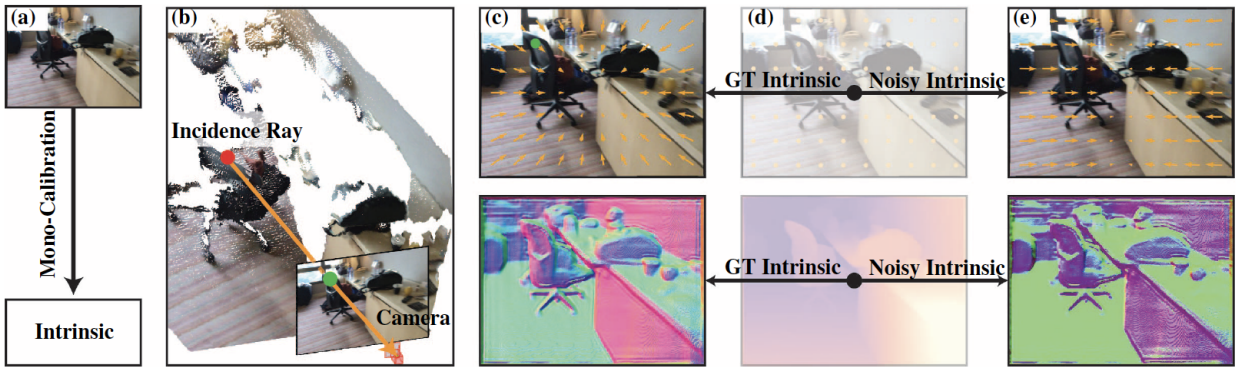


Figure 4.1 In (a), our work focuses on monocular camera calibration for in-the-wild images. We recover the intrinsic from monocular 3D-prior. In (c) - (e), an estimated depthmap is converted to surface normal using a groundtruth and noisy intrinsic individually. Noisy intrinsic distorts the point cloud, consequently leading to inaccurate surface normal. In (e), the normal presents a different color to (d). Motivated by the observation, we develop a solver that utilizes the consistency between the two to recover the intrinsic. However, the solution exhibits numerical instability. We then propose to learn the incidence field as an alternative 3D monocular prior. The incidence field is the collection of the pixel-wise incidence ray, which originates from a 3D point, targets at a 2D pixel, and crosses the camera origin, as shown in (b). Similar to depthmap and normal, a noisy intrinsic leads to a noisy incidence field, as in (e). By same motivation, we develop neural network to learn in-the-wild incidence field and develop a RANSAC algorithm to recover intrinsic from the estimated incidence field.

board. However, such 3D structures are not always available in in-the-wild images. As a solution, alternative methods relax the assumptions. For example, [111] and [91] calibrate using common objects such as human faces and objects' 3D bounding boxes. Another significant line of research [133, 208, 61, 10, 281, 293, 136] is based on the Manhattan World assumption [49], which posits that all planes within a scene are either parallel or perpendicular to each other. This assumption is further relaxed [284, 109, 121] to estimate the lines that are either parallel or perpendicular to the direction of gravity. The intrinsic parameters are recovered by determining the intersected vanishing points of detected lines, assuming a central focal point and an identical focal length.

While the assumptions are relaxed, they may still not hold true for in-the-wild images. This creates a contradiction: although we enable robust models to estimate in-the-wild monocular depthmap, generating its 3D point cloud remains infeasible due to the missing intrinsic. A similar challenge arises in monocular 3D object detection, as we face limitations in projecting the detected 3D bounding boxes onto the 2D image. In AR/VR applications, the absence of intrinsic precludes

placing multiple reconstructed 3D objects within a canonical 3D space. The absence of a reliable, assumption-free monocular intrinsic calibrator has become a bottleneck in deploying these 3D sensing applications.

Our method is motivated by the consistency between the monocular depthmap and surface normal map. In Fig. 7.1 (c) - (e), an incorrect intrinsic distorts the back-projected 3D point cloud from the depthmap, resulting in distorted surface normals. Based on this, intrinsic is optimal when the estimated monocular depthmap aligns consistently with the surface normal. We present a solution to recover the complete 4 DoF intrinsic by leveraging the consistency between the surface normal and depthmap. However, the algorithm is numerically ill-conditioned as its computation depends on the accurate gradient of depthmap. This requires depthmap estimation with low bias and variance.

To resolve it, we propose an alternative approach by introducing an additional novel 3D monocular prior in complementation to the depthmap and surface normal map. We refer to this as the incidence field, which depicts the incidence ray between the observed 3D point and the projected 2D pixel on the imaging plane, as shown in Fig. 7.1 (b). The combination of the incidence field and the monocular depthmap describes a 3D point cloud. Compared to the original solution, the incidence field is a direct pixel-wise parameterization of the camera intrinsic. This implies that a minimal solver based on the incidence field only needs to have low bias. We then utilize a deep neural network to perform the incidence field estimation. A non-learning RANSAC algorithm is developed to recover the intrinsic parameters from the estimated incidence field.

We consider the incidence field a monocular 3D prior. Similar to depthmap and surface normal, the incidence field is invariant to the image cropping or resizing. This encourages its generalization over in-the-wild images. To empirically support our argument, we collected multiple public datasets into a comprehensive dataset with diverse indoor, outdoor, and object-centric images captured by different imaging devices. We further boost the variety of intrinsic by resizing and cropping the images in a similar manner as [91]. Finally, we include zero-shot testing samples to benchmark real-world monocular camera calibration performance.

	[317, 318, 138, 316, 313, 111, 91]	[133, 208, 281, 136]	[109, 138]	[121]	Ours
DoF	4	1	1	3	4
Assumption	Specific-Objects	Manhattan	Manhattan-Train	Manhattan-Train	None
Train Data	-	-	Panorama Image	Panorama Image	Calibrated Image

Table 4.1 **Camera Calibration Methods from Strong to Relaxed Assumptions.** Non-learning methods [317, 318, 138, 316, 313, 111, 91, 133, 208, 281, 136] rely on strong assumptions. Learning based methods [109, 138, 121] relax the assumptions to training data. Our method makes no assumptions in either training or testing. This enables training with any calibrated images while [109, 138] consume panorama images. Despite that, we calibrate complete 4 DoF intrinsic.

We showcase downstream applications that benefit from monocular camera calibration. Despite the aforementioned 3D sensing tasks, we present two intriguing additional applications. One is detecting and restoring image resizing and cropping. When an image is cropped or resized, it disrupts the assumption of a central focal point and identical focal length, leading to irregular intrinsic. Using the estimated intrinsic parameters, we restore the edited image by adjusting its intrinsic to a regularized form. The other application involves two-view uncalibrated camera pose estimation. With established image correspondence, a fundamental matrix [100] is determined. However, there does not exist an injective mapping between the fundamental matrix and camera pose [99]. This raises a counter-intuitive fact: inferring the pose from two uncalibrated images is infeasible. But our method enables uncalibrated two-view pose estimation via applying monocular camera calibration.

We summarize our contributions as follows:

◇ Our approach tackles monocular camera calibration from a novel perspective by relying on monocular 3D priors. Our method makes no assumption for the to-be-calibrated image.

◇ Our algorithm provides robust monocular intrinsic estimation for in-the-wild images, accompanied by extensive benchmarking and comparisons against other baselines.

◇ We demonstrate its benefits on additional intriguing diverse and novel downstream applications.

4.2 Related Works

Monocular Camera Calibration with Geometry. One line of work [133, 208, 61, 10, 281, 293, 136] assumes the Manhattan World assumption [49], where all planes in 3D space are either parallel or perpendicular. Under the assumption, line segments in the image converge at the

vanishing points, from which the intrinsic is recovered. LSD [262] and EDLine [3] develop robust line estimators. Others jointly estimate the horizon line and the vanishing points [311, 224, 141]. In Tab. 4.1, recent learning-based methods [283, 284, 109, 138, 121] relax the assumption to training data. They train the model using panorama images whose vanishing point and horizon lines are known. Still, the assumption constrains [283, 284, 109, 138] in modeling intrinsic as 1 DoF camera. Recently, [121] relaxes the assumption to 3 DoF via regressing the focal point. In comparison, our method makes no assumption. This enables us to calibrate 4 DoF intrinsic and train with any calibrated images.

Monocular Camera Calibration with Object. Zhang's method [317] based on a checkerboard pattern is widely regarded as the standard for camera calibration. Several works generalize this method to other geometric patterns such as 1D objects [318], line segments [316], and spheres [313]. Recent works [111] and [91] extend camera calibration to real-world objects such as human faces. Optimizers, including BPnP [37] and PnP [233] are developed. However, the usage of specific objects restricts their applications. In contrast, our approach applies to any image.

Image Cropping and Resizing. Detecting content-based image manipulation [155] is extensively researched. But few studies geometric manipulation, such as resizing and cropping. On resizing, [56] regresses the image aspect ratio with a deep model. On cropping, a recent proactive method [306] is developed. We demonstrate image calibration also addresses image geometric manipulation. Our method does not need to encrypt images, complementing content-based manipulation detection.

Uncalibrated Two-View Pose Estimation. With the fundamental matrix estimated, the two-view camera pose is determined up to a projective ambiguity if images are uncalibrated. Alternative solutions [235, 125, 169] exist by employing deep networks to regress the pose. However, regression hinders the usage of geometric constraints, which proves crucial in calibrated two-view pose estimation [322, 240, 218]. Other work [101, 76] use more than two uncalibrated images for pose estimation. Our work complements prior studies by enabling a minimum uncalibrated two-view solution.

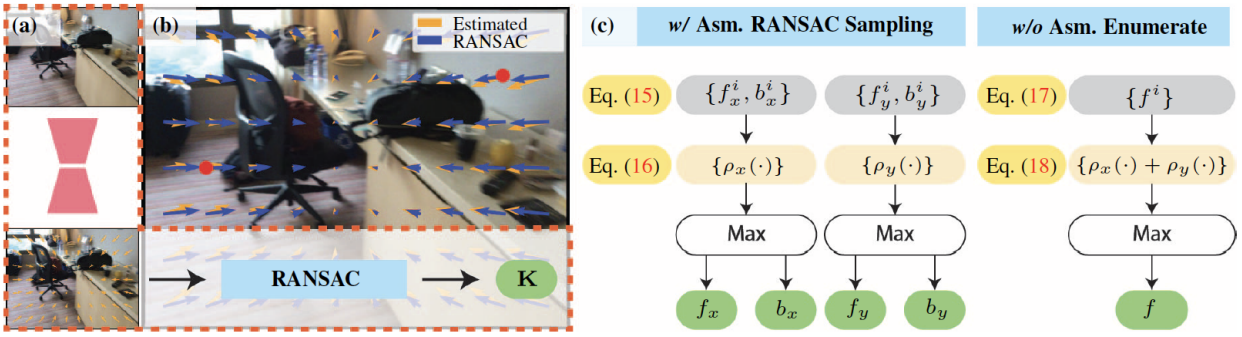


Figure 4.2 We illustrate the framework for the proposed monocular camera calibration algorithm. In (a), a deep network maps the input image $\mathbf{I}$ to the incidence field $\mathbf{V}$. A RANSAC algorithm recovers intrinsic from $\mathbf{V}$. In (b), we visualize a single iteration of RANSAC. An intrinsic is computed with two incidence vectors randomly sampled at red pixel locations. From Eq. (4.2), an intrinsic determines the incidence vector at a given location. The optimal intrinsic maximizes the consistency with the network prediction (blue and orange). Subfigure (c) details the RANSAC algorithm. Different strategies are applied depending on if a simple camera is assumed. If not assumed, we independently compute (f_x, b_x) and (f_y, b_y). If assumed, there is only 1 DoF of intrinsic. We proceed by enumerating the focal length within a predefined range to determine the optimal value.

4.3 Method

In this section, we first show how to estimate intrinsic parameters by using monocular 3D priors, such as the surface normal map and monocular depthmap. We then introduce the incidence field as a new monocular 3D prior, which complements the surface normal map and monocular depthmap. We describe the training strategy and the network used to learn the incidence field. After estimating the incidence field, we present a RANSAC algorithm to recover the 4 DoF intrinsic parameters. Lastly, we explore various feasible downstream applications of the proposed algorithms. As this work focuses on studying intrinsic parameters in monocular images captured by modern imaging devices, we ignore the estimation of skew, radial, or tangential distortion. Fig. 6.3 shows algorithm framework.

4.3.1 Intrinsic Calibration from Monocular 3D Priors

Our method aims to use generalizable monocular 3D priors without assuming the 3D scene geometry. Hence, we start with monocular depthmap $\mathbf{D}$ and surface normal map $\mathbf{N}$. Assume there exists a learnable mapping between the input image $\mathbf{I}$, depthmap $\mathbf{D}$, and normal map $\mathbf{N}$: $\mathbf{D}, \mathbf{N} = \mathbb{D}_\theta(\mathbf{I})$, where $\mathbb{D}_\theta$ can be a learned network. We denote the intrinsic $\mathbf{K}_{\text{simple}}$, $\mathbf{K}$, and its

inverse $\mathbf{K}^{-1}$ as:

$$\mathbf{K}_{\text{simple}} = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} f_x & 0 & b_x \\ 0 & f_y & b_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}^{-1} = \begin{bmatrix} 1/f_x & 0 & -b_x/f_x \\ 0 & 1/f_y & -b_y/f_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.1}$$

The notation $\mathbf{K}_{\text{simple}}$ suggests a simple camera model with the identical focal length and central focal point assumption. Given a 2D homogeneous pixel location $\mathbf{p}^\intercal = \begin{bmatrix} x & y & 1 \end{bmatrix}$ and its depth value $d = \mathbf{D}(\mathbf{p})$, the corresponding 3D point is defined as:

$$\mathbf{P} = d \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = d \cdot \mathbf{K}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = d \cdot \begin{bmatrix} \frac{x-b_x}{f_x} \\ \frac{y-b_y}{f_y} \\ 1 \end{bmatrix} = d \cdot \mathbf{v}, \tag{4.2}$$

where the vector $\mathbf{v}$ is an incidence ray, originating from the 3D point $\mathbf{P}$, directed towards the 2D pixel $\mathbf{p}$, and passing through the camera's origin. The incidence field is determined by the collection of incidence rays associated with each pixel, where $\mathbf{v} = \mathbf{V}(\mathbf{p})$.

4.3.2 Intrinsic from Monocular 3D Prior Constraints

In this section, we explain how to determine the intrinsic matrix $\mathbf{K}$ using the estimated surface normal map $\mathbf{N}$ and depthmap $\mathbf{D}$. Given the estimated depth $d = \mathbf{D}(\mathbf{p})$ and normal $\mathbf{n} = \mathbf{N}(\mathbf{p})$ at 2D pixel location $\mathbf{p}$, a local 3D plane is described as:

$$\mathbf{n}^\intercal \cdot d \cdot \mathbf{v} + c = 0. \tag{4.3}$$

By taking derivative in x-axis and y-axis directions, we have:

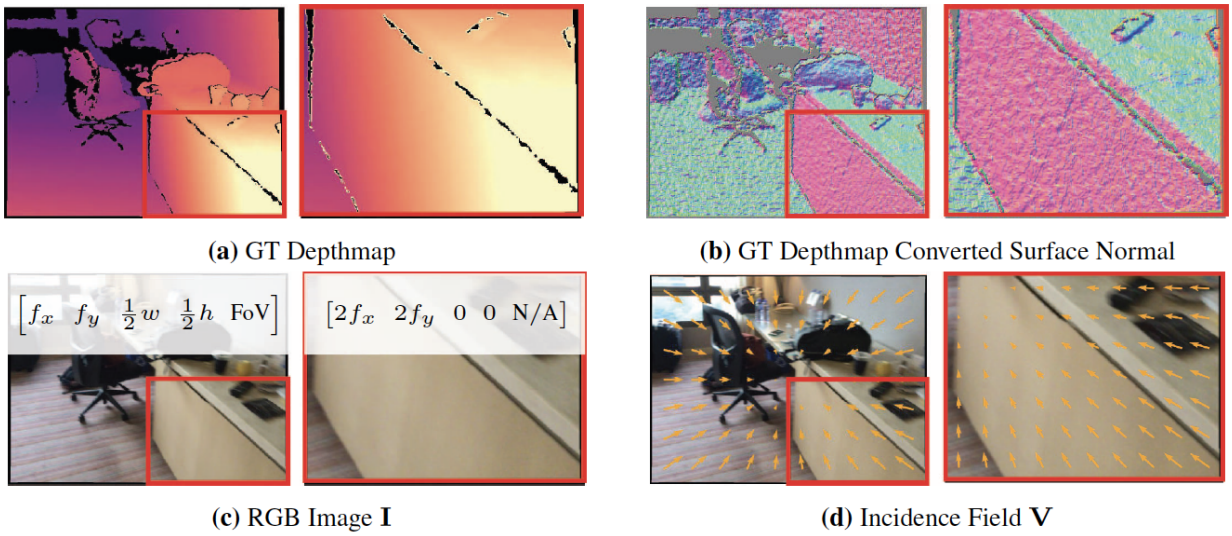
$$\mathbf{n}^\intercal \nabla_x (d \cdot \mathbf{v}) = 0, \quad \mathbf{n}^\intercal \nabla_y (d \cdot \mathbf{v}) = 0. \tag{4.4}$$

Note the bias b of the 3D local plane is independent of the camera projection process. Without loss of generality, we show the case of our method for x-direction. Expanding Eq. (4.4), we obtain:

$$n_1 \nabla_x (d \cdot \frac{x - b_x}{f_x}) + n_2 \frac{y - b_y}{f_y} \nabla_x (d) + n_3 \nabla_x (d) = 0, \tag{4.5}$$

where $\nabla_x(d)$ represents the gradient of the depthmap $\mathbf{D}$ in the x-axis and can be computed, for example, using a Sobel filter [129]. Next, re-parametrize the unknowns in Eq. (4.5) to get:

$$a_1 f_x f_y + a_2 f_x b_y + a_3 f_y b_x + a_4 f_y + a_5 f_x = 0. \tag{4.6}$$



(a) GT Depthmap **(b)** GT Depthmap Converted Surface Normal

(c) RGB Image $\mathbf{I}$ **(d)** Incidence Field $\mathbf{V}$

Figure 4.3 In (a) and (b), we highlight the ground truth depthmap of a smooth surface, such as a table's side. Even with the ground truth depthmap, the resulting surface normals exhibit noise patterns due to the inherent high variance. This makes the intrinsic solver based on the consistency of the depthmap and surface normals numerically unstable. Further, (a)-(d) demonstrate a scaling and cropping operation applied to each modality. In (c), the intrinsic changes per operation, leading to ambiguity if a network directly regresses the intrinsic values. Meanwhile, the FoV is undefined after cropping. In comparison, the incidence field remains invariant to image editing, same as the surface normal and depthmap.

Divide both sides of the equation by f_x to get:

$$a_1 f_y + a_2 b_y + a_3 r b_x + a_4 r + a_5 = 0, \tag{4.7}$$

where $r = \frac{f_y}{f_x}$. By stacking Eq. (4.7) with $N \geq 4$ randomly sampled pixels, we acquire a linear system:

$$\mathbf{A}_{N\times 4}\ \mathbf{X}_{4\times 1} = \mathbf{B}_{4\times 1}, \tag{4.8}$$

where the intrinsic parameter to be solved is stored in a vector $\mathbf{X}_{4\times 1}^{\intercal} = \begin{bmatrix} f_y & b_y & r b_x & r \end{bmatrix}^{\intercal}$. This solves the other intrinsic parameters as:

$$f_y = f_y,\ b_y = b_y,\ f_x = \frac{f_y}{r},\ b_x = \frac{r b_x}{r}. \tag{4.9}$$

The known constants are stored in matrix $\mathbf{A}_{N\times 4}$ and $\mathbf{B}_{4\times 1}$. If we choose $N = 4$ in Eq. (4.8), we obtain a minimal solver where the solution $\mathbf{X}$ is computed by performing Gauss-Jordan Elimination. Conversely, when $N > 4$, the linear system is over-determined, and $\mathbf{X}$ is obtained using a least squares solver. The above suggests the intrinsic is recoverable from the monocular 3D prior.

4.3.3 Incidence Field as Monocular 3D Prior

Eq. (4.9) relies on the consistency between the surface normal and depthmap gradient, which may require a low-variance depthmap estimate. From Fig. 4.3, even groundtruth depthmap leads to spurious normal due to its inherent high variance. Minimal solver in Eq. (4.9) can lead to a poor solution.

As a solution, we propose to directly learn the incidence field $\mathbf{V}$ as a monocular 3D prior. In Eq. (4.2) and Fig. 7.1, the combination of the incidence field $\mathbf{V}$ and the monocular depthmap $\mathbf{D}$ creates a 3D point cloud. In Eq. (4.3), the incidence field $\mathbf{V}$ can measure the observation angle between a 3D plane and the camera. Similar to depthmap $\mathbf{D}$ and surface normal map $\mathbf{N}$, the incidence field $\mathbf{V}$ is invariant to the image cropping and resizing. Consider an image cropping and resizing described as:

$$\mathbf{x}' = \Delta\mathbf{K}\ \mathbf{x}, \quad \Delta\mathbf{K} = \begin{bmatrix} \Delta f_x & 0 & \Delta c_x \\ 0 & \Delta f_y & \Delta c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}' = \Delta\mathbf{K}\mathbf{K}, \tag{4.10}$$

where $\mathbf{K}'$ is the intrinsic after transformation. The surface normal map $\mathbf{N}$ and depthmap $\mathbf{D}$ after transformation is defined as:

$$\mathbf{N}'(\mathbf{x}') = \mathbf{N}(\mathbf{x}) = \mathbf{N}(\Delta\mathbf{K}^{-1}\mathbf{x}'), \quad \mathbf{D}'(\mathbf{x}') = \mathbf{D}(\mathbf{x}) = \mathbf{D}(\Delta\mathbf{K}^{-1}\mathbf{x}'). \tag{4.11}$$

Similarly, the incidence field after transformation is:

$$\mathbf{V}'(\mathbf{x}') = (\mathbf{K}')^{-1}\mathbf{x}' = \mathbf{K}^{-1}(\Delta\mathbf{K})^{-1}\ \mathbf{x}' = \mathbf{K}^{-1}\ \mathbf{x} = \mathbf{v} = \mathbf{V}(\mathbf{x}). \tag{4.12}$$

Eq. (4.12) suggests that the incidence field $\mathbf{V}$ is a parameterization of the intrinsic matrix that is **invariant** to image resizing and image cropping. Other invariant parameterizations of the intrinsic matrix, such as the camera field of view (FoV), rely on the central focal point assumption and only cover a 2 DoF intrinsic matrix. An illustration is put in Fig. 4.3.

4.3.4 Learn Monocular Incidence Field

Given the strong connection between the monocular depthmap $\mathbf{D}$ and camera incidence field $\mathbf{V}$, we adopt NewCRFs [309], a neural network used for monocular depth estimation, for incidence

field estimation. We change the last output head to output a three-dimensional normalized incidence field $\widetilde{\mathbf{V}}$ with the same resolution as the input image $\mathbf{I}$. We adopt a cosine similarity loss defined as:

$$\widetilde{\mathbf{V}} = \mathbb{D}_\theta(\mathbf{I}), \quad L = \frac{1}{N}\sum_{i=1}^{N} \widetilde{\mathbf{V}}^\intercal(\mathbf{x}_i)\widetilde{\mathbf{V}}_{\text{gt}}(\mathbf{x}_i). \tag{4.13}$$

We normalize the last dimension of the incidence field to one before feeding to the RANSAC algorithm. That is to say, $\mathbf{V}^\intercal(\mathbf{x}_i) = \begin{bmatrix}\tilde{v}_1/\tilde{v}_3 & \tilde{v}_2/\tilde{v}_3 & 1\end{bmatrix}^\intercal = \begin{bmatrix}v_1 & v_2 & 1\end{bmatrix}^\intercal$.

4.3.5 Intrinsic from Monocular Incidence Field

Since the network inference executes on GPU device, we adopt a GPU-end RANSAC algorithm to recover the intrinsic $\mathbf{K}$ from the incidence field $\mathbf{V}$. Unlike a CPU-based RANSAC, we perform fixed K_r iterations of RANSAC without termination. In RANSAC, we use the minimal solver to generate K_c candidates and select the optimal one that maximizes a scoring function (see Fig. 6.3).

RANSAC *w.o* Assumption. From Eq. (4.2), the incidence vector $\mathbf{v}$ relates to the intrinsic $\mathbf{K}$ as:

$$\mathbf{v}^\intercal = \mathbf{K}^{-1}\mathbf{x} = \begin{bmatrix}\frac{x-b_x}{f_x} & \frac{y-b_y}{f_y} & 1\end{bmatrix}^\intercal. \tag{4.14}$$

From Eq. (4.14), a minimal solver for intrinsic is straightforward. In the incidence field, randomly sample two incidence vectors $(\mathbf{v}^1)^\intercal = \begin{bmatrix}v_x^1 & v_y^1 & 1\end{bmatrix}^\intercal$ and $(\mathbf{v}^2)^\intercal = \begin{bmatrix}v_x^2 & v_y^2 & 1\end{bmatrix}^\intercal$. The intrinsic is:

$$\begin{cases} f_x = \frac{x^1 - x^2}{v_x^1 - v_x^2} \\ b_x = \frac{1}{2}(x^1 - v_x^1 f_x + x^2 - v_x^2 f_x) \end{cases}, \quad \begin{cases} f_y = \frac{y^1 - y^2}{v_y^1 - v_y^2} \\ b_y = \frac{1}{2}(x^1 - v_y^1 f_y + x^2 - v_y^2 f_y) \end{cases}. \tag{4.15}$$

Similarly, the scoring function is defined in x-axis and y-axis, respectively:

$$\rho_x(f_x, b_x, \{\mathbf{x}\}, \{\mathbf{v}\}) = \sum_{i=1}^{N_k}\left(\|\frac{x^i - b_x}{f_x} - v_x^i\| < k_x\right), \quad \rho_y(f_y, b_y, \{\mathbf{x}\}, \{\mathbf{v}\}) = \sum_{i=1}^{N_k}\left(\|\frac{y^i - b_y}{f_y} - v_y^i\| < k_y\right). \tag{4.16}$$

RANSAC *w/* Assumption. If a simple camera model is assumed, *i.e.*, intrinsic only has an unknown focal length, it only needs to estimate 1-DoF intrinsic. We enumerate the focal length candidates as:

$$\{f\} = \{f_{\min} + \frac{i}{N_f}(f_{\max} - f_{\min}) \mid 0 \le i \le N_f\}. \tag{4.17}$$

Dataset	Calibration	Scene	ZS	Syn.	Perspective [121]		Ours		Ours + Asm.	
					e_f	e_b	e_f	e_b	e_f	e_b
NuScenes [33]	Calibrated	Driving	✗	✔	0.610	0.248	**0.102**	**0.087**	0.402	0.400
KITTI [83]	Calibrated	Driving	✗	✔	0.670	0.221	**0.111**	**0.078**	0.383	0.368
Cityscapes [48]	Calibrated	Driving	✗	✔	0.713	0.334	**0.108**	**0.110**	0.387	0.367
NYUv2 [222]	Calibrated	Indoor	✗	✔	0.449	0.409	**0.086**	**0.174**	0.376	0.379
ARKitScenes [13]	Calibrated	Indoor	✗	✔	0.362	0.410	**0.140**	**0.243**	0.400	0.377
SUN3D [289]	Calibrated	Indoor	✗	✔	0.442	0.501	**0.113**	**0.205**	0.389	0.383
MVImgNet [307]	SfM	Object	✗	✔	0.204	0.500	**0.101**	**0.081**	0.108	0.072
Objectron [2]	Label	Object	✗	✔	0.178	0.339	**0.078**	**0.070**	0.088	0.079
MegaDepth [146]	SfM	Outdoor	✗	✗	0.493	**0.000**	0.137	0.046	**0.109**	**0.000**
Waymo [235]	Calibrated	Driving	✔	✗	0.564	**0.020**	0.210	0.053	**0.157**	**0.020**
RGBD [232]	Pre-defined	Indoor	✔	✗	0.264	**0.000**	0.097	0.039	**0.067**	**0.000**
ScanNet [52]	Calibrated	Indoor	✔	✗	0.385	**0.010**	0.128	0.041	**0.109**	**0.010**
MVS [80]	Pre-defined	Hybrid	✔	✗	0.312	**0.000**	0.170	0.028	**0.127**	**0.000**
Scenes11 [36]	Pre-defined	Synthetic	✔	✗	0.348	**0.000**	0.170	0.044	**0.117**	**0.000**

Table 4.2 **In-the-Wild Monocular Camera Calibration.** We benchmark in-the-wild monocular camera calibration performance. On the training dataset except MegaDepth, we synthesize novel intrinsic by cropping and resizing. Note the synthesized images violate the focal point and focal length assumption. [**Key:** ZS = Zero-Shot, Asm. = Assumptions, Syn. = Synthesized]

The scoring function under the scenario is defined as the summation over x-axis and y-axis:

$$\rho(f, \{\mathbf{x}\}, \{\mathbf{v}\}) = \rho_x(f_x, w/2, \{\mathbf{x}\}, \{\mathbf{v}\}) + \rho_y(f_y, h/2, \{\mathbf{x}\}, \{\mathbf{v}\}). \tag{4.18}$$

4.3.6 Downstream Applications

Image Crop & Resize Detection and Restoration. Eq. (4.10) defines a crop and resize operation:

$$\mathbf{x}' = \Delta\mathbf{K}\,\mathbf{x}, \quad \mathbf{K}' = \Delta\mathbf{K}\,\mathbf{K}, \quad \mathbf{I}'(\mathbf{x}') = \mathbf{I}'(\Delta\mathbf{K}\,\mathbf{x}) = \mathbf{I}(\mathbf{x}). \tag{4.19}$$

When a modified image $\mathbf{I}'$ is presented, our algorithm calibrates its intrinsic $\mathbf{K}'$ and then:

Case 1: The original intrinsic $\mathbf{K}$ is known. *E.g.*, determine $\mathbf{K}$ with the camera type through the image-associated EXIF file [4]. Image manipulation is computed as $\Delta\mathbf{K} = \mathbf{K}'\,\mathbf{K}^{-1}$. A manipulation is detected if $\Delta\mathbf{K}$ deviates from an identity matrix. The original image restores as $\mathbf{I}(\mathbf{x}) = \mathbf{I}'(\Delta\mathbf{K}\,\mathbf{x})$. Interestingly, the four corners of image $\mathbf{I}'$ are mapped to a bounding box in original image $\mathbf{I}$ under manipulation $\Delta\mathbf{K}$. We thus quantify the restoration by measuring the bounding box. See Fig. 6.7.

Case 2: The original intrinsic $\mathbf{K}$ is unknown. We assume the genuine image possess an identical focal length and central focal point. Any resizing and cropping are detected when matrix $\mathbf{K}'$ breaks this assumption. Note, the rule can not detect aspect ratio preserving resize and centered crop. We restore the original image by defining an inverse operation $\Delta\mathbf{K}$ restore $\mathbf{K}'$ to an intrinsic fits the assumption.

FoV (°)	Upright [136] PAMI'13	erceptual [109] CVPR'18	CTRL-C [138] ICCV'21	Perspective [121] CVPR'23	Ours *w/o* Asm.	Ours *w/* Asm.
Mean	9.47	4.37	3.59	3.07	2.49	**2.47**
Median	4.42	3.58	2.72	2.33	1.96	**1.92**

Table 4.3 **Comparisons to Monocular Camera Calibration with Geometry** on GSV dataset [5]. We follow the training and testing protocol of [138]. For a fair comparison, we convert the estimated intrinsic to camera FoV on the y-axis direction, following [138, 121], and report our results *w/* and *w/o* the assumptions.

3D Sensing Related Tasks. With intrinsic estimated, multiple applications become available for in-the-wild images. *E.g.*, depthmap to point cloud, uncalibrated two-view pose estimation, and etc.

4.4 Experiments

4.4.1 Monocular Camera Calibration In-The-Wild

Datasets. Our method is trained whenever a calibrated intrinsic is provided, making it applicable to a wide range of publicly available datasets. In Tab. 4.2, we incorporate datasets of different application scenarios, including indoor, outdoor scenes, driving, and object-centric scenes. Dataset MVS [80] is a hybrid dataset involved with indoor, outdoor, and object-centric images. Many of the datasets utilize only a single type of camera for data collection, resulting in a scarcity of intrinsic variations. Similar to [138], we employ random resizing and cropping to synthesize more intrinsic, marked in Tab. 4.2 column "Syn.". In augmentation, we first resize all images to a resolution of 480×640. We then uniformly random resize up to two times its size and subsequently crop to a resolution of 480×640. As MegaDepth [146] collects images captured by various cameras from the Internet, we disable its augmentation. We document the intrinsic parameters of each dataset in Supp.

In Tab. 4.2 column "Calibration", we assess intrinsic quality into various levels. "Calibrated" suggests accurate calibration with a checkerboard. "Pre-defined" is less accurate, indicating the default intrinsic provided by the camera manufacturer without a calibration process. "SfM" signifies that the intrinsic is computed via an SfM method [209]. "Labeled" means the intrinsic manually labeled by a human.

In-The-Wild Monocular Camera Calibration. We benchmark in-the-wild monocular calibration performance on Tab. 4.2. For trained datasets, except for MegaDepth, we test on synthetic data

Methods	BIWIRGBD-ID [188]						CAD-120 [236]					
	e_f	e_{f_x}	e_{f_y}	e_b	e_{b_x}	e_{b_y}	e_f	e_{f_x}	e_{f_y}	e_b	e_{b_x}	e_{b_y}
Louraki [158]	0.662	0.662	0.662	-	0.387	0.222	0.732	0.732	0.732	-	0.255	0.180
Fetzer [76] WACV'20	0.845	0.845	0.845	-	0.001	0.005	0.679	0.679	0.679	-	0.001	0.005
BPnP [37] CVPR'19	0.675	0.675	0.675	-	0.322	0.479	1.178	1.178	1.178	-	0.103	0.129
FaceCalib [111] FG'23	0.133	0.133	0.133	-	0.026	0.042	0.151	0.151	0.151	-	0.023	0.063
Ours	0.034	0.029	**0.016**	0.020	0.011	0.018	0.137	0.137	0.054	0.042	0.042	0.008
Ours + Assumptions	**0.019**	**0.019**	0.019	**0.000**	**0.000**	**0.000**	**0.047**	**0.047**	**0.047**	**0.000**	**0.000**	**0.000**

Table 4.4 **Comparisons to Monocular Camera Calibration with Object.** We compare to the recent FaceCalib [111], which calibrates the camera using video containing human faces. We report our results *w/* and *w/o* assuming a simple camera model. We perform **zero-shot** prediction without training using Tab. 4.2 model.

using random cropping and resizing. For the unseen test dataset, we refrain from applying any augmentation to better mimic real-world application scenarios.

We compare to the recent baseline [121], which regresses intrinsic via a deep network. Note, [121] can not train on arbitrary calibrated images as requiring panorama images in training. A fair comparison using the same training and testing images is in Tab. 4.4 and Sec. 4.4.2. [121] provides models with two variations: one assumes a central focal point, and another does not. We report with the former model whenever the input image fits the assumption. From Tab. 4.2, our method demonstrates superior generalization across multiple unseen datasets. Further, the result *w/* assumption outperforms *w/o* assumption whenever the input images fit the assumption. Tested on an RTX-2080 Ti GPU, the combined network inference and calibration algorithm runs on average in 87 ms.

4.4.2 Monocular Camera Calibration with Geometry

Methods in this line of research hold a Manhattan World assumption, positing that images consist of planes that are either parallel or perpendicular to each other. Stated in Tab. 4.1, baselines [109, 138, 121] relax the assumption to training data. Our method imposes no assumption in both training and testing.

This brings three benefits. First, the assumption restricts their training to panorama images. In contrast, our model is trainable with any calibrated images. This yields improved generalization, as shown in Tab. 4.2. Second, it constrains the baselines to a simple camera parameterized by FoV. We consider the proposed incidence field a more generalizable and invariant parameterization

(a) Input & Incidence (b) Est. Restoration (c) GT. Restoration (d) Original Image

Figure 4.4 **Image Crop & Resize Detection and Restoration.** Image editing, including cropping and resizing changes intrinsic. As in Sec. 4.3.6, monocular calibration is applicable to detect and restore image manipulations. We visualize the zero-shot samples on ScanNet and Waymo. More examples are in Supp.

of intrinsic. *E.g.*, while FoV remains invariant to image resizing, it still changes after cropping. However, the incidence field is unaffected in both cases. In Tab. 4.4, the substantial improvement we achieved ($0.60 = 3.07 - 2.47$) over the recent SoTA [121] empirically supports our argument. Third, our method calibrates the 4 DoF intrinsic with a non-learning RANSAC algorithm. Baselines instead regress the intrinsic. This renders our method more robust and interpretable. In Fig. 6.3 (b), the estimated intrinsic quality is visually discerned through the consistency achieved between the two incidence fields.

4.4.3 Monocular Camera Calibration with Objects

We compare to the recent object-based camera calibration method FaceCalib [111]. The baseline employs a face alignment model to calibrate the intrinsic over a video. Both [111] and ours perform zero-shot prediction. We report performance using Tab. 4.2 model. Compared to [111], our method is more general as it does not assume a human face present in the image. Meanwhile, [111] calibrates over a **video**, while ours is a **monocular** method. For a fair comparison, we report the video-based results as an averaged error over the videos. We report results *w/* and *w/o* assuming a simple camera model. Since the tested image has a central focal point, when the assumption applied, the error of the focal point diminished. In Tab. 4.4, we outperform SoTA substantially. The error metrics are in Supp.

Methods	KITTI [83]		NYUv2 [222]		ARKitScenes [13]		Waymo [235]		RGBD [232]		ScanNet [52]		MVS [80]	
	mIOU	Acc	mIOU	Acc	mIOU	Acc	mIOU	Acc	mIOU	Acc	mIOU	Acc	mIOU	Acc
Baseline	0.686	0.795	0.621	0.710	0.586	0.519	0.581	0.721	0.636	0.681	0.597	0.811	0.595	0.667
Ours	**0.842**	**0.852**	**0.779**	**0.856**	**0.691**	**0.837**	**0.681**	**0.796**	**0.693**	**0.781**	**0.709**	**0.887**	**0.638**	**0.795**

Table 4.5 **Image Crop and Resize Restoration.** Stated in Sec. 4.3.6, our method also encompasses the restoration of image manipulations. Use model reported in Tab. 4.2, we conduct evaluations on both seen and unseen datasets.

4.4.4 Downstream Applications

Image Crop & Resize Detection and Restoration. Content-based image manipulation detection and restoration [16, 154] is extensively studied. However, few explore geometric manipulation, including resizing and cropping. In Sec. 4.3.6, our method also addresses the detection and restoration of geometric manipulations in images. Using the model reported in Tab. 4.2, we benchmark its performance in Tab. 4.5. Random manipulations following Sec. 4.4.1 contribute to 50% of both train and test sets, and the other 50% are genuine images. In Tab. 4.5, we evaluate restoration with mIOU and report detection accuracy (*i.e.* binary classification of genuine vs edited images). From the table, our method generalizes to the unseen dataset, achieving an averaged mIOU of 0.680. Meanwhile, we substantially outperform the baseline, which directly regresses the intrinsic. The ablation suggests the benefit of the incidence field as an invariant intrinsic parameterization. Beyond performance, our algorithm is interpretable. In Fig. 6.7, the perceived image geometry is interpretable for humans.

Uncalibrated Two-View Camera Pose Estimation. With correspondence between two images, one can infer the fundamental matrix [100]. However, the pose between two uncalibrated images is determined by a projective ambiguity. Our method eliminates the ambiguity with monocular camera calibration. In Tab. 4.6, we benchmark the uncalibrated two-view pose estimation and compare it to recent baselines. The result is reported using Tab. 4.2 model by assuming unique intrinsic for **both** images. We perform zero-shot in ScanNet. For MegaDepth, it includes images collected over the Internet with diverse intrinsics. Interestingly, in ScanNet, our uncalibrated method outperforms a calibrated one [152]. In Supp, we plot the curve between pose performance and intrinsic quality. The challenging setting suggests itself an ideal task to evaluate the intrinsic quality.

Methods	Calibrated	ScanNet [52]			MegaDepth [146]		
		@5°	@10°	@20°	@5°	@10°	@20°
SuperGlue [204] CVPR'19	✔	16.2	33.8	51.8	42.2	61.2	75.9
DRC-Net [152] ICASSP'22	✔	7.7	17.9	30.5	27.0	42.9	58.3
LoFTR [234] CVPR'21	✔	22.0	40.8	57.6	52.8	69.2	81.2
ASpanFormer [38] ECCV'22	✔	25.6	46.0	63.3	55.3	71.5	83.1
PMatch [331] CVPR'23	✔	29.4	50.1	67.4	61.4	75.7	85.7
PMatch [331] CVPR'23	✗	11.4	29.8	49.4	16.8	30.6	47.4

Table 4.6 **Uncalibrated Two-View Camera Pose Estimation.** We use the model reported in Tab. 4.2 and assume distinct camera models for **both** frames. During calibration, we apply the simple camera assumption. The last two rows ablate the performance using GT intrinsic and our estimated intrinsic.

4.5 Conclusion

We calibrate monocular images through a novel monocular 3D prior referred as incidence field. The incidence field is a pixel-wise parameterization of intrinsic invariant to image resizing and cropping. A RANSAC algorithm is developed to recover intrinsic from the incidence field. We extensively benchmark our algorithm and demonstrate robust in-the-wild performance. Beyond calibration, we show multiple downstream applications that benefit from our method.

Limitation. In real application, whether to apply the assumption still waits human input.

Broader Impacts. We do not anticipate any potential negative social impact arising from this work.

CHAPTER 5

LIGHTEDDEPTH: VIDEO DEPTH ESTIMATION IN LIGHT OF LIMITED INFERENCE VIEW ANGLES

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.

5.1 Introduction

Depth estimation is a fundamental task for applications such as 3D reconstruction [31], robotics [132], and autonomous driving [310]. The depth is self-contained in the scene motion brought by the camera movement. The classic SfM methods [157, 209, 196, 287, 85] 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. 7.1 (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 [280, 240, 258, 268, 291] formulate their solutions as deep SfM, shown in Fig. 7.1 (b). They can be grouped into two types [268]. Type **I** methods [280, 240, 258] execute SfM within a fixed frame window, embedding bundle-adjustment as a differentiable module within a network. Type **II** methods [268, 291] 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 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 [240] and RAFT [241]. 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 [322] in using the five-point algorithm [143] with RANSAC [77] 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 [280, 240, 258] 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

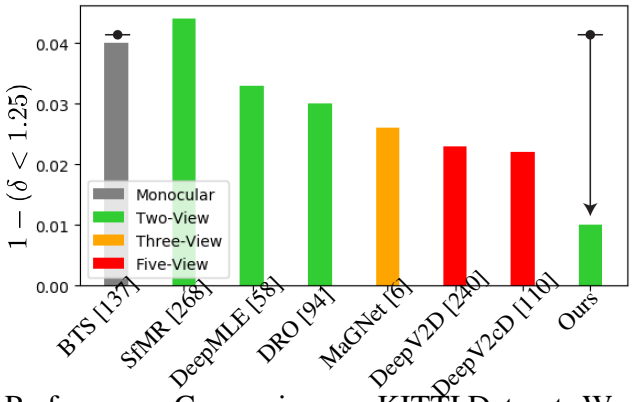


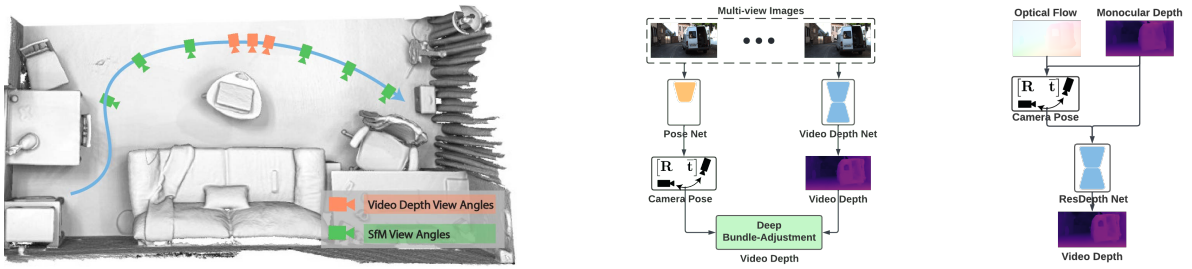
Figure 5.1 Video Depth Performance Comparison on KITTI Dataset. We mark the methods taking different numbers of frames with different colors. We propose a two-view video depth estimation method that substantially outperforms prior two-view, three-view, and five-view methods. Our method uses a monocular depth as initialization. The arrow marks our improvement when using the BTS [137] as the initialization. Comparison is detailed in Tab. 5.1.

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 sub-tasks: 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) Limited view angles of video depth

(b) Prior Multi-View

(c) Ours Two-View

Figure 5.2 (a) Unlike classic SfM, video depth estimation possesses significantly fewer view angles during inference. (b) Prior multi-view video depth estimation works [240, 237, 258] 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).

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 [84] and NYUv2 [178] 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 [240].

5.2 Prior Works

5.2.1 Pose and Depth from Multi-View System

Structure-from-motion (SfM) [157, 209, 196, 287, 85] is the classic approach to recover scene geometry and camera motion from video. After proper initialization, the pose and 3D points

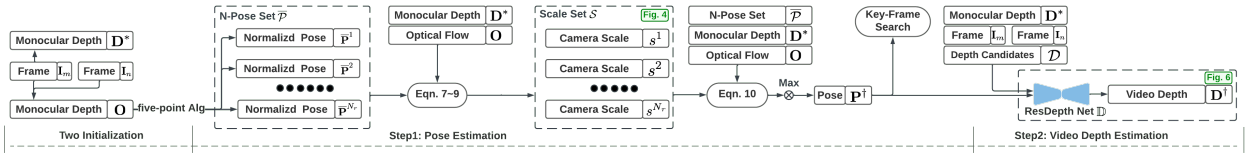


Figure 5.3 Our algorithm takes two RGB inputs ($\mathbf{I}_m, \mathbf{I}_n$), the initial mono-depth $\mathbf{D}^*$, and flowmap $\mathbf{O}$ as inputs. Our proposed framework consists of 2 key steps: (1) An improved five-point algorithm. Given flowmap $\mathbf{O}$ and mono-depth map $\mathbf{D}$, apply 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. 5.4 and 5.6.

are finetuned by bundle-adjustment over the input point correspondences. Visual simultaneously localization and mapping (vSLAM) methods [230, 244, 74, 73, 174, 179, 242, 287] 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 [240, 280, 94, 237, 75, 323, 258, 110] 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, [280] and [240] separately propose a first-order and second-order deep optimization scheme. [280] applies an exhaustive search over a local region in the pose parameter space. Given the projection flow computed by the current depth and pose, [240] employs a motion module to estimate a residual flow term. The pose is refined via applying a Gauss-Newton update [285]. 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 [240, 280].

5.2.2 Deep Two-View Structure-from-Motion

SfMR [268] revisits the classic two-view SfM [57, 130] 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.

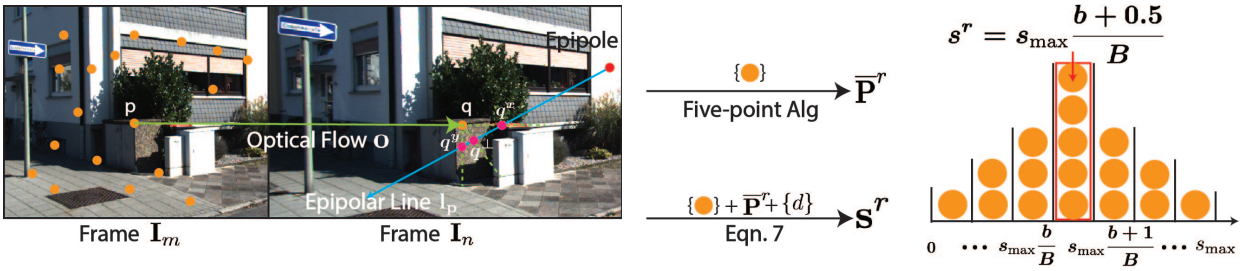
Our method improves [268] in multiple perspectives. First, we validate that the optical flow is more robust than the projection flow between immediate frames (detailed in Supp.Tab. 1.). This completes the motivation of estimating normalized pose from the flowmap instead of applying deep bundle-adjustment. In comparison, [268] only discusses its improvement over classic SIFT [159] based two-view SfM. Second, we improve indoor pose estimation performance by including the additional projection constraint. Third, the normalized depth in [268] 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 [268] 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.

5.2.3 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 [266, 55, 298, 299, 265, 156] assume sufficient view variations, estimating without an init mono-depthmap. A concurrent MVS work [6], however, positions itself to infer depth within a limited frame window. [6] 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 [6] uses groundtruth poses and more frames, we still outperform this iterative method, as in Tab. 5.1.

5.3 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



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

Figure 5.4 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 $\mathbf{l_p}$. In Eqn. 5.6, 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 $\mathbf{l_p}$. (b) One normalized pose $\overline{\mathbf{P}}^r$ is initiated by five-point algorithm. Next, with Eqn. 5.7, 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.

to compute the depth $\mathbf{D}^\dagger$ and pose $\mathbf{P}^\dagger$ by optimizing the photometric constraint:

$$\mathbf{P}^\dagger, \mathbf{D}^\dagger = \arg\min_{\mathbf{P},\mathbf{D}} h_p\left(g\left(f\left(\mathbf{D},\mathbf{P}\right),\mathbf{I}_n\right),\mathbf{I}_m\right), \tag{5.1}$$

where $h_p(\cdot)$ can be defined in forms such as structural similarity index measure (SSIM) [276, 327]. Recent multi-view works [240, 280, 94, 237, 75, 323, 258, 110] focus on improved mechanisms which, **in inference time**, enforce Eqn. 5.1. Typically, they adopt an iterative and alternative optimization scheme, minimizing Eqn. 5.1 by iteratively solving:

$$\begin{cases} \mathbf{P}^\dagger = \arg\min_{\mathbf{P}} h_p\left(g\left(f\left(\mathbf{D},\mathbf{P}\right),\mathbf{I}_j\right),\mathbf{I}_i\right) & \text{(5.2a)} \\ \mathbf{D}^\dagger = \arg\min_{\mathbf{D}} h_p\left(g\left(f\left(\mathbf{D},\mathbf{P}\right),\mathbf{I}_j\right),\mathbf{I}_i\right). & \text{(5.2b)} \end{cases}$$

For simplicity, Eqn. 5.2 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 [240, 280, 94, 237, 75, 323, 258, 110]. 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:

$$\begin{cases} \overline{\mathbf{P}}^{\dagger}, s^{\dagger} = \arg\min_{\overline{\mathbf{p}}, s} \left(h_e \left(\overline{\mathbf{P}}, \mathbf{O} \right) + \right. \\ \qquad\qquad \left. \lambda \cdot h_c \left(f \left(\mathbf{D}^*, p \left(\overline{\mathbf{P}}, s \right) \right), \mathbf{O} \right) \right) & \text{(5.3a)} \\ \mathbf{D}^{\dagger} = \arg\min_{\mathbf{D}} h_p \left(g \left(f \left(\mathbf{D}^*, p \left(\overline{\mathbf{P}}, s \right) \right), \mathbf{I}_j \right), \mathbf{I}_i \right). & \text{(5.3b)} \end{cases}$$

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. 5.3.1.

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. 6.3.

5.3.1 Pose Estimation

We optimize Eqn. 5.3a in camera pose estimation. Given the flowmap $\mathbf{O}$ and mono-depthmap $\mathbf{D}^*$, we reformulate the five-point [143] algorithm with RANSAC [77] 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. 5.4. 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 [143] 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 $\overline{\mathcal{P}} = \{\overline{\mathbf{P}}^r \mid r \in N_r\}$.

(a) KITTI Flow (b) NYUv2 Flow

Figure 5.5 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. 5.10. 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.

Pixel-wise scale estimation. Given any normalized pose $\overline{\mathbf{P}} = \begin{bmatrix} \mathbf{R} & \bar{\mathbf{t}} \end{bmatrix}$, 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 $\mathbf{p} = \begin{bmatrix} p^x & p^y & 1 \end{bmatrix}^\intercal$ and $\mathbf{q} = \begin{bmatrix} q^x & q^y & 1 \end{bmatrix}^\intercal$ 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:

$$d'\mathbf{q} = d' \begin{bmatrix} q^x & q^y & 1 \end{bmatrix}^\intercal = d\mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1}\,\mathbf{p} + s\mathbf{K}\,\bar{\mathbf{t}}. \tag{5.4}$$

The d and d' refer to depth at frame $\mathbf{I}_m$ and $\mathbf{I}_n$. By arranging Eqn. 5.4, we acquire the relationship between depth d and scale s at horizontal and vertical directions separately as:

$$d^x = s\frac{x - q^x \cdot z}{q^x \mathbf{m_3}^\intercal\,\mathbf{p} - \mathbf{m_1}^\intercal\,\mathbf{p}},\ d^y = s\frac{y - q^y \cdot z}{q^y \mathbf{m_3}^\intercal\,\mathbf{p} - \mathbf{m_2}^\intercal\,\mathbf{p}}. \tag{5.5}$$

Here $\begin{bmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3 \end{bmatrix}^\intercal = \mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1}$, $\begin{bmatrix} x & y & z \end{bmatrix}^\intercal = \mathbf{K}\,\bar{\mathbf{t}}$. As in Fig. 5.4 (a), optical flow induced pixel $\mathbf{q}$ may not reside on the epipolar line $\mathbf{l_p}$, 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:

$$s = \arg\min_s \left(d^x - d\right)^2 + \left(d^y - d\right)^2. \tag{5.6}$$

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

$$\log(s) = \log(d) + m, \tag{5.7}$$

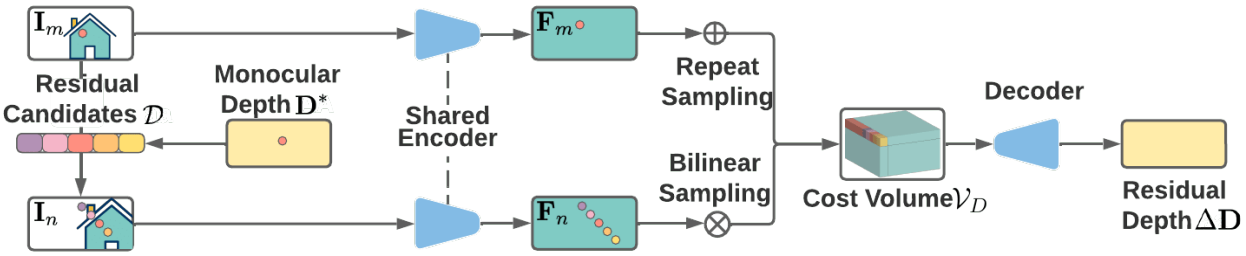


Figure 5.6 Illustration of video depth estimation. The shared encoder is drawn as one for simplicity in Fig. 6.3. The encoder and decoder of video depth network $\mathbb{D}$ are plotted. We dynamically sample the residual depth candidates $\mathcal{D}$ in log space centering around the initial depthmap $\mathbf{D}^*$. 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 $\mathbb{D}$.

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. 5.4. 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:

$$\mathbf{r}[b] = \sum_{k=1}^{N_k}\left(\frac{b}{B}\cdot s_{\max} \le s_k < \frac{b+1}{B}\cdot s_{\max}\right). \tag{5.8}$$

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:

$$s^r = s_{\max}\frac{b+0.5}{B}, \qquad b = \arg\max_{0\le b<B}\mathbf{r}[b]. \tag{5.9}$$

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. 5.5, we introduce an additional projection constraint h_c to stabilize the five-point algorithm 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:

Method	Venue	Frame	Labels	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
DORN [79]	CVPR'18	1	D	0.069	0.300	2.857	0.112	0.945	0.998	0.996
BTS [137]	Arxiv'18	1	D	0.059	0.245	2.756	0.096	0.956	0.993	**0.998**
AdaBins [18]	CVPR'21	1	D	0.058	0.190	2.360	0.088	0.964	0.995	**0.999**
NeWCRFs [309]	CVPR'22	1	D	**0.052**	**0.155**	**2.129**	**0.079**	**0.974**	**0.997**	**0.999**
Ours + BTS [137]	CVPR'23	2	D+F	**0.037**	0.110	1.809	**0.059**	0.987	**0.998**	**0.999**
Ours + AdaBins [18]		2	D+F	0.045	0.108	1.817	0.064	0.987	**0.998**	**0.999**
Ours + NeWCRFs [309]		2	D+F	0.041	**0.107**	**1.748**	**0.059**	**0.989**	**0.998**	**0.999**
BA-Net [237]	ICLR'19	5	D+P	0.083	0.025	3.640	0.134	-	-	-
SfMR [268]	CVPR'21	2	D+F+P	0.055	0.224	2.273	0.091	0.956	0.984	0.993
DeepMLE [58]	Arxiv'22	2	D+F+P	0.060	0.203	2.257	0.089	0.967	**0.995**	**0.999**
DRO [94]	Arxiv'21	2	D+P	0.047	0.199	2.629	0.082	0.970	0.994	0.998
MaGNet [6]	CVPR'22	3	D	0.051	**0.160**	2.077	0.079	0.974	**0.995**	**0.999**
DeepV2D [240]	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 [110]	ICPRAI'22	5	D+P	**0.037**	0.167	**1.984**	**0.073**	**0.978**	0.994	-
Ours + MonoDepth2 [88]	CVPR'23	2	D+F	0.032	0.106	1.889	0.057	0.986	**0.998**	**0.999**
Ours + BTS [137]		2	D+F	0.029	0.098	1.729	0.053	0.989	**0.998**	**0.999**
Ours + AdaBins [18]		2	D+F	0.030	0.089	1.655	0.052	0.989	**0.998**	**0.999**
Ours + NeWCRFs [309]		2	D+F	**0.028**	**0.087**	**1.597**	**0.049**	**0.991**	**0.998**	**0.999**

Table 5.1 **KITTI Monocular Video Depth Evaluation** on Eigen split [71] with Garg crop [82] capped at 80 meters using semi-dense groundtruth [257]. The lower half table applies median scaling [325] 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 [166, 32]]

$$\begin{cases} h_e(\overline{\mathbf{P}}^r, \{\mathbf{o}\}) = \sum_{k=1}^{N_r} \left(\mathbf{q}_k^\intercal \mathbf{K}^{-\top} \mathbf{E} \mathbf{K}^\intercal \mathbf{p}_k < k_e \right) & \text{(5.10a)} \\ h_c(\overline{\mathbf{P}}^r, s^r, \{\mathbf{p}\}, \{\mathbf{q}\}, \{d\}) = \\ \quad \sum_{k=1}^{N_r} \left(\| f(d_k, p(\overline{\mathbf{P}}^r, s^r)) - \mathbf{q}_k \|^2 < k_c \right). & \text{(5.10b)} \end{cases}$$

Here $\mathbf{E}$ is an essential matrix, expressed by the matrix form of the cross product $[\,]_\times$ as $\mathbf{E} = \mathbf{R}[\bar{\mathbf{t}}]_\times$. 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. 5.5, 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

Method	Venue	Frame	Abs Rel	Sc Inv	RMSE	log10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
DORN [79]	CVPR'18	1	0.115	-	0.509	-	0.828	0.965	0.992
BTS [137]	Arxiv'18	1	0.108	0.115	0.404	0.047	0.885	0.978	0.994
AdaBins [18]	CVPR'21	1	0.103	0.106	0.370	0.044	0.903	0.983	0.997
NewCRFs [309]	CVPR'22	1	**0.095**	**0.090**	**0.334**	**0.041**	**0.922**	**0.992**	**0.998**
Ours + BTS [137]	CVPR'23	2	0.102	0.098	0.356	0.044	0.903	0.984	0.997
Ours + AdaBins [18]		2	0.095	0.089	0.326	0.040	0.923	0.990	0.998
Ours + NewCRFs [309]		2	**0.090**	**0.080**	**0.306**	**0.038**	**0.935**	**0.995**	**0.999**
DfUSMC [98]	CVPR'16	Multi	0.447	0.456	1.793	0.169	0.487	0.697	0.814
DeMoN [258]	CVPR'17	2	0.144	0.179	0.775	0.061	0.805	0.951	0.985
DeepV2D [240]	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 [137]	CVPR'23	2	0.070	0.098	0.280	0.030	0.948	0.991	0.998
Ours + AdaBins [18]		2	0.064	0.089	0.255	0.027	0.961	0.994	0.999
Ours + NewCRFs [309]		2	**0.057**	**0.080**	**0.230**	**0.025**	**0.971**	**0.996**	**0.999**

Table 5.2 **NYUv2 Monocular Video Depth Evaluation.** Results in the lower half table apply median scaling in evaluation. Results of DeMoN [258] is from [240]. Results of 2-view DeepV2D [240] are evaluated with the published code and pretrained model. [Key: Red color marks Best, Blue color marks the Second Best, Frame marks the number of frames in inference]

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. 6.3, 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. 5.8 and Eqn. 5.9 if odometry is desired.

5.3.2 Video Depth Estimation

To this end, we have optimized Eqn. 5.3a. To optimize Eqn. 5.3b 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. 6.3). 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:

$$\mathcal{D} = \left\{ \mathbf{D}_i \parallel \mathbf{D}_i = \exp(\Delta d_i) \cdot \mathbf{D}^* \right\}_{i=1}^{k_\mathcal{D}}. \tag{5.11}$$

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

$$\mathcal{F}_d^* = \left\{\mathbf{F}_i^* \parallel \mathbf{F}_i^* = g(f(\mathbf{D}_i, \mathbf{P}), \mathbf{F}_n)\right\}_{i=1}^{k_\mathcal{D}}. \tag{5.12}$$

$\mathcal{V}_D$ is then constructed by stacking $\mathcal{F}_d^*$ and the repetition of input feature $\mathbf{F}_n$, illustrated in Fig. 5.6.

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:

$$\mathbf{D}^\dagger = \mathbf{D}^* \cdot \exp(\mathbf{\Delta D}) = \mathbf{D}^* \cdot \exp(\mathbb{D}(\mathcal{V}_D)). \tag{5.13}$$

Supervision Signal. Following [137], we use a scale-invariant loss, to supervise the training of the depth network,

$$D(w) = \frac{1}{n}\sum_{i=1}^{n} w_i^2 - \Big(\frac{1}{n}\sum_{i=1}^{n} w_i\Big)^2 + (1-\mu)\Big(\frac{1}{n}\sum_{i=1}^{n} w_i\Big)^2, \tag{5.14}$$

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

5.3.3 Equality of Scale and Video Depth Learning

In Fig. 6.3, scale is required before video depth estimation. Though scale can be optimized over an initial mono-depthmap, 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. 5.7, we define the optimal scale s^* as the average of pixel-wise scale $\mathbf{s}$:

$$\log(s^*) = \frac{1}{n}\sum_{i=1}^{n} \log(s_i) = \frac{1}{n}\sum_{i=1}^{n} \big(\log(d_i) + m_i\big). \tag{5.15}$$

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:

$$\begin{aligned} L_{s^*} &= \| \log(\tilde{s}) - \log(s^*) \| \\ &\leq \frac{1}{n}\sum_{i=1}^{n} \| \log(\tilde{d}_i) - \log(d_i) \| + \| \frac{1}{n}\sum_{i=1}^{n} (\tilde{m}_i - m_i) \|. \end{aligned} \tag{5.16}$$

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. 5.16, 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. 5.5.

Seq	Err	BetterGen* [322]	LTMVO* [334]	DfVWild* [90]	MLF-VO [120]	SfMR [268]	LSR* † [264]	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 [240]	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 5.3 **KITTI Odometry Evaluation**. Results in the right of the table are trained on Eigen split [71] 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) [264]. [Key: Red color marks Best, Blue color marks the Second Best]

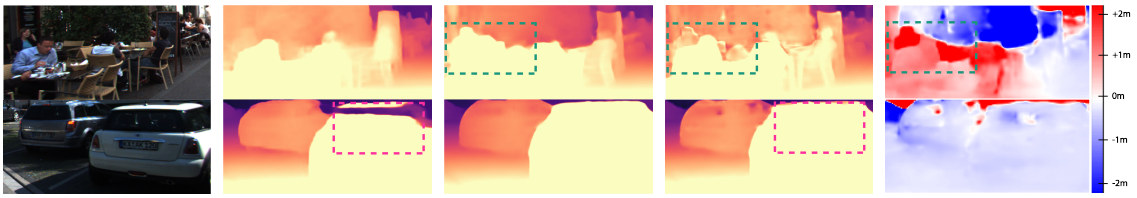
5.4 Experiments

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 [128] 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 [137] during training but test against various mono-depth inputs. For all three monocular methods, BTS [137], AdaBins [18], and NewCRFs [309], we use the author released models. The Monodepth2 [88] is re-trained by us. For flow, we adopt the publicly available model of RAFT [241] trained using the synthetic datasets [166]. 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 [322] 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 [139] algorithm as a replacement if the five-point algorithms fail.

5.4.1 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 [71], evaluated on semi-dense groundtruth [257] under Garg crop [82] capped at 80 meters. Tab. 5.1 reports results in standard



(a) RGB Input $\mathbf{I}_t$ (b) DeepV2D [240] (c) Init $\mathbf{D}^*$ (d) Finetuned $\mathbf{D}^\dagger$ (e) Residual (meter)

Figure 5.7 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.

7 metrics [71], 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 [94, 268], 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 [240], 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. 5.7 shows our improvement qualitatively. Finally, our performance gain over prior SoTA does not attribute to monocular initialization. In Tab. 5.1, our result still substantially outperforms DeepV2D with a lightweight MonoDepth2 monocular initialization.

NYUv2 Depth NYUv2 dataset [178] has RGB and depth image pairs in indoor environments. Our experiment follows the standard train/test split [71]. As NYUv2 is captured by a handheld camera, rotation frequently dominates camera motion across frames, which is undesirable for video depth estimation (see Fig. 5.5). Despite all the hurdles, our 2-view performance grouped with NewCRFs [309] 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 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 met-

ScanNet	DeMoN [258]	BA-Net [237]	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 5.4 **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: Red color marks Best, Blue color marks the 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 5.5 **Ablation on Outdoor Video Depth Estimation.** [Key: 'ResDepth'= Residual depth learning (Sec. 5.3.2). 'PoseEstimation'= Proposed Pose Estimation Method (Sec. 5.3.1). 'ScaleNet'=Further refine pose scale with an additional ScaleNet (detailed in Supplementary).]

rics [90]. For methods [322, 334, 90, 264, 268, 120], we follow [90] to train/test on sequences 00-08/09-10. For DeepV2D [240], as trained on Eigen split [71], 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 [322] during inference. In Tab. 5.3, we outperform SoTA on rotation and translation errors.

ScanNet Pose ScanNet [52] 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.

5.4.2 Ablation Study

The Equality between Scale and Video Depth Learning In Tab. 5.5 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. 5.3 and 1.24 from Tab. 5.5). This is expected, as the LiDAR depth possesses less noise than IMU and GPS pose. Thus we empirically demonstrate the equality

	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 5.6 **Ablation on Indoor Video Depth Estimation.** [Key: 'FivePoint'=Baseline Five-point algorithm with RANSAC. 'PoseEstimation'=Proposed Pose Estimation Method (Sec. 5.3.1). 'KeySearch'=Keyframe search. Bold marks the best score.]

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 [268] in Tab. 5.1 and the performance gap in row 1 and 2 of Tab. 5.5. Meanwhile, it avoids artifacts in moving objects, as in Fig. 5.7.

Pose Estimation and Key-frame Search Compared to using baseline five-point algorithm over flow estimate [268, 322], our proposed method benefits both pose and depth performance, as shown in Tabs. 5.4 and 5.6. Also, ensuring sufficient camera translation shows noticeable improvement, as shown in Tab. 5.6.

Computational Efficiency We compare the running time to DeepV2D [240] on an RTX 2080 Ti GPU, for 192×1088 images. In Fig. 6.3, our inference has $1 + 2$ steps: initialization of flow [241] and mono-depth [137], 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.

5.5 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.

CHAPTER 6

RSFM: REVISIT SELF-SUPERVISED DEPTH ESTIMATION WITH LOCAL STRUCTURE-FROM-MOTION

Both self-supervised depth estimation and Structure-from-Motion (SfM) recover scene depth from RGB videos. Despite sharing a similar objective, the two approaches are disconnected. Prior works of self-supervision backpropagate losses defined within immediate neighboring frames. Instead of learning-through-loss, this work proposes an alternative scheme by performing local SfM. First, with calibrated RGB or RGB-D images, we employ a depth and correspondence estimator to infer depthmaps and pair-wise correspondence maps. Then, a novel bundle-RANSAC-adjustment algorithm jointly optimizes camera poses and one depth adjustment for each depthmap. Finally, we fix camera poses and employ a NeRF, however, without a neural network, for dense triangulation and geometric verification. Poses, depth adjustments, and triangulated sparse depths are our outputs. For the first time, we show self-supervision within 5 frames already benefits SoTA supervised depth and correspondence models. Despite self-supervision, our pose algorithm has certified global optimality, outperforming optimization-based, learning-based, and NeRF-based prior arts.

6.1 Introduction

Monocular depth estimation [79, 137] infers depthmap from a single image. It is an essential vision task with applications in AR/VR [182], autonomous driving [84], and 3D reconstruction [31]. Most methods [309, 19, 195, 189] supervise the model with groundtruth collected from stereo cameras [319] or LiDAR [84]. Recently, self-supervised depth [90, 88, 327] has drawn significant attention due to its potential to scale up depth learning from massive unlabeled RGB videos.

Classic SfM methods [209, 228, 205, 286, 1, 50] also reconstruct scene depth from unlabled RGB videos. Despite its relevance, SfM is rarely applied to self-supervised depth learning. We outline two potential reasons. First, SfM is an off-the-shelf algorithm unrelated to the depth estimator. Scale ambiguity renders SfM poses and depths at different scales compared to depth models. Second, self-supervision has a well-defined training scheme to work with universal unlabeled videos. It backpropagates through photometric loss computed within immediate neighboring frames, *e.g.*,

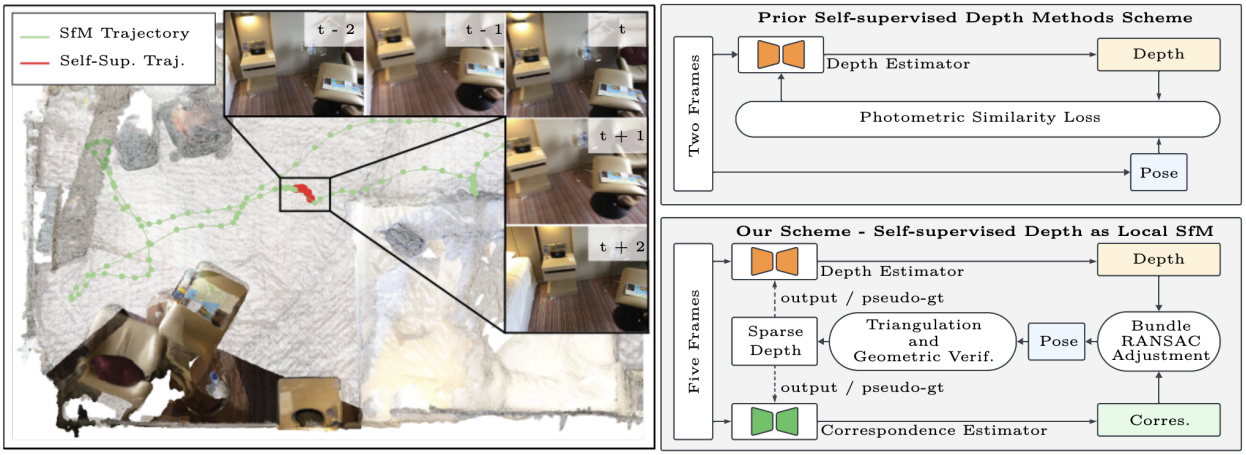


Figure 6.1 **Revisit Self-supervision with Local SfM.** The work proposes alternating the learning-through-loss with a local SfM pipeline for self-supervised depth estimation. We summarize our differences. On self-supervision: (1) Instead of using naive two-view camera poses, we propose a Bundle-RANSAC-Adjustment pose optimization algorithm with multi-view constraints. (2) Instead of backpropagating through a loss, we produce a sparse point cloud with explicit triangulation and geometric verification. The point cloud serves as either output or pseudo-groundtruth for self-supervision. On SfM: (1) Our local SfM is adapted to use estimated monocular depthmaps and automatically resolve their scale inconsistency between pairs of images. (2) We maintain accuracy under significant sparse view variations, *e.g.*, red trajectories. We generalize SfM to as few as 5 frames, similar to the number of images used to define self-supervision loss.

red trajectory in Fig. 7.1. In contrast, SfM is more selective to input videos. It requires images of diverse view variations (green trajectory in Fig. 7.1), being inaccurate and unstable when applied to a small frame window.

This work connects self-supervision with SfM. We replace the self-supervision loss with a complete SfM pipeline that maintains robustness to a local window. Shown in Fig. 6.2, with N frames as input, our algorithm outputs $N-1$ camera poses, $N-1$ depth adjustments, and the sparse triangulated point cloud. In initialization, N monocular depthmaps and $N \times (N-1)$ pairwise correspondence maps are inferred. Next, we propose a Bundle-RANSAC-Adjustment pose estimation algorithm that retains accuracy for second-long videos. The algorithm utilizes the 3D priors from monocular depthmap to compensate for the deficient camera views. Correspondingly, we optimize $N-1$ depth adjustments to alleviate the depth scale ambiguity by temporally aligning to the root frame depth.

The Bundle-RANSAC-Adjustment extends two-view RANSAC with multi-view bundle-adjustment

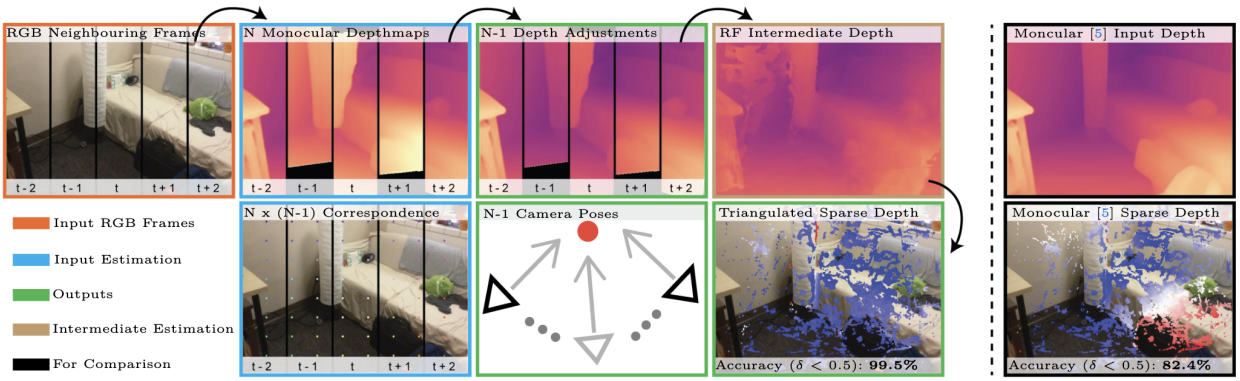


Figure 6.2 **Local Structure-from-Motion.** With N neighboring frames, we extract monocular depthmaps and pairwise dense correspondence maps with methods, *e.g.*, ZoeDepth [19] and PDC-Net [251]. Next, skipping the root frame, we optimize the rest $N-1$ camera poses and depth adjustments. The depth adjustments render input depthmaps **temporally consistent**. Fixing poses and adjustments, we use the Radiance Field (RF) for triangulation. A geometrically verified sparse root depthmap is output. Our local SfM applies **self-supervision** with only 5 RGB frames. Yet, our sparse output already outperforms the input supervised depth with SoTA performance.

(BA). The algorithm has quadratic complexity and is designed for parallel GPU computation. We RANdomly SAmple and hypothesize a set of normalized poses. In Consensus checking, we apply BA to evaluate a robust inlier-counting scoring function over multi-view images. Camera scales and depth adjustments are determined during BA to maximize the scoring function.

Next, we freeze the optimized poses and employ a Radiance Field (RF), *i.e.*, a NeRFF [170] without a neural network, for triangulation. We optimize RF to achieve multi-view depthmap and correspondence consistency within a shared 3D frustum volume. For outputs, we apply geometric verification to extract multi-view consistent point cloud, *i.e.*, a sparse root depthmap.

Fig. 7.1 contrasts our method with prior self-supervised depth and SfM methods. To our best knowledge, there has not been prior work showing geometry-based self-supervised depth benefits supervised models. However, self-supervision is supposed to augment supervised models with unlabeled data. In Fig. 6.2, our unique pipeline gives the **first** evident results, that self-supervision with **as few as** 5 frames already benefits supervised models.

Despite depths, our multi-view RANSAC pose has certified global optimality under a robust scoring function. It outperforms prior arts in optimization-based [209, 330], learning-based [240, 273], and NeRF-based [253] pose algorithms.

Beyond pose and depth, our method has diverse applications. The depth adjustments from our method provide empirically consistent depthmaps, being important for AR image compositing. When with RGB-D inputs, our method enables self-supervised correspondence estimation. Our accurate pose estimation gives improved projective correspondence than the SoTA supervised correspondence input. An example is in Fig. 6.9. We summarize our contributions as:

- We propose a novel local SfM algorithm with Bundle-RANSAC-Adjustment.
- We show the **first** evident result that self-supervised depth with **as few as** 5 frames already benefit SoTA supervised models.
- We achieve SoTA sparse-view pose estimation performance.
- We enable self-supervised temporally consistent depthmaps.
- We enable self-supervised correspondence estimation with 5 RGB-D frames.

6.2 Related Works

Structure-from-Motion. SfM is a comprehensive task [209, 286]. A typical pipeline is, correspondence extraction [160, 254, 30], two-view initialization [17, 143], triangulation [142, 183], and local & global bundle-adjustment [209, 286]. Classic methods require diverse view variations for accurate reconstruction. Our method compensates SfM on scarse camera views via introducing deep depth estimator. Further, we suggest SfM itself is a self-supervised learning pipeline, as in Fig. 7.1. Finally, our SfM is not up-to-scale and shares the metric space as the input depthmap.

Sparse Multi-view Pose Estimation. Estimating poses from sparse frames is crucial for self-supervision [88, 325, 197, 312, 47], video depth estimation [330, 240, 93, 258], and sparse-view NeRF [253, 60, 119, 148, 180]. Camera poses are estimated either by learning [88, 47, 240, 93], optimization [330, 322] or together with NeRF [253, 148]. We propose an additional multi-view RANSAC pipeline with improved accuracy.

Self-supervised Depth and Correspondence Estimation. Multiple works improve self-supervised depth in different ways, including learning loss [88, 278, 191], architecture [95, 326],

camera pose [165, 322, 45], joint with semantics segmentation [327], and using large-scale data [229, 295]. Recently, [229] shows self-supervision only performs on-par with supervised models under substantially more data. [295] shows the benefit of self-supervision via exploiting non-geometry monocular semantic consistency. Our method shows the first evident results where self-supervision benefits supervised models with only 5 consecutive frames.

Consistent Depth Estimation. AR applications necessitate temporally consistent depthmaps, *i.e.*, depthmaps from different temporal frames reside in the same 3D space. Recent works [321, 162] align depthmap according to the poses and points from the off-the-shelf COLMAP algorithm. Our method seamlessly integrates SfM with monocular depthmaps, outputting consistent depth and poses.

Test Time Refinement (TTR). TTR aims to improve self-supervised / supervised depth estimators in testing time with RGB video [45, 35, 279, 221, 134]. Methods [116, 245] rely on off-the-shelf algorithms for pseudo depth and pose labels. Recently, [116] first shows TTR improves supervised models. TTR is our downstream application, which details strategies for utilizing noisy pseudo-labels.

6.3 Methodology

Our method runs sequentially. From N calibrated images $\mathcal{I}$, we extract N monocular depthmaps $\mathcal{D}$ and $N \times (N-1)$ pair-wise dense correspondence C. We split the N images into one root frame $\mathbf{I}_o$ in the center of the N-frame window where $o = \lfloor \frac{N+1}{2} \rfloor$, and $N-1$ support frames $\mathbf{I}_i$, where $i \in \mathbb{N}^+ = [1, N] \backslash \{o\}$. In Sec. 6.3.1, after setting the root frame as identity pose, we use Bundle-RANSAC-Adjustment to optimize $N-1$ poses $\mathcal{P}$ and $N-1$ depth adjustments $\mathcal{R}$. Next, in Sec. 6.3.2, we apply triangulation by optimizing a frustum Radiance Field (RF) $\mathbf{V}$, *i.e.*, a NeRF without network. Finally, in Sec. 6.3.3, we apply geometric verification by rendering multi-view consistent 3D points from RF. An overview is in Fig. 6.3.

6.3.1 Bundle-RANSAC-Adjustment Pose Estimation

We generalize two-view RANSAC with multi-view constraints through Bundle-Adjustment. Sec. 6.3.1.1 describes our pipeline. In Sec. 6.3.1.2, we propose Hough transform to accelerate

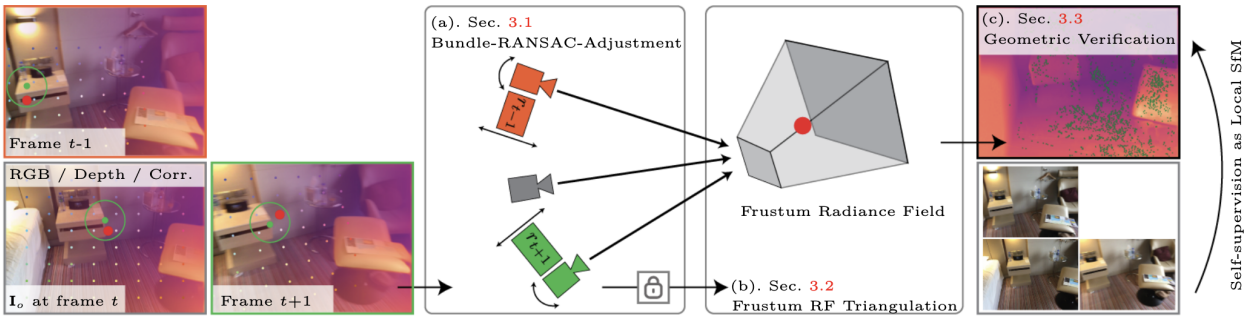


Figure 6.3 **Algorithm Overview.** After extracting monodepths and correspondence maps from inputs: (a) We apply Bundle-RANSAC-Adjustment to optimize $N-1$ camera poses $\mathcal{P}$ and $N-1$ depth adjustments $\mathcal{R}$. (b) We fix poses and depth adjustments and optimize a frustum Radiance Field (RF) for triangulation. (c) We apply geometric verification to extract multi-view consistent 3D points via rendering with RF. We further detail step (a) in Fig. 6.4, 6.5, and 6.6, and steps (b) and (c) in Fig. 6.7.

computation. We discuss the time complexity in Sec. 6.3.1.3.

6.3.1.1 Optimization Pipeline

RANdom SAmple. We use five-point algorithm [143] as the minimal solver. We execute it between root and each support frame, extracting a pool of $(N-1)\times K$ normalized poses (*i.e.*, pose of unit translation), $\overline{\mathcal{Q}} = \{\overline{\mathbf{P}}_i^k \mid i \in \mathbb{N}^+, k \in [1, K]\}$, where $\overline{\mathbf{P}}_i^k \in \mathbb{R}^{3\times4}$. The K is the number of normalized poses extracted per frame. We term a set of $N-1$ normalized poses as a group $\overline{\mathcal{P}} \in \mathbb{R}^{(N-1)\times3\times4}$. Two-view RANSAC enumerates over single normalized pose $\overline{\mathbf{P}}$. Our multi-view algorithm hence enumerates over normalized pose group $\overline{\mathcal{P}}$. We initialize the optimal group $\overline{\mathcal{P}}^*$ as the top candidate from K poses of $\overline{\mathcal{Q}}$ for each frame. See examples in Fig. 6.4.

Bundle-Adjustment Consensus. While computing consensus counts, the camera scales $\mathcal{S}$ and depth adjustments $\mathcal{R}$ are automatically determined with bundle-adjustment to maximize a robust scoring function:

$$\rho_i = \phi(\overline{\mathcal{P}}) = \max_{\mathcal{S},\mathcal{R}} f(\mathcal{S}, \mathcal{R} \mid \overline{\mathcal{P}}, \mathcal{D}, C). \tag{6.1}$$

Search for Optimal Group. Our multi-view RANSAC has a significantly larger solution space than two-view RANSAC. With N view inputs, we determine the optimal group out of K^{N-1} combinations. Hence, we iteratively search for the optimal group with a greedy strategy. For each

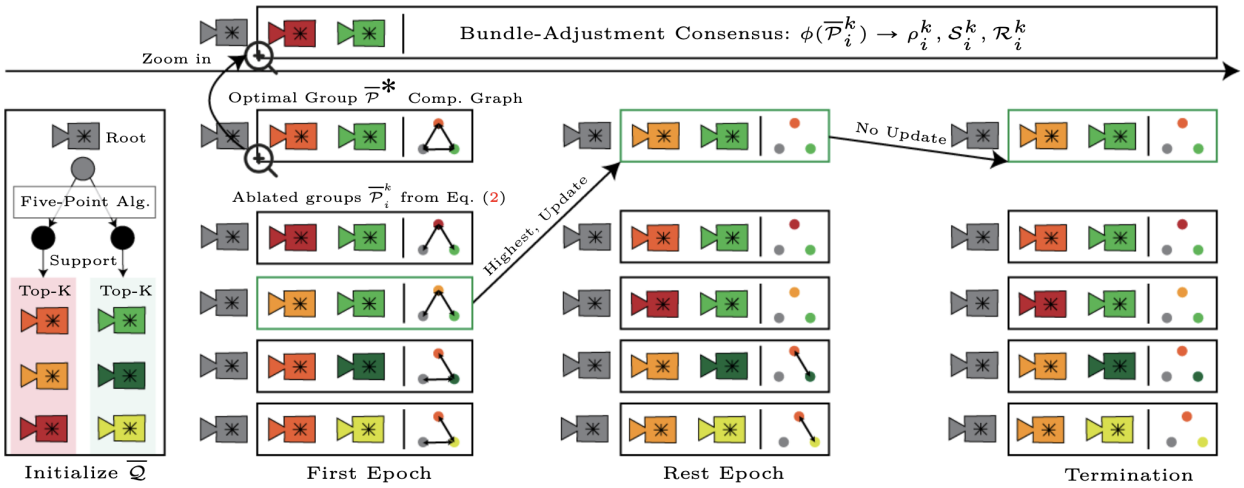


Figure 6.4 **Pose Optimization Pipeline.** We show a sample execution when $N=3$ and $K=3$. We initialize normalized pose candidates pool $\overline{\mathcal{Q}}$. Optimal group $\overline{\mathcal{P}}^*$ is set to top candidates within $\overline{\mathcal{Q}}$. In each epoch, Eq. (6.2) ablates pose group $\overline{\mathcal{P}}_i^k$. Each group is scored with Eq. (6.1) via BA with Hough Transform, detailed in Sec. 6.3.1.2. The optimal group with the highest score is updated with Eq. (6.3). Termination occurs when the maximum score stabilizes. We maintain quadratic complexity by avoiding repetitive computation after the first epoch, shown with the Comp. Graph, detailed in Sec. 6.3.1.3.

epoch, we ablate $(N-1)(K-1)$ additional pose groups:

$$\overline{\mathcal{P}}_i^k = \overline{\mathcal{P}}_i^* \setminus \{\overline{\mathbf{P}}_i^*\} \cup \{\overline{\mathbf{P}}_i^k\}, \tag{6.2}$$

where $i \in \mathbb{N}^+$ and $k \in [1, K]$. Combine Eq. (6.2) and Fig. 6.4, taking frame i as an example, we replace the optimal pose $\overline{\mathbf{P}}_i^*$ by its $K-1$ other candidates $\overline{\mathbf{P}}_i^k$, generating $K-1$ groups. For N frames, we have $(N-1)(K-1)+1$ groups. We apply bundle-adjustment to each group to evaluate Eq. (6.1). As shown in Fig. 6.3 and Fig. 6.4, we select the normalized pose together with its optimized scales and depth adjustments that maximize the scores as the output,

$$\mathcal{P}_i^* = b(\overline{\mathcal{P}}_i^*, \mathcal{S}_i^*),\ \mathcal{R}_i^* = \mathcal{R}_i^k,\ \text{where } k = \arg\max\{\rho_i^k\},\ \overline{\mathcal{P}}_i^* = \overline{\mathcal{P}}_i^k,\ \mathcal{S}_i^* = \mathcal{S}_i^k, \tag{6.3}$$

where $b(\cdot)$ combines normalized poses with scales. Fig. 6.2 third column plots an adjusted temporal consistent depthmap after applying $\mathcal{R}^*$. In Fig. 6.4, the algorithm terminates when the maximum score stops increasing.

Scoring Function. Similar to other RANSAC methods, we adopt robust inlier-counting based

scoring functions. Expand Eq. (6.1) for a specific group $\overline{\mathcal{P}}$:

$$\phi(\overline{\mathcal{P}}) = \sum_{i,i\neq j} \sum_{j} f_{i,j}(s_i, s_j, r_i, r_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, \mathbf{D}_i, \mathbf{D}_j, \mathbf{C}_{i,j}), \tag{6.4}$$

where i, j are frame index. We set per-frame camera scale, depth, depth adjustment, and correspondence as $s \in \mathcal{S}$, $\mathbf{D} \in \mathcal{D}$, $r \in \mathcal{R}$, and $\mathbf{C} \in C$. The scoring function $f_{i,j}(\cdot)$ has various forms. First, we describe a 2D scoring function:

$$f_{i,j}^{\text{2D}}(\cdot) = \sum_{m} \mathbf{1}\left(\|\pi(s_i, s_j, r_i \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, d_i^m) - \mathbf{c}_{i,j}^m\|_2 < \lambda^{\text{2D}}\right), \tag{6.5}$$

where $m \in [1, M]$ indexes sampled pixels per frame pair. $f_{i,j}^{\text{2D}}(\cdot)$ measures the inlier count between depth projected correspondence and input correspondence. $\pi(\cdot)$ is projection process. Intrinsic is skipped. d and $\mathbf{c}$ are depth and correspondence sampled from $\mathbf{D}$ and $\mathbf{C}$. An example is in Fig. 6.5. The $\mathbf{1}(\cdot)$ is the indicator function. The projected pixel is an inlier if it resides within the circle of radius λ^{2D} and center at correspondence $\mathbf{c}_{i,j}^m$ (denoted as $\mathbf{p}_j$ in Fig. 6.5). $\mathbf{c}_{i,j}^m$ is sampled from correspondence map $\mathbf{C}_{i,j}$. Second, we introduce a 3D scoring function:

$$f_{i,j}^{\text{3D}}(\cdot) = \sum_{m} \mathbf{1}\left(\|\pi^{-1}(s_i \mid \overline{\mathbf{P}}_i, r_i, d_i^m) - \pi^{-1}(s_j \mid \overline{\mathbf{P}}_j, r_j, d_j^m)\|_2 < \lambda^{\text{3D}}\right). \tag{6.6}$$

Depth pair d_i and d_j is determined by correspondence. Unlike the 2D one, the 3D function fixes depth adjustment r. Function $\pi^{-1}(\cdot)$ back-projects 3D point.

6.3.1.2 Hough Transform Acceleration

Maximizing Eq. (6.1) for each pose group is computationally prohibitive, as shown in Fig. 6.4. We propose Hough Transform for acceleration. We use Eq. (6.5), the 2D function $f^{\text{2D}}(\cdot)$ as an example for illustration. See our motivation in Fig. 6.5.

Hough Transform. The relative pose between $\overline{\mathbf{P}}_i$ and $\overline{\mathbf{P}}_j$ is defined as:

$$\mathbf{P}_{i,j} = \mathbf{P}_j \mathbf{P}_i^{-1} = \begin{bmatrix} \mathbf{R}_{i,j} & s_{i,j}\bar{\mathbf{t}}_{i,j} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_j \mathbf{R}_i^{-1} & -s_i \mathbf{R}_j \mathbf{R}_i^{-1} \bar{\mathbf{t}}_i + s_j \bar{\mathbf{t}}_j \end{bmatrix}, \tag{6.7}$$

where $\mathbf{R}, \bar{\mathbf{t}}$, and s are rotation, normalized translation and pose scale. From Eq. (6.7) and Fig. 6.5, $\bar{\mathbf{t}}_{i,j}$ is controlled by the scale s_i and s_j, and thus we have:

$$\lim_{s_i \to +\inf} \bar{\mathbf{t}}_{i,j} = -\mathbf{R}_j \mathbf{R}_i^{-1} \bar{\mathbf{t}}_i, \qquad \lim_{s_j \to +\inf} \bar{\mathbf{t}}_{i,j} = \bar{\mathbf{t}}_j. \tag{6.8}$$

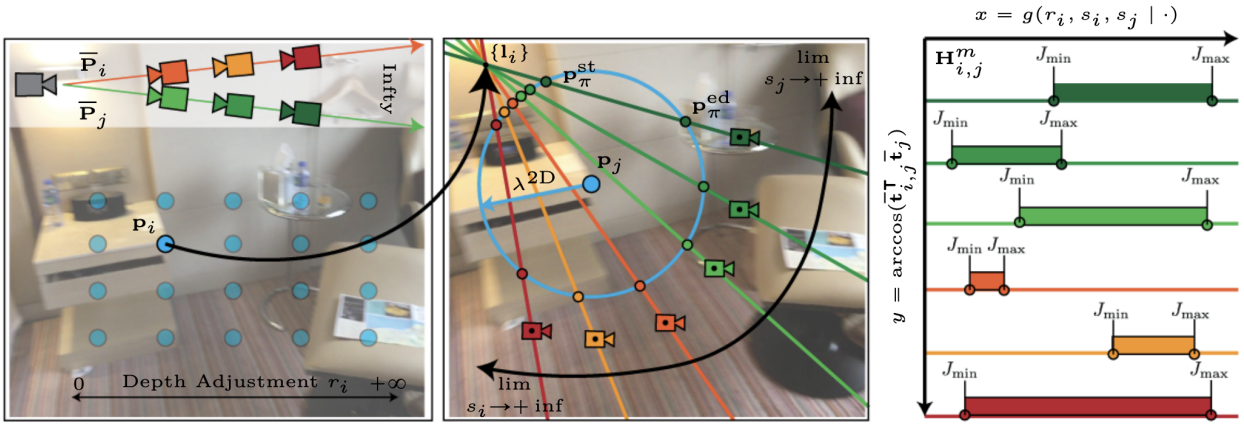


Figure 6.5 **Hough Transform between Two Normalized Poses.** With fixed normalized poses, there exists three variables, scales s_i & s_j of $\overline{\mathbf{P}}_i$ & $\overline{\mathbf{P}}_j$ and adjustment r_i. Pixel $\mathbf{p}_i$ and $\mathbf{p}_j$ are corresponded. Ablating pose scales maps pixel $\mathbf{p}_i$ to a set of epipolar lines $\{\mathbf{l}_i\}$, however, bounded by Red and Green at infinite scales. We have three observations. First, with fixed normalized poses, epipolar lines $\mathbf{l}_i$ have limited possibilities. Second, scale s and depth adjustment d are equivalent, both adjusting projection on epipolar line. Third, per epipolar line, to be an inlier, the projection has to reside within the line-circle intersection, between $\mathbf{p}_\pi^{\text{st}}$ and $\mathbf{p}_\pi^{\text{ed}}$. The observations motivate us to discretize the solution space to a 2D matrix, *i.e.*, Hough Transform. Right figure plots an example transformation $\mathbf{H}_{i,j}^m$ from frame i to j on the mth pixel $\mathbf{p}_i$.

For a pixel $\mathbf{p}_i$ on frame i, its corresponding epipolar line $\mathbf{l}_i$ on frame j is:

$$\mathbf{l}_i = \mathbf{K}^{-\mathsf{T}} [\,\bar{\mathbf{t}}_{i,j}]_\times \mathbf{R}_{i,j} \mathbf{K}^{-1} \mathbf{p}_i. \tag{6.9}$$

Eq. (6.8) and Eq. (6.9) suggest the epipolar line has limited possibilities. Operation $[\cdot]_\times$ is the cross product in matrix form. Further, as the depth re-projected pixel $\mathbf{p}_\pi$ of $\mathbf{p}_i$ always locate on the epipolar line $\mathbf{l}_i$ [99], we have:

$$\mathbf{l}_i^{\mathsf{T}} \mathbf{p}_\pi = 0, \;\; \mathbf{p}_\pi = \pi(s_i, s_j, r_i \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j, d_i). \tag{6.10}$$

To be an inlier of the scoring function $f^{\text{2D}}(\cdot)$, we have:

$$\|\mathbf{p}_\pi - \mathbf{p}_j\|_2 \leq \lambda^{\text{2D}}. \tag{6.11}$$

Combining Eq. (6.10), Eq. (6.11) and Fig. 6.5, to be an inlier, the projected pixel $\mathbf{p}_\pi$ has to reside within the line segment, with two end-points computed by the line-circle intersection. The circle centers at corresponded pixel $\mathbf{p}_j$ on frame j with a radius λ^{2D}. We denote the two end-points $\mathbf{p}_\pi^{\text{st}}$ and $\mathbf{p}_\pi^{\text{ed}}$. Function $J(\cdot)$ follows [330] Supp. Eq. (4), which maps a projected pixel $\mathbf{p}_\pi$ and adjusted depth $r_i d_i$ to camera scale $s_{i,j}$ as: $s_{i,j} = J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi)$.

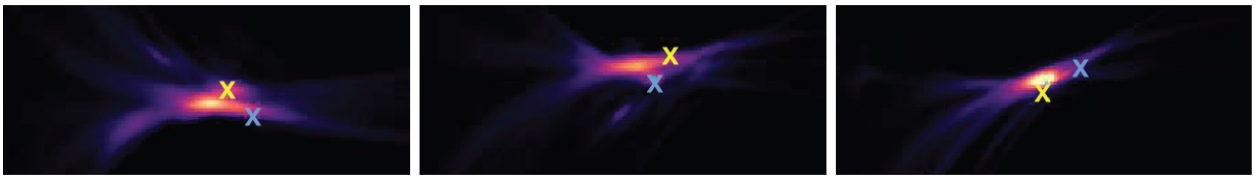


Figure 6.6 **Visualize Hough Transform Matrix $\mathbf{H}_i^j$** from Eq. (6.18). Area with higher intensity suggests more inlier counts. Given normalized pose group, for N views, there exists $N \times (N-1)$ matrices $\mathbf{H}_i^j$, constraining $N-1$ scale and $N-1$ adjustments. We plot the start and end points after optimizing Eq. (6.19) in the figure.

Corollary 1 *A pixel is an inlier iff:*

$$J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi^{st}) \leq s_{i,j} \leq J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi^{ed}). \tag{6.12}$$

Corollary 2 *Scale and depth are equivalent as;*

$$s_{i,j} = J(\overline{\mathbf{P}}_{i,j}, r_i d_i, \mathbf{p}_\pi) = r_i \cdot J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi). \tag{6.13}$$

Combine Eqs. (6.12) and (6.13),

$$J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{st}}) \leq \frac{s_{i,j}}{r_i} \leq J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{ed}}). \tag{6.14}$$

Set $g(\cdot)$ maps the variables under optimization to intermediate term $\frac{s_{i,j}}{r_i}$:

$$J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{st}}) \leq g(r_i, s_i, s_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j) \leq J(\overline{\mathbf{P}}_{i,j}, d_i, \mathbf{p}_\pi^{\text{ed}}). \tag{6.15}$$

The ith pixel is an inlier if and only if its projection satisfies Eq. (6.15). Note, the value space of function $g(\cdot)$ is mapped to a 2D space $\mathbf{H}$ after Hough Transform:

$$x = g(r_i, s_i, s_j \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j), \;\; y = \arccos(\bar{\mathbf{t}}_{i,j}^\intercal \bar{\mathbf{t}}_j), \tag{6.16}$$

where x and y are transformed coordinates. From Eq. (6.16), x is a synthesized translation magnitude and y is angular variable. We then set $x \in [0, x_{\max}]$, and $y \in [0, \theta_{\max}]$, where $\theta_{\max} = \arccos(-\bar{\mathbf{t}}_j^\intercal \mathbf{R}_{i,j} \bar{\mathbf{t}}_i)$. Finally, the value of $\mathbf{H}$ is:

$$\forall y \in [0, \theta_{\max}], \;\; \mathbf{H}(x \mid y) = 1, \text{if} \;\; x \in [J_{\min}, J_{\max}], \tag{6.17}$$

where $J_{\min}$ and $J_{\max}$ are the two bounds from Eq. (6.15). The transformation over the scoring function $f_{i,j}^{\text{2D}}$ with all M sampled pixels between frame $\mathbf{I}_i$ and $\mathbf{I}_j$:

$$\mathbf{H}_{i,j} = \sum_m \mathbf{H}_{i,j}^m, \;\; f_{i,j}^{\text{2D}}(s_i, s_j, r_i \mid \overline{\mathbf{P}}_i, \overline{\mathbf{P}}_j) = \mathbf{H}_{i,j}(x, y), \tag{6.18}$$

where x and y are functions of s_i, s_j, r_i. Eq. (6.1) becomes:

$$\phi(\overline{\mathcal{P}}) = \max_{\mathcal{S},\mathcal{R}} \sum_i \sum_{j, j\neq i} \mathbf{H}_{i,j}(x(\mathcal{S},\mathcal{R}), y(\mathcal{S},\mathcal{R})). \tag{6.19}$$

In our implementation, we discretize $\mathbf{H}_{i,j}$ to a 2D matrix.

Accelerate Bundle-Adjustment Consensus. The BA determines $N-1$ camera scales and $N-1$ depth adjustments to maximize the scoring function $\phi(\cdot)$ in Eq. (6.19). With Hough transform, BA maximizes the summarized intensity via **indexing** $N \times (N-1)$ Hough transform matrices $\mathbf{H}$. It avoids BA repetitively enumerating all sampled pixels. Fig. 6.6 shows an example optimization process.

Certified Global Optimality of robust inlier-counts scoring function Eq. (6.5) and Eq. (6.6) are achieved after optimization. See Fig. 6.8 for more analysis.

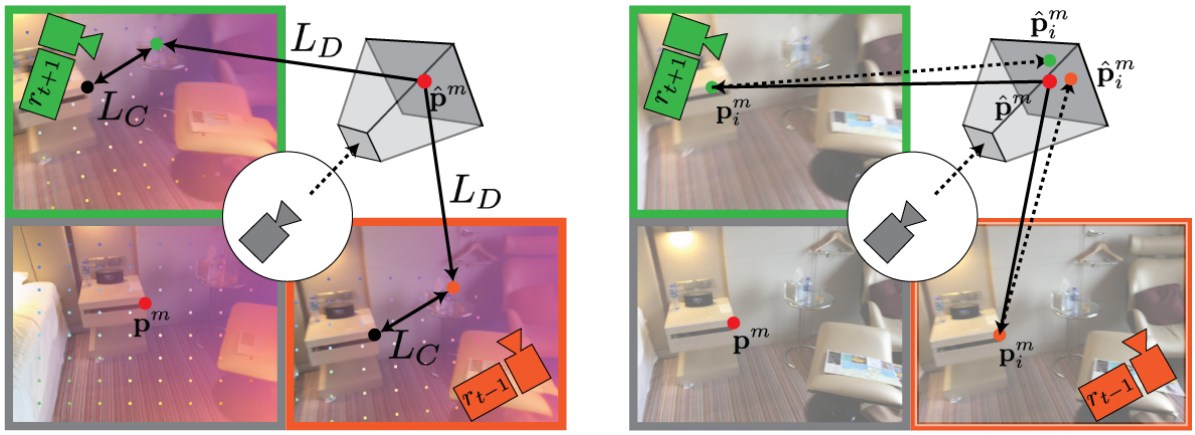
Optimization with RGB-D. With GT depthmap, the algorithm switches to the 3D scoring function $f_{i,j}^{\text{3D}}(\cdot)$. The depth adjustment is fixed to 1 and the 2D line-circle intersection becomes 3D line-sphere intersection.

6.3.1.3 Computational Complexity

Naive Time Complexity. From Eq. (6.2) and Fig. 6.4, in each epoch, we evaluate $(N-1)(K-1)$ pose groups with Hough Transform Acceleration. Suppose each group takes T iterations to optimize Eq. (6.19), the time complexity is:

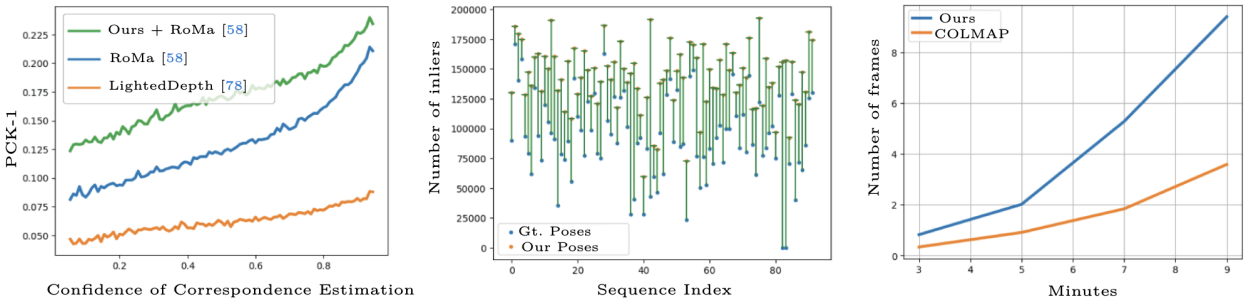
$$O((N-1)(K-1) \cdot N(N-1) \cdot (M+T)), \tag{6.20}$$

where each group computes $N(N-1)$ Hough matrices $\mathbf{H}$. Each matrix enumerates M sampled pixels, see Eq. (6.18). Maximizing Eq. (6.19) becomes indexing $\mathbf{H}$, hence has constant time complexity T, where $T << M$.



(a) Triangulation (b) Geometric Verification

Figure 6.7 **Triangulation** optimizes frustum RF for multiview consistency *w.r.t.* depth and correspondence. **Geometric Verification** inferences RF for sparse multiview consistent 3D points. For simplicity, in (a), we only plot L_c defined from the root frame.



(a) Self-supervised Correspondence (b) Certified Global Optimality (c) Run-time Comparison

Figure 6.8 **Ablation Studies** on the ScanNet.

Counting Unique Hough Matrices. Most computation is spent on Hough matrices. In Fig. 6.4, each connection in the computation graph suggests two unique Hough matrices. We minimize time complexity by only computing **unique** Hough matrices. In Fig. 6.4 first epoch, the initial optimal group $\overline{\mathcal{P}}^*$ has $N(N-1)$ matrices. Each ablated group only differs by one pose, hence introducing $2(N-1)(N-1)(K-1)$ matrices. The first-epoch complexity is then:

$$O^H(N(N-1)M + 2(N-1)^2(K-1)M) + O^{\text{BA}}(N(N-1)(K-1)T). \tag{6.21}$$

Only the Hough transform is accelerated. As $T << M$, the complexity of BA is neglectable. After the first epoch, $\overline{\mathcal{P}}^*$ only updates one pose per epoch, hence introducing $2(N-2)(K-1)$ matrices. The complexity for the rest epochs is,

$$O^H(2(N-2)(K-1)M) + O^{\text{BA}}(N(N-1)(K-1)T). \tag{6.22}$$

Dataset	Method	Density	$\delta_{0.5}$	δ_1	SI_{log}	A.Rel	S.Rel	RMS	RMS_{log}
ScanNet [52]	ZoeDepth [19]	9.1%	0.877	0.963	6.655	0.056	0.016	0.154	0.075
	⌟ Ours		**0.902**	**0.976**	**5.901**	**0.050**	**0.014**	**0.149**	**0.070**
	ZeroDepth [153]	5.6%	0.641	0.834	12.860	0.124	0.086	0.337	0.152
	⌟ Ours		**0.686**	**0.877**	**9.463**	**0.106**	**0.067**	**0.295**	**0.133**
	Metric3D [303]	2.6%	0.804	0.946	6.708	0.067	0.020	0.150	0.084
	⌟ Ours		**0.854**	**0.968**	**4.170**	**0.055**	**0.014**	**0.125**	**0.068**
KITTI360 [147]	ZoeDepth [19]	4.0%	0.677	0.899	14.154	0.103	0.490	3.521	0.153
	⌟ Ours		**0.719**	**0.910**	**13.220**	**0.094**	**0.474**	**3.499**	**0.145**
	ZeroDepth [153]	4.5%	0.584	0.844	16.468	0.132	0.819	3.486	0.183
	⌟ Ours		**0.654**	**0.877**	**13.881**	**0.115**	**0.772**	**3.395**	**0.164**
	Metric3D [303]	3.2%	0.846	0.958	9.226	0.072	0.508	2.194	0.104
	⌟ Ours		**0.860**	**0.963**	**8.896**	**0.068**	**0.487**	**2.139**	**0.101**

Table 6.1 **Self-Supervised Depth Estimation.** We apply self-supervision with 5 frames via executing the local SfM. We output improved sparse depthmaps over SoTA supervised inputs. The evaluation is conducted over the root frame.

While Eq. (6.22) has linear complexity, our method only updates one pose per epoch. Updating poses in all frames like other SfM methods is still quadratic.

6.3.2 Frustum Radiance Field Triangulation

Frustum Radiance Field. Now, we fix the optimized pose $\mathcal{P}^*$. Then we employ a frustum radiance field $\mathbf{V}$ of size $H \times W \times D$ for dense triangulation. Field $\mathbf{V}$ is defined over the root frame $\mathbf{I}_o$ and shares similarity with the categorical depthmap [79, 18]. We follow [277, 253] in rendering the depth d. The RGB estimation is skipped as unrelated. A 3D ray originated from pixel $\mathbf{p}_i$ at frame i is discretized into a set of 3D points and depth labels. With slight abuse of notation, we denote $\{\hat{\mathbf{p}}_{i,t} = \mathbf{o} + d_t\mathbf{r} \mid t \in [1, T]\}$, where $\hat{\mathbf{p}}$ is a 3D point, d_t is depth label and $\mathbf{r}$ is ray direction. Set integration interval $\delta_t = d_{t+1} - d_t$, depth d is:

$$d(\mathbf{p}_i) = \sum_t \alpha_t d_t, \ \alpha_t = T_t(1 - \exp(-\sigma_t\delta_t)), \ T_t = \exp(-\sum_{t' \in [1,t]} \sigma_{t'}\delta_{t'}). \tag{6.23}$$

We set the camera origin of frame i as $\mathbf{o}$. Instead of regressing occupancy δ with MLP [277, 253], we directly interpolate the radiance field $\mathbf{V}$:

$$\delta_t = \mathbf{V}(u, v, w), \text{where } \begin{bmatrix} u & v & w \end{bmatrix}^\intercal = \pi(\mathbf{E}, \hat{\mathbf{p}}_{i,t}). \tag{6.24}$$

Matrix $\mathbf{E}$ is the identity matrix. Function $\pi(\cdot)$ is projection function. Compared to using the MLP, frustum radiance field $\mathbf{V}$ is more computationally efficient [78].

Triangulation. Classic triangulation method [209] operates on a single 3D point. The RF provides additional constraints where all optimized points share a canonical 3D volume. In Fig. 6.7, we supervise $\mathbf{V}$ for multi-view consistency between dense depthmap $\mathcal{D}$ and correspondence map C. On depth:

$$L_D = \frac{1}{NM} \sum_i \sum_m \|\pi(\mathbf{P}_i, \hat{\mathbf{p}}^m) - d_i^m\|_1. \tag{6.25}$$

Here, $\hat{\mathbf{p}}^m$ is rendered from the root frame, following depth computed with Eq. (6.23). To apply correspondence consistency, we have:

$$L_C = \frac{1}{N(N-1)M} \sum_i \sum_{j,j\neq i} \sum_m \|\pi(\mathbf{P}_j, \hat{\mathbf{p}}_i^m) - \mathbf{q}_{i,j}^m\|_1, \tag{6.26}$$

where $\hat{\mathbf{p}}_i^m = \pi^{-1}(\mathbf{P}_i, \mathbf{p}_i^m, d(\mathbf{p}_i^m))$, $\mathbf{p}_i^m = \pi(\mathbf{P}_i, \hat{\mathbf{p}}^m)$. With slight abuse of notation, function $\pi(\cdot)$ returns depth for L_D, and location for L_C. We **always** first render from the root frame and subsequently project to N frames. From there, we project to other supported frames again, forming $N(N-1)$ pairs.

6.3.3 Geometric Verification

With the RF optimized, we apply geometric verification to acquire sparse multi-view consistent 3D points, as in Fig. 6.7:

$$C = \{\sum_{i,i\neq o} c_i^m \geq n^{\mathrm{c}}\},\ c_i^m = 1 \text{ if } \sum_{i,i\neq o} \|\hat{\mathbf{p}}_i^m - \hat{\mathbf{p}}^m\|_2 \leq \lambda^c. \tag{6.27}$$

We follow the same rendering process as training, where $\hat{\mathbf{p}}_i^m$ is computed with Eq. (6.26). First, we render 3D points from the root frame, project them to other views, and render 3D points from there again. A point is valid if a minimum of n^{c} views are consistent with the root.

6.4 Experiments

6.4.1 Self-supervised Depth Estimation

We benchmark whether self-supervision benefits supervised depth in unseen test data. For the correspondence estimator, we use PDC-Net [251]. For depth estimators, we adopt recently published in-the-wild depth estimator, including ZoeDepth [19], ZeroDepth [153], and Metric3D [303]. We evaluate with ScanNet [52] and KITTI360 [147] where all models perform zero-shot prediction.

Method	$\delta_{0.5}$	δ_1	SI_{log}	A.Rel	S.Rel	RMS	RMS_{log}
ZoeDepth [19]	0.658	0.894	**9.242**	0.104	0.039	0.255	0.128
⌟ Ours	**0.793**	**0.942**	**9.242**	**0.079**	**0.024**	**0.203**	**0.105**
ZeroDepth [153]	0.351	0.589	**20.145**	0.254	0.223	0.565	0.287
⌟ Ours	**0.490**	**0.725**	**20.145**	**0.199**	**0.156**	**0.457**	**0.237**
Metric3D [303]	0.533	0.753	**12.425**	0.216	0.339	0.495	0.228
⌟ Ours	**0.664**	**0.838**	**12.425**	**0.137**	**0.126**	**0.345**	**0.175**

Table 6.2 **Consistent Depth Estimation.** We measure the numerical improvement by aligning the support frame depthmaps to the root frame with our depth adjustment scalars. The evaluation is conducted on support frames on ScanNet [52].

Method	Train	Test	PCK-1	PCK-3	PCK-5	AEPE
PDC-Net [251]			0.119	0.511	0.743	4.612
⌟ LightedDepth [330]	M	S	0.061	0.341	0.563	6.590
⌟ Ours			**0.178**	**0.658**	**0.866**	**2.898**
RoMa [69]			0.144	0.583	0.815	3.333
⌟ LightedDepth [330]	S	S	0.066	0.359	0.588	5.974
⌟ Ours			**0.183**	**0.638**	**0.844**	**3.067**

Table 6.3 **Self-Supervised Correspondence Estimation.** We improve correspondence with RGB-D inputs, using metrics from [251]. The entry train and test are training and testing datasets of correspondence estimators. [Key: M=MegaDepth, S=ScanNet]

Test Data. In dense correspondence estimation, methods [331, 253, 251] output confidence score per correspondence. We follow [253, 251] to set a minimum threshold of 0.95. We run on ScanNet test split and it returns 92 sequences with sufficient correspondence. We form our test split by sampling 5 neighboring frames per valid sequence. Similarly, we run on KITTI360 data and randomly select 100×5 test split, *i.e.*, 100 sequences with 5 frames each. We consider it a comprehensive experiment. Similar to SPARF [253], our triangulation trains a NeRF-like structure. For reference, SPARF experiment on DTU dataset [118] includes only 15 sequences each with 3 images. In comparison, we include around 100 sequences.

Evaluation Protocols. We evaluate on **root** frame. We remove the scale ambiguity in the local SfM system to correctly reflect depth improvement. Specifically, we adjust all 5 depthmaps by an identical scalar computed between estimated root and GT depthmap, *i.e.*, the median scaling [90]. This eliminates scale ambiguity in the root frame while preserving it in support frames.

Results. In Tab. 6.1, our point cloud has a density of 2.6% – 9.1%, which amounts to **10 – 30**k points on a 480×640 image. On accuracy, we have **unanimous** improvement over all supervised

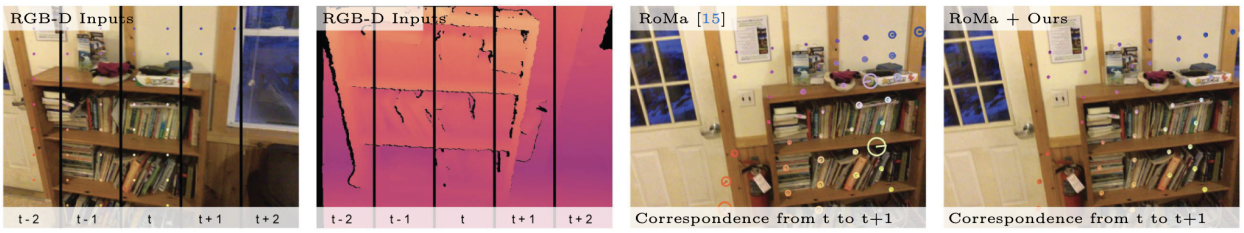


Figure 6.9 **Self-supervised Correspondence Estimation** enabled by our method with RGB-D inputs. The correspondence error is marked by the radius of the circle.

models of both datasets. Especially, we outperform strong baselines of ZoeDepth on ScanNet and Metric3D on KITTI360.

6.4.2 Consistent Depth Estimation

We evaluate on ScanNet. We follow Sec. 6.4.1 data split but evaluate the **support** frames. Temporal consistent depth is essential for AR applications [162]. Tab. 6.2 reflects the performance gain by aligning support frames to root with adjustments, which are jointly estimated with camera poses, see Fig. 6.2 and Fig. 6.3.

6.4.3 Self-supervised Correspondence Estimation

Real-world image correspondence label is expensive, *e.g.* KITTI provides only 200 optical flow labels. Existing datasets, such as MegaDepth and ScanNet, require large-scale 3D reconstruction with manual verification. Hence, correspondence estimators can not fine-tune on general RGB-D datasets like NYUv2 [222] or KITTI [83]. But our method enables self-supervised correspondence estimation on RGB-D data when using 3D scoring function Eq. (6.6). The camera poses are optimized with the point cloud specified by depthmap and correspondence. The accurate pose in turn improves projective correspondence. In Tab. 6.3, with 5 RGB-D frames, our method improves projective correspondence over inputs. We use the same test split as Sec. 6.4.1. The evaluation accumulates correspondence of each frame pair. Fig. 6.8a shows our improvement is **unanimous** over both confident and unconfident estimation. A visual example is in Fig. 6.9.

6.4.4 Sparse-view Pose Estimation

Comparison with Optimization-based and Learning-based Poses. Previous studies either evaluate two-view pose [240, 93], or SLAM-like odometry [273]. For more comparison, following

Frames	Method	Zero-shot	Suc. (%)	PCK-3	C3D-3	Rot.	Trans.
5	COLMAP [209]	✓	36.7	0.584	0.863	0.577	1.296
	Ours	✓	100.0	**0.727**	**0.904**	**0.422**	**1.062**
	DeepV2D [240] - ScanNet	✗		0.526	0.805	0.945	1.496
	DeepV2D [240] - NYUv2	✓		0.530	0.771	1.041	1.568
	DeepV2D [240] - KITTI	✓		0.125	0.387	4.908	4.231
	LightedDepth [330]	✓		0.651	0.832	0.469	1.550
	DRO [93] - ScanNet	✗	100.0	0.656	0.853	0.385	1.200
	DRO [93] - KITTI	✓		0.003	0.211	3.610	5.469
	DUSt3R [273] *w.o.* Intrinsic	✓		0.364	0.705	0.487	2.074
	DUSt3R [273] *w.t.* Intrinsic	✓		0.594	0.824	0.570	1.759
	Ours	✓		**0.799**	**0.900**	**0.368**	**1.120**

Table 6.4 **Sparse-view Pose Comparison** with optimization-based and learning-based methods. We only compare against COLMAP on its success sequences. Our method performs **zero-shot** testing on ScanNet while outperforming DeepV2D [240], DRO [93] with ScaNet [52] in training set. DUSt3R [273] trains on a similar dataset ScanNet++ [301].

Sec. 6.4.1 ScanNet split, we keep root frame and gradually add neighboring frames. In Tab. 6.4, LightedDepth [330] and ours both use PDC-Net [251] correspondence and ZoeDepth [19] mono-depth. COLMAP [209] uses PDC-Net correspondence. In evaluation, we follow [253] in aligning to GT poses. In Tab. 6.4, our **zero-shot** pose accuracy significantly outperforms all prior arts, including [273, 93, 240] with ScanNet [52] or ScanNet++ [301] in their training set. See Supp. for complete comparison from 3 to 9 frames. In Fig. 6.8, we attribute our superiority to certified global optimality over robust measurements.

Comparison with NeRF-based Poses. Sparse view NeRF methods optimize NeRF jointly with camera poses, mandating a sophisticated and time-consuming optimization scheme. *E.g.*, SPARF [253], takes one day to optimize the pose and NeRF. Typically, their poses are initialized with COLMAP. Our method provides an alternative initialization with superior performance. In Tab. 6.5, our initialization achieves better or on-par pose performance than SoTA [253] while only taking ~3 minutes (Fig. 6.7). Our lower performance on Replica dataset might be due to ZoeDepth not being trained on synthetic data. Our work suggests the straightforward "first-pose-then-NeRF" scheme also applies to short videos.

Certified Global Optimality . In Fig. 6.8b, our Bundle-RANSAC-Adjustment **always** finds more inliers than groundtruth poses. To our best knowledge, we are the **first** work that extends RANSAC to a multi-view system.

Method	Frames	LLFF [214]		Replica [231]	
		Rot.	**Trans.**	**Rot.**	**Trans.**
BARF [148]	3	2.04	11.6	3.35	16.96
RegBARF [148, 180]		1.52	5.0	3.66	20.87
DistBARF [148, 11]		5.59	26.5	2.36	7.73
SCNeRF [119]		1.93	11.4	0.65	4.12
SPARF [253]		<u>0.53</u>	<u>2.8</u>	**0.15**	**0.76**
Ours		**0.46**	**1.9**	<u>0.52</u>	<u>4.09</u>

Table 6.5 **Sparse-view Pose Comparison** with NeRF-based methods following [253].

Run-time. In Fig. 6.8c, we run approximately 3× slower than COLMAP. But both have quadratic complexity. With 3/5/7/9 frames, we take 0.8/2.0/5.3/9.4 minutes on RTX 2080 Ti GPU, while COLMAP uses 0.3/0.9/1.8/3.6 minutes on Intel Xeon 4216 CPU. COLMAP runs sequentially. But our method is highly parallelized. Our core operation Hougn Transform scales up with more GPUs.

6.5 Conclusion

By revisiting self-supervision with local SfM, we first show self-supervised depth benefits SoTA supervised model with only 5 frames. We have SoTA sparse-view pose accuracy, applicable to NeRF rendering. We have diverse applications including self-supervised correspondence and consistent depth estimation.

Limitation. The NeRF-like triangulation constrains our method from applying to large-scale self-supervised learning. Its efficiency requires improvement.

CHAPTER 7

MOTION-FROM-STRUCTURE: LEVERAGING MONOCULAR DEPTH PRIORS FOR MULTI-VIEW TASKS

Structure-from-Motion (SfM) is a classical 3D vision task for recovering camera parameters and scene geometry from multi-view images. Recent advances in deep learning and vision foundation models have led to more robust monocular depth estimation (MDE) models that can directly predict structure from a single image without relying on camera motion. However, using MDE in SfM remains challenging due to its high error variance and the need for affine corrections. While prior works have incorporated MDE into SfM pipelines, it is generally used only to initialize sparse keypoints, discarding most of its dense predictions. In this paper, we introduce the notion of Motion-from-Structure (MfS), which fully leverages the density of monocular depth priors to infer camera motion. By reformulating bundle adjustment to distinguish inlier and outlier depth pixels, we eliminate the need for per-pixel adjustments and offer a plug-and-play method that integrates seamlessly with arbitrary MDE models. We show the efficacy of our approach on multi-view tasks, including pose estimation, structure-from-motion, and camera re-localization. Our method achieves state-of-the-art results on camera pose estimation, efficiently scaling to thousands of frames and highlighting the potential of MDE for multi-view tasks.

7.1 Introduction

Structure-from-Motion (SfM) is a cornerstone of 3D computer vision for estimating camera intrinsics and extrinsics from image collections. Its versatility has fueled applications across diverse domains, including 3D reconstruction [81], neural rendering [170], camera re-localization [86], and robot navigation [96]. Traditional SfM methods [211] operate by jointly optimizing camera motion and 3D point positions, relying on sparse feature correspondences. However, these methods often struggle with scenes lacking sufficient texture or with large baseline motions, leading to potential degeneracy and inaccurate results.

The advent of deep learning has revolutionized monocular depth estimation (MDE) [190, 19], enabling the direct inference of dense depth maps or point clouds from single images, indepen-

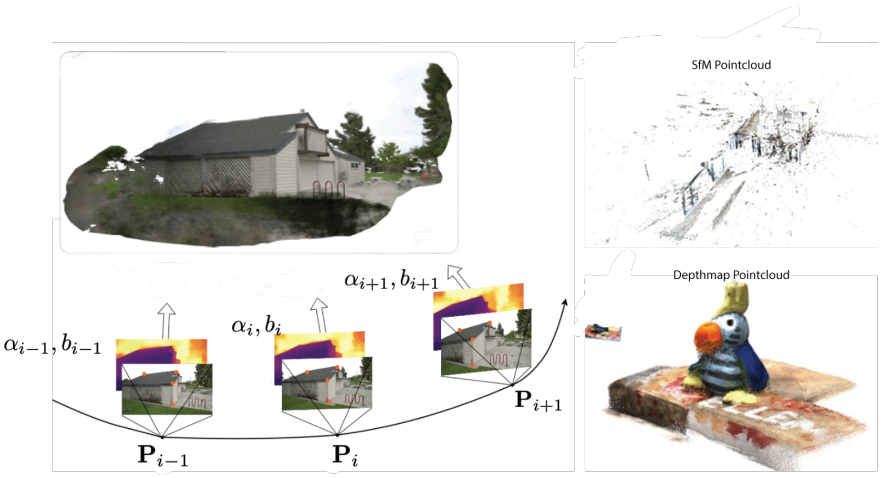


Figure 7.1 **Motion-from-Structure from Monocular Depth.** (Left) We directly estimate camera parameters given monocular depthmaps while jointly optimizing affine depth corrections. Unlike methods that use depthmaps for SfM initialization, our method avoids per-pixel adjustments and network fine-tuning, extending arbitrary monocular depth estimation models to multi-view tasks. (Right) We challenge whether SfM triangulation consistently improves monocular depth, particularly with limited motion parallax and scene texture. We evaluate a "lower-bound" approach side-stepping SfM triangulation by relying on robust monocular networks, and found it performs surprisingly well.

dently of camera motion. This rich structural prior was shown to benefit various downstream applications [51, 215]. However, leveraging MDE for multi-view tasks received less attention. While some recent works [26, 20, 227, 66] have explored integrating MDE into SfM pipelines, they typically use it to initialize sparse keypoints, discarding its dense predictions and relying heavily on refinement with traditional bundle adjustment.

The performance of SfM-derived point clouds can be scene-dependent, sometimes failing to surpass the quality of monocular depth maps (Figure 7.1). This observation motivates our "Motion-from-Structure" approach, which aims to leverage the dense structural information provided by MDE to directly recover camera motion, effectively side-stepping the triangulation step [102, 12]. This approach has several key advantages: it establishes a robust "lower bound" for pose estimation, mitigating degeneracy issues inherent in traditional SfM; and it effectively aligns individual monocular depth maps into a coherent 3D scene representation. Unlike prior methods that rely on per-pixel depth adjustments and network parameter fine-tuning, our method offers a plug-and-play solution that can seamlessly integrate with any MDE model (see Table 7.1).

Optimization	Ace-Zero [26]	FlowMap [227]	VGGSfM [267]	MASt3R-SfM [66]	Ours
Network Forward	✓	✓	✗	✗	✗
Network Backward	✓	✓	✗	✗	✗
Pixel-wise Depth	✓	✓	✓	✓	✗

Table 7.1 Our method relies solely on depthmap pointcloud without adjusting network parameters and pixel-wise depth values.

A key challenge of utilizing monocular depth maps in conventional SfM lies in adapting methods optimized for sparse, accurate point clouds to leverage the the dense, high error estimates of MDE. Prior methods [267, 227, 66, 26] pre-select accurate depth pixels through neural guidance, which involves training a network to predict noise measurements. Neural guidance, while effective, still requires optimizing network parameters during bundle adjustment, leading to increased memory consumption and hindering scalability [227, 66, 186]. A second challenge in aligning independent depth maps for multi-view images is the necessity of optimizing an affine depth correction per image [20, 308].

To estiamte camera parameters with dense but noisy depth maps, while jointly optimizing the required affine depth correction, we use a robust inlier-counting score inspired by RANSAC [77]. Our bundle adjustment maximizes the projective inliers between depth and correspondence maps. To address the non-differentiability and threshold sensitivity associated with inlier counting, we compute inliers across all thresholds, transforming the discrete RANSAC process into a continuous cumulative distribution function (CDF) [9]. This allows us to naturally represent the noise measurement of each depth pixel as a probability derived from the CDF and its corresponding projective residual, resulting in a smooth, differentiable, and robust optimization.

Our proposed projective inlier function is flexible and compatible with robust loss functions from prior work [211], offering a plug-and-play framework that extends arbitrary monocular depth networks to large-scale multi-view 3D vision tasks. Our main contributions are three-fold:

1. A novel bundle adjustment algorithm that can efficiently handle the high noise and affine ambiguities of dense monocular depth maps. (Sec. 7.3.2)

2. An effective SfM framework successfully leveraging arbitrary MDE to multi-view 3D vision

tasks. (Sec. 7.3.3)

3. State-of-the-art performance in camera pose estimation and re-localization across multiple datasets. (Sec. 7.4)

7.2 Related Work

Foundation Models in Multiview 3D Vision. Efforts to develop foundation models for monocular depthmap estimation [190] and binocular correspondence estimation [70, 251] have been ongoing. Pioneering studies [274, 140, 315] have unified monocular and binocular tasks within a binocular pointmap estimation framework. They demonstrate its potential for tackling multi-view 3D vision challenges, including camera extrinsic and intrinsic estimation. Their formulation nevertheless includes an optimization process to convert the dense network prediction to low DoF camera parameters. Our method benefits them with an enhanced optimization objective function specifically designed for dense and high-variance deep network outputs. Beyond that, our work encourages the community to reconsider the merits between depthmap and pointcloud network as monocular depth networks [190] show equal performance with pointcloud network [66].

RANSAC. RANdom Sample Consensus (RANSAC) algorithms [9] aim at robust low-DoF parameter estimation in the presence of noisy data. Our work similarly handles noisy input as consuming high-variance network predictions instead of an accurate sparse point cloud. Several RANSAC works [246, 9] focus on improving the scoring function via generalizing from binary [77] to continuous values. Among them, MAGSAC [9] can be considered a special case of our algorithm with an added assumption that residuals follows a truncated chi-squared distribution. Unlike [9], we leverage dense predictions, and specifically the induced residual distribution, from pre-trained monocular depth models to derive an improved scoring function.

SfM with Deep Learning. There have been several pioneering works combine deep learning with SfM [267, 66, 26]. [26, 227] include the network backpropagation during SfM Bundle-Adjustment. VGGSfM [267] instead formulates SfM BA as a network forward process. However, due to higher computational complexity, both strategies either limit the network size or the scale of SfM. Our

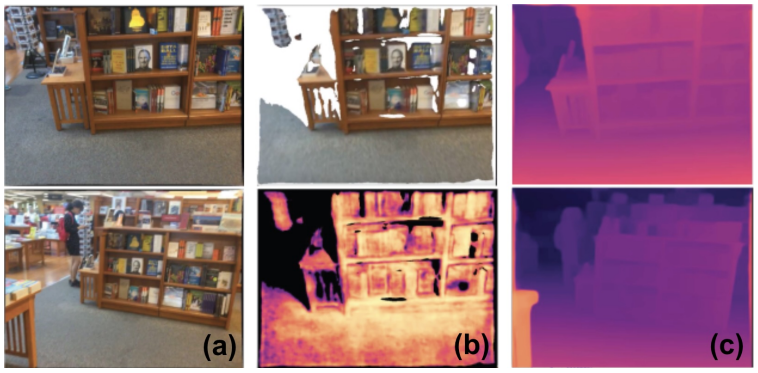


Figure 7.2 Given (a) input frames, our method consumes their (b) dense correspondence with confidence scores and (c) depthmaps.

method takes a different approach by decoupling network inference from SfM BA. This enables our method to benefit ongoing developments of vision foundation models for the SfM process. Finally, unlike [66], our method supports any monocular depth network beyond MASt3R. Our method highlights the potential of leveraging monocular networks for the SfM problem. Note, despite our similar name to [263], we address a different problem.

7.3 Method

To leverage depth priors for multi-view tasks, we initialize depthmaps and dense correspondence maps using a monocular depth estimator (i.e. , ZoeDepth [19]) and a binocular correspondence estimator (i.e. , RoMa [70]). An example is in Fig. 7.2. We sub-sample the dense correspondence map into points to initialize two-view odometry. Our method performs hierarchical Bundle Adjustment, starting from a coarse stage (Sec. 7.3.3.2) to a fine stage (Sec. 7.3.3.3). Fig. 7.3 compares our algorithm to conventional SfM pipelines.

7.3.1 Overview

Problem Definition. Given as input an unordered collection of N frames $\{I_i\}_{i\in[N]}$, we optimize for camera intrinsics $\mathcal{K} = \{\mathbf{K}_i\}$ and extrinsics $\mathcal{P} = \{\mathbf{P}_i\}$. Using a pre-trained depth network $\mathcal{N}_\mathbf{D}$ and a correspondence network $\mathcal{N}_\mathbf{C}$, we extract N depthmaps $\mathcal{D} = \{\mathbf{D}_i = \mathcal{N}_\mathbf{D}(I_i)\}$ and pairwise correspondence maps $\mathcal{C} = \{\mathbf{C}_{i,j} = \mathcal{N}_\mathbf{C}(I_i, I_j), i \neq j\}$. We jointly optimize per-frame affine corrections $\mathcal{A} = \{\alpha_i, b_i \mid i \leq M\}$, producing aligned depth maps $\mathbf{D}'_i = \alpha_i \cdot \mathbf{D}_i + b_i$.

Optimization. Let $\mathcal{X} = \{\mathcal{P}, \mathcal{K}, \mathcal{A}\}$ denote the set of all variables to optimize, and $\mathbf{X}_i =$

Algorithm 7.1 SfM Pipeline

Input: Image set
Output: Camera poses and 3D points
Sparse correspondence
Two-view geometry estimation
Incremental SfM:
    - Iterative register new images
    - Triangulate new 3D points
    - Bundle adjustment to refine structure and poses
Final Optimization: Global bundle adjustment
Output: Optimized poses and 3D points

Algorithm 7.2 MfS Pipeline

Input: Image set
Output: Camera poses
Dense depth & correspondences
Two-view geometry estimation
Coarse Stage:
    - Optimize sub-graph log-CDF scores.
Fine Stage:
    - Active Sampling.
    - Optimize global Euclidean CDF scores.
Output: Optimized poses

Figure 7.3 Comparison of conventional SfM pipelines, *e.g.*, COLMAP [211], to the proposed MfS approach.

$\left(\mathbf{P}_i, \mathbf{K}_i, \mathbf{A}_i\right)$. We formulate this optimization as maximizing a scoring function $\mathcal{S}$

$$\mathcal{X}^* = \underset{\mathcal{X}=(\mathcal{P},\mathcal{K},\mathcal{A})}{\arg\max}\ \mathcal{S}(\mathcal{X} \mid \mathcal{D}, \mathcal{C}). \tag{7.1}$$

We define $\mathcal{S}$ as a summation of a suitable quality function Q over frame pairs (I_i, I_j) in the pose graph $\mathcal{G}$ (Sec. 7.3.3.1)

$$\mathcal{S}(\mathcal{X} \mid \mathcal{D}, \mathcal{C}) = \frac{1}{M} \sum_{(i,j)\in\mathcal{G}} Q(\mathbf{X}_i, \mathbf{X}_j \mid \mathbf{D}_i, \mathbf{D}_j, \mathbf{C}_{i,j}), \tag{7.2}$$

where $M \gg 1$ is the number of sampled correspondences.

Three-Stage SfM Pipeline. Our SfM runs in three stages: (1) initialization, (2) coarse-stage SfM, and (3) fine-stage SfM. The coarse-stage SfM focuses on robustness by roughly aligning images by randomly sampling depth and correspondence pixels. The fine-stage SfM refines camera poses, prioritizing pixels with lower reprojection errors. The following sections begin with the core of our algorithm: the Bundle-Adjustment process for inlier-outlier separation, followed by detailed discussion of initialization, coarse-stage SfM, and fine-stage SfM.

7.3.2 Separate Inliers from Outliers in BA

Motivation. As the reader may notice, in Sec. 7.3.1, we use the term “maximizing the scoring function” instead of the more common “minimizing a loss function" found in other SfM litera-

ture [186]. This choice emphasizes our connection to RANSAC methods, as both approaches focus on optimizing low DoF camera parameters from densely noisy inputs. Specifically, we assume dense monocular depthmaps contain sufficient inliers to support camera localization, despite mixed with outliers. Thus, the Bundle-Adjustment is designed to maximize inliers from dense depthmaps. After presenting the preliminaries, this subsection starts with a naive yet robust binary scoring function. However, the non-differentiability of the binarized function poses challenges for Bundle-Adjustment. To address this, we generalize it to a smooth form by leveraging depthmap density.

Sampling Depth and Correspondence. We only consider pairs of frames (I_i, I_j) with a co-visibility score at least ν, defined as the percentage of pixels visible in both frames. For each co-visible frame pair, we downsample the dense full-resolution depthmaps and correspondence maps to a fixed number of pixels κ. Specifically, between frame i and j, we sample κ depth pixels on frame i and κ i-to-j correspondence pixels. We only sample correspondences with a confidence score at least χ.

Projective Residuals. We define the residual $r_{i,j,k}$ as the 2D discrepancy in the k^{th} sampled correspondence $c_{i,j,k} \in \mathbf{C}_{i,j}$. Denoting $c_{i,j,k}$ as $(p_{i,j,k}, q_{i,j,k}) \in I_i \times I_j$, we write

$$r_{i,j,k} = \left\| \pi_{i \to j}\left(\mathbf{D}'_i[p_{i,j,k}]\right) - q_{i,j,k} \right\|_2, \tag{7.3}$$

where the operator $\pi_{i \to j}$ projects the pixel $p_{i,j,k}$ in frame I_i, with its corrected depth value in $\mathbf{D}'_i$, to frame I_j. The projection is defined by the camera intrinsics $\mathbf{K}_i$, $\mathbf{K}_j$ and extrinsics $\mathbf{P}_i$, $\mathbf{P}_j$ [99]. Other robust norms may also be used in Eq. (7.3), *e.g.*, the Cauchy function used in [211].

Residuals to Binary Scoring Function. Given a residual threshold τ, we realize Eq. (7.2) by setting $Q := Q^{\text{b}}_{\tau}$ where

$$Q^{\text{b}}_t(\mathbf{X}_i, \mathbf{X}_j \mid \mathbf{D}_i, \mathbf{D}_j, \mathbf{C}_{i,j}) = \textstyle\sum_k \mathbb{1}[r_{i,j,k} < \tau], \tag{7.4}$$

where $\mathbb{1}(\cdot)$ is the indicator function. Intuitively, a depth pixel is considered an inlier if its projective residual is below the threshold τ. The binarized scoring function in Eq. (7.4) is widely used in RANSAC algorithms [77] for its superiority in managing densely noisy inputs. However, the

RANSAC algorithm is mostly applied to problem of low Degree-of-Freedom (DoF), *e.g.*, essential and fundamental matrix estimation [181, 100]. In contrast, the multi-view pose estimation problem has a significantly larger solution space. This necessitates a continuous scoring function to enable first-order and second-order optimization methods.

Binary Scoring Function to CDF. The dense depthmaps provide enough samples of projective residuals to utilize their distributional properties, leveraging the deep priors of the pre-trained MDE model. Letting R denote the set of all residuals at the current epoch, we model the residual r as a random variable following an empirical distribution $\mathcal{R}$ we obtain by kernel density estimation (KDE) [223, 126], i.e.,

$$r \sim \mathcal{R} = \mathrm{KDE}(R), R = \{r_{i,j,k} \mid (i,j) \in \mathcal{G}, k \in [\kappa]\} \tag{7.5}$$

Taking inspiration from MAGSAC [9], we smooth out the binary scoring function Eq. (7.4) with a threshold τ as:

$$\begin{aligned} \mathcal{S}_\tau(\mathcal{X} \mid \mathcal{D}, C) &= \frac{1}{M} \sum_{i,j,k} \mathbb{1}(r_{i,j,k} < \tau) \\ &\approx 1 \cdot \int_0^\tau p(r)\, \mathrm{d}r + 0 \cdot \int_\tau^{+\infty} p(r)\, \mathrm{d}r = F(\tau), \end{aligned} \tag{7.6}$$

where $p(r)$ and $F(\tau) = \Pr[r < \tau]$ are the probability and cumulative distribution function (CDF) of $\mathcal{R}$, respectively.

Beyond Binary Scoring Function. Dense depthmaps contain depth pixels with varying noise levels. Intuitively, a large threshold τ in Eq. (7.6) encourages to register camera at an approximately correct location. A small threshold τ in Eq. (7.6) improves accuracy but risks local minima. To fully leverage dense depthmaps, we extend scoring function Eq. (7.6) beyond a single threshold by integrating up to a maximum $\tau_{\max}$ as:

$$\mathcal{S}(\mathcal{X} \mid \mathcal{D}, C) = \int_0^{\tau_{\max}} p(t) \cdot \mathcal{S}_t(\mathcal{X} \mid \mathcal{D}, C)\, \mathrm{d}t. \tag{7.7}$$

Intuitively, Eq. (7.7) extends Eq. (7.6) by summing over infinitely many thresholds. Crucially, thresholds are sampled according to the natural residual distribution $\mathcal{R}$ induced by the rich depth

Algorithm 7.3 Forward and Backward BA Scoring Function

1: $\mathcal{R} = \{p,\ F\} := \mathrm{KDE}(R)$ ▷ smooth score
2: $\mathcal{S} = \frac{1}{M} \sum_{i,j,k} F(r_{i,j,k}) \cdot \mathbb{1}[r_{i,j,k} < \tau_{\max}]$ ▷ forward
3: $\frac{\partial}{\partial \mathbf{x}} \mathcal{S} = \frac{1}{M} \sum_{i,j,k} p(r_{i,j,k}) \cdot \frac{\partial}{\partial \mathbf{x}} r_{i,j,k}$ ▷ backward

priors from the pre-trained MDE model. Formally, the proposed scoring function is:

$$\mathcal{S}(\mathcal{X} \mid \mathcal{D}, C) = \frac{1}{M} \sum_{i,j,k} F(r_{i,j,k}) \cdot \mathbb{1}[r_{i,j,k} < \tau_{\max}]. \tag{7.8}$$

Distinguishing Inliers from Outliers. From Eq. (7.8), the BA process naturally differentiates inliers from outliers by assigning higher values to depth pixels with smaller residuals while down-weighting those with larger residuals. Fig. 7.4 illustrates how the BA process differentiates inliers from outliers. Further, Eq. (7.8) inherits the robustness. For instance, applying Eq. (7.8) to update the example variable $\mathbf{x}$, e.g., camera rotation component, by computing its gradient:

$$\frac{\partial}{\partial \mathbf{x}} F(r_{i,j,k}) = p_r(r_{i,j,k}) \cdot \frac{\partial}{\partial \mathbf{x}} r_{i,j,k}, \tag{7.9}$$

where gradient of extreme residual values, i.e., those with low probability, is suppressed. Finally, after optimization, the noise level of a depth pixel is represented by its residual's probability. Algorithm 7.3 provides a succinct summary of the proposed Eq. (7.7) scoring function.

Scalability. Our approach is highly parallelizable, making it suitable for large-scale SfM, thanks to its efficient data structure consisting of simple sets of depth and correspondence pairs. This is in contrast to traditional approaches requiring full 3D point clouds, which introduces complex for parallel processing, and more recent methods which run out of memory upon processing upwards of 200 views, e.g., FlowMap [227] and VGGSfM [267] as reported in [66].

7.3.3 SfM Pipeline

The subsection outlines the proposed SfM process, including initialization, coarse-stage SfM, and fine-stage SfM.

7.3.3.1 Initialization

Pose Graph. We construct a weighted undirected graph $\mathcal{G}$ using correspondence maps C. Each edge $g_{i,j} \in \mathcal{G}$ is defined as the visibility between frame i and j, *i.e.*, the percentile of pixels visible

in both frames.

Intrinsic Initialization. In each frame, we use [274] to extract the dense pointcloud estimation. Next, the dense pointcloud is converted to an incidence field, where we apply the RANSAC intrinsic calibration method proposed in [328]. If a shared intrinsic is assumed for the input image collection, we initialize it as the median.

Extrinsic and Depth Adjustments Initialization. We adopt a greedy strategy to initiate a spanning tree from the pose graph $\mathcal{G}$. The root node is chosen as the one with the highest degree. A new node is added such that it maximizes the total degree of the graph after its inclusion. *E.g.*, when frame i is added, its extrinsic $\mathbf{P}_i$ and depth scale adjustment s_i are simultaneously initialized. The depth bias adjustment b_i is initialized to 0.

7.3.3.2 Coarse-Stage SfM

Logged Residual. As in Fig. 7.5, coarse stage prioritizes to register frames with an approximately correct location to avoid local minimum. We apply logarithm operation to the L2 norm residual in Eq. (7.3) to enhance robustness:

$$r^{1}_{i,j,k} = \log(1 + r_{i,j,k}). \tag{7.10}$$

Graph Decomposition. Suppose the frame i is poorly registered, its corresponding residual $r_{i,j,k}$ exhibits significantly large values. Due to the robustness property of Eq. (7.8), the residuals with larger values are automatically assigned lower weights and smaller gradients. These characteristics cause poorly registered frames to become "stuck" in a local minimum. We propose a graph decomposition strategy to mitigate the occurrence of early local minima. For the graph $\mathcal{G}$, we decompose it into a N subgraphs $\mathcal{G}_i$:

$$\mathcal{G} = \sum_{i \in N} \mathcal{G}_i, \quad \mathcal{G}_i = \{\mathcal{X}_i, \ \mathcal{E}_i\}, \tag{7.11}$$

where $\mathcal{X}_i = \{\mathbf{I}_i\} \cup \mathcal{N}(\mathbf{I}_i)$, and $\mathcal{E}_i = \{(\mathbf{I}_i, \mathbf{I}_j) \mid \mathbf{I}_j \in \mathcal{N}(\mathbf{I}_i)\}$. Each subgraph $\mathcal{G}_i$ is a directed graph, includes the i-th frame $\mathbf{I}_i$ and its neighbouring frames $\mathcal{N}(v_i)$. Correspondingly, the Eq. (7.8)

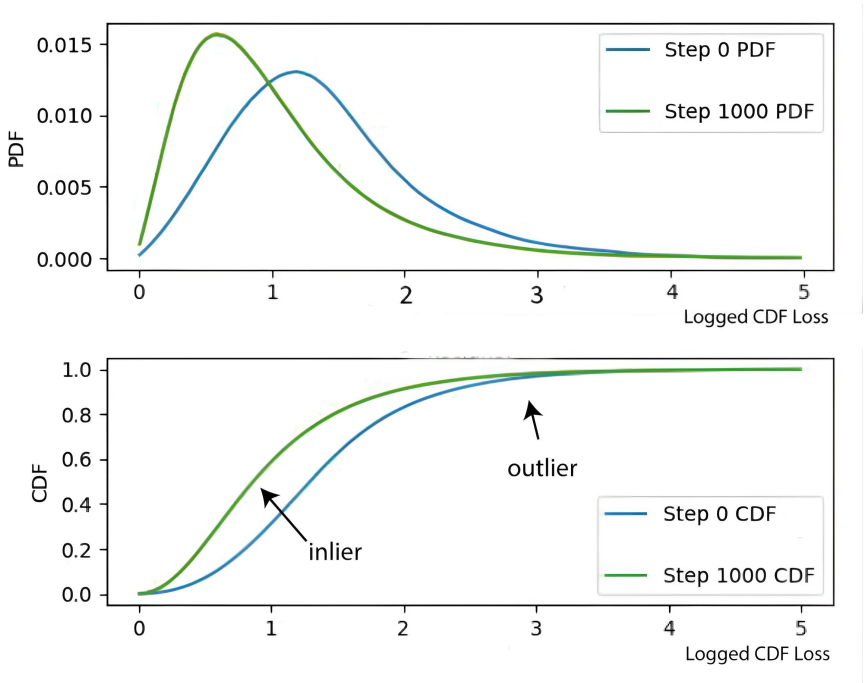


Figure 7.4 **Distinguishing Inliers from Outliers** with Bundle-Adjustment: Distributions of projective residuals before and after BA show residuals shifting towards zero, indicating the system is selecting more inliers.

scoring function is formulated as:

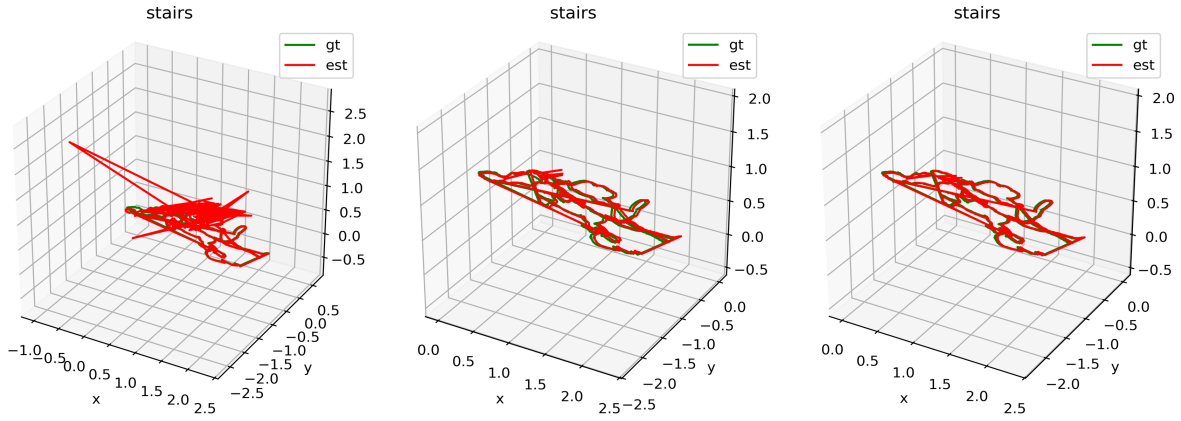
$$\mathcal{S}^d(\mathcal{P},\mathcal{K},\mathcal{A} \mid \mathcal{D},C) = \frac{1}{N}\sum_{i,j,k}\phi^{\mathrm{c}}(r^{\mathrm{l}}_{i,j,k}), \tag{7.12}$$

where $\phi^{\mathrm{c}}(r^{\mathrm{l}}_{i,j,k}) = F_{r_i}(r^{\mathrm{l}}_{i,j,k})$, and $r^{\mathrm{l}}_i \sim \mathcal{S}(\mathcal{R}_i)$. For each logged residual $r^{\mathrm{l}}_{i,j,k}$ from frame $\mathbf{I}_i$, we obtain its CDF using the distribution computed only with the subgraph $\mathcal{G}_i$.

7.3.3.3 Fine-Stage SfM

From Random to Active Sampling. Our method assumes accurate 3D pointclouds from depthmap estimation for intrinsic and extrinsic calibration. However, the random sampling strategy in Sec. 7.3.3.1 still includes noisy depth pixels. While the robust scoring function Eq. (7.8) suppresses noisy pixels, actively sampling accurate ones could further improve performance. Therefore, in the fine-stage SfM, we prioritize depth pixels with smaller residuals. First, we accumulate pair-wise residuals as follows:

$$f(d_{i,m}) = \frac{1}{\|\mathcal{N}(\mathbf{I}_i)\|}\sum_{j\in\mathcal{N}(\mathbf{I}_i)} r_{i,j,k}. \tag{7.13}$$



(a) Initialization (b) Coarse Stage (c) Fine Stage

Figure 7.5 **Hierarchical Bundle Adjustment (BA).** We visualize our coarse-stage (Sec. 7.3.3.2) and fine-stage (Sec. 7.3.3.3) BA process using the 7-Scenes dataset [219] - sequence "Stairs". With a poor initialization, the coarse-stage Bundle Adjustment registers camera poses to an approximately correct location. Then, the fine-stage optimization further improves pose accuracy.

Scene	COLMAP [210]		ACE-Zero [26]		FlowMap [227]		VGGSfM [267]		DF-SfM [105]		MASt3R-SfM [66]		MfS (Ours)	
	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA
courtyard	56.3	60.0	4.0	1.9	7.5	3.6	50.5	51.2	80.7	74.8	89.8	64.4	94.7	94.7
delivery area	34.0	28.1	27.4	1.9	29.4	23.8	22.0	19.6	82.5	82.0	83.1	81.8	83.1	83.0
electro	53.3	48.5	16.9	7.9	2.5	1.2	79.9	58.6	82.8	81.2	100.0	95.5	95.6	78.2
facade	92.2	90.0	74.5	64.1	15.7	16.8	57.5	48.7	80.9	82.6	74.3	75.3	100.0	99.2
kicker	87.3	86.2	26.2	16.8	1.5	1.5	100.0	97.8	93.5	91.0	100.0	100.0	100.0	98.9
meadow	0.9	0.9	3.8	0.9	3.8	2.9	100.0	96.2	56.2	58.1	58.1	58.1	100.0	58.1
office	36.9	32.3	0.9	0.0	0.9	1.5	64.9	42.1	71.1	54.5	100.0	98.5	100.0	86.2
pipes	30.8	28.6	9.9	1.1	6.6	12.1	100.0	97.8	72.5	61.5	100.0	100.0	100.0	96.7
playground	17.2	18.1	3.8	2.6	2.6	2.8	37.3	40.8	70.5	70.1	100.0	93.6	94.7	93.8
relief	16.8	16.8	16.8	17.0	6.9	7.7	59.6	57.9	32.9	32.9	34.2	40.2	100.0	98.9
relief 2	11.8	11.8	7.3	5.6	8.4	2.8	69.9	70.3	40.9	39.1	57.4	76.1	100.0	98.9
terrace	100.0	100.0	5.5	2.0	33.2	24.1	38.7	29.6	100.0	99.6	100.0	100.0	100.0	100.0
terrains	100.0	99.5	15.8	4.5	12.3	13.8	70.4	54.9	100.0	91.9	58.2	52.5	100.0	95.4
Average	49.0	47.8	16.4	9.7	10.1	8.8	65.4	58.9	74.2	70.7	81.2	79.7	**97.5**	**90.9**

Table 7.2 **Multi-view pose estimation** benchmark on ETH3D dataset [213, 212] in terms of RRA (@5) and RTA (@5). (sparse-set SfM)

Eq. (7.13) calculates the average residual of each depth pixel across its connected frame pairs. Next, we employ a Non-Neighborhood Suppression strategy, similar to Non-Maximum-Suppression (NMS) in detection literature. We begin by sampling depth pixels with the smallest residuals, excluding their neighbors as each is selected. In summary, the proposed active sampling utilizes depthmap density to approximate the triangulation process in classic SfM literature. We assume that depth pixels within a spatial neighborhood inherently capture its variance.

Residual and Pose Graph. We change the Eq. (7.3) residual to its simple L2 norm. Meanwhile, we define the pose graph to include all images as in Eq. (7.8). Also see Fig. 7.5.

7.4 Experiments

We demonstrate the efficacy of our method through evaluations on two fundamental 3D vision tasks: structure-from-motion (SfM) and camera re-localization.

7.4.1 Datasets

We distinguish two types of SfM datasets, selecting a representative of each. We denote *sparse-set* datasets as those with minimal visual overlap between frames, selecting the ETH3D dataset [213, 212], following MASt3R-SfM [66]. We denote *dense-set* datasets as those with high amounts of visual overlap between frames, typically present in video sequences with hundreds to thousands of frames. Due to their scale, dense-set data poses significant challenges for traditional feature-matching approaches. Here we select the ScanNet dataset [53]. Its ground-truth odometry enables direct comparison with COLMAP [210] for both calibrated and uncalibrated camera localization settings. From the 100 ScanNet test sequences, we sample at 5 FPS then select the 71 sequences where the frames do not exceed 2500, ensuring that preprocessing remains manageable and COLMAP [210] runs successfully. For camera re-localization, we use the standard 7-scenes dataset [219] following the protocol of marepo [41].

7.4.2 Implementation Details

We parameterize camera pose with a 9D rotation matrix following SPARF [253]. Across experiments, for the coarse-stage sub-graph CDF scoring function, we include pixels with a reprojection error smaller than exp(15) in L2-norm. In the fine-stage CDF scoring function, we include pixels with a reprojection error below 20 for ScanNet and 7-Scenes, and below 35 for ETH3D to accommodate its high-resolution images. We use the Adam optimizer [127] for $50,000$ iterations with a learning rate of $1e^{-4}$. Within each pair of frames, we sample $\kappa = 300$ pixels. The range of Neighborhood Suppression strategy in active sampling is set to $\mathcal{N}(\mathbf{I}) = 8$. We exclude an image pair if less than $\nu < 15\%$ of its pixels are co-visible. During pre-processing, we sample only from the dense correspondence map where the confidence scores exceed $\chi > 0.3$.

Method	Depth	Corres.	Calibrated			Uncalibrated		
			Acc@3°	Acc@5°	Acc@10°	Acc@3°	Acc@5°	Acc@10°
COLMAP [210]	-	SuperPoint [63]	0.398	0.589	0.783	0.342	0.505	0.670
Ours	ZoeDepth [19]	RoMa [70]	0.396	0.614	0.823	0.372	0.586	0.811
	DUSt3R [274]	RoMa [70]	0.426	0.631	0.830	0.403	**0.615**	0.820
	UniDepth [190]	RoMa [70]	0.432	0.636	0.833	**0.407**	0.612	**0.823**
	DUSt3R [274]	MASt3R [140]	0.432	0.639	0.837	0.384	0.596	0.811
	UniDepth [190]	MASt3R [140]	**0.439**	**0.645**	**0.841**	0.393	0.598	0.817
GLOMAP [186]	-	SuperPoint [63]	0.067	0.160	0.347	0.062	0.148	0.331
Ours	DUSt3R [274]	MASt3R [140]	**0.432**	**0.639**	**0.836**	**0.407**	**0.621**	**0.825**

Table 7.3 **Structure-from-motion** benchmark on the ScanNet dataset [53]. (dense-set SfM)

7.4.3 Structure-from-Motion Evaluations

We evaluate SfM performance on both sparse-set and dense-set datasets, following the MASt3R-SfM evaluation protocol and metrics [66].

Sparse-Set. As shown in Table 7.2, our method achieves state-of-the-art performance on ETH3D with a significant improvement over competing baselines. Notably, our method achieves 100% in RRA on 10/13 and 95% in RTA on 9/13 scenes, with expected improvement at @3 and @1 benchmarks.

Dense-Set. To highlight the plug-and-play modularity of our method, we employ a variety of depth and correspondence estimators, comparing against COLMAP [211] and GLOMAP [186]. We report results in Table 7.3, observing superior performance in all configurations. We generally obtain the best performance with UniDepth as depth estimator, and RoMa and MASt3R as correspondence estimators in the uncalibrated and calibrated regimes respectively. We observe that textureless ScanNet challenges GLOMAP's global registration strategy, creating a gap to COLMAP.

These results indicate that the rich information from monocular depth priors enable our proposed MfS approach to achieve precise pose estimation beyond the best classical approaches, even in large-scale scenarios.

7.4.4 Camera Re-Localization Evaluations

Recall that camera re-localization is the task of processing a collection of *mapping* images with known camera poses to enable accurate pose estimation of new *query* images. Several approaches have been proposed for this challenging task, starting from geometric methods based on indexing

Category	Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
FM	AS [207]	4/1.96	3/1.53	2/1.45	9/3.61	8/3.10	7/3.37	3/2.22	5.1/2.46
	HLoc [203]	2/0.79	2/0.87	2/0.92	3/0.91	5/1.12	4/1.25	6/1.62	**3.4**/**1.07**
E2E	SC-wLS [288]	3/0.76	5/1.09	3/1.92	6/0.86	8/1.27	9/1.43	12/2.80	6.6/1.45
	NeuMaps [238]	2/0.81	3/1.11	2/1.17	3/0.98	4/1.11	4/1.33	4/1.12	3.1/1.09
	PixLoc [206]	2/0.80	2/0.73	1/0.82	3/0.82	4/1.21	3/1.20	5/1.30	**2.9**/**0.98**
SCR	ACE [22]	1.9/0.7	1.9/0.9	0.9/0.6	2.7/0.8	4.2/1.1	4.2/1.3	3.9/1.1	2.8/0.93
	DSAC* [25]	1.9/1.11	1.9/1.24	1.1/1.82	2.6/1.18	4.2/1.41	3.0/1.70	4.2/1.42	2.7/1.41
	HSCNet [145]	2/0.7	2/0.9	1/0.9	3/0.8	4/1.0	4/1.2	3/0.8	2.7/0.90
	HSCNet++ [272]	2/0.63	2/0.79	1/0.8	2/0.65	3/0.85	3/1.09	3/0.83	**2.29**/**0.81**
APR	Direct-PN [43]	10/3.52	27/8.66	17/13.1	16/5.96	19/3.85	22/5.13	32/10.6	20/7.26
	DFNet [42]	3/1.15	9/3.71	8/6.08	7/2.14	10/2.76	9/2.87	11/5.58	8/3.47
	LENS [172]	3/1.3	10/3.7	7/5.8	7/1.9	8/2.2	9/2.2	14/3.6	8/3.00
	marepo [41]	2.1/1.24	2.3/1.39	1.8/2.03	2.8/1.26	3.5/1.48	4.2/1.71	5.6/1.67	**3.2**/**1.54**
FoundationMDE	MfS (Ours)	2.2/0.77	1.9/0.80	1.1/0.80	3.0/0.91	4.3/1.04	3.7/1.32	2.7/0.78	**2.7**/**0.92**

Table 7.4 **Camera relocalization** benchmark on the 7-Scenes dataset [220]. Note only centimeter precision was reported for most methods.

input images into an explicit map, e.g., as a 3D point cloud, to more recent learning methods that directly encode the scene into the weights of a neural network. State-of-the-art methods can be roughly categorized into: feature matching (FM), end-to-end (E2E), scene coordinate regression (SCR), and absolute pose regression (APR). To comprehensively compare against existing methods, we use the 7-scenes dataset [219] following the benchmarks reported in marepo [41] and HSCNet++ [272].

Implementation. Our method remains the same except for the initialization stage, where we adopt RoMa [70] along with DUSt3R's two-view estimation [274], using the image with the highest similarity score retrieved by DIR [89]. This simple adaptation testifies to the robustness of our optimization strategy, leveraging deep priors for monocular depth. We note that processing the 7-Scenes dataset [219] particularly benefits from the scalability of our approach and multi-core implementation, given the sheer size of the dataset.

Analysis. As summarized in Table 7.4, our approach is comparable to or surpasses state-of-the-art camera localization algorithms. Noticeably, our method exhibits superior robustness due to the adoption of the robust inlier-counting scoring function philosophy. In the challenging Stairs scene, characterized by extensive repetitive and textureless surfaces, our method successfully registers the cameras by maximizing the inlier count. This is achieved by leveraging the sub-graph scoring

Ablation	Acc@3°	Acc@5°	Acc@10°
Initialization	0.125	0.359	0.678
w.o. Coarse-Stage SfM	0.351	0.582	0.810
w.o. Fine-Stage SfM	0.396	0.607	0.821
Full Scheme	0.432	0.636	0.833
DUSt3R [274] Depthmap	0.426	0.631	0.830
DUSt3R [274] Pointcloud	0.296	0.497	0.726

Table 7.5 **Ablation** on calibrated ScanNet [210]: UniDepth [190] (top) and DUSt3R [274] (bottom) with RoMa [70].

function with a logarithmic loss. Notably, the state-of-the-art methods HSCNet++, ACE, and DSAC+ are all learning-based scene coordinate regression approaches. Given the poor ground truth quality of the 7-Scenes dataset [23], these learning-based methods may inadvertently learn dataset-specific biases, potentially skewing the comparison in their favor.

7.4.5 Ablation Study

We ablate the key design decisions below in Tab. 7.5.

Algorithm Stages. The coarse stage focuses on registering all frames to their correct locations even under poor initialization. The fine-stage SfM further refines pose accuracy by emphasizing a small subset of reliable depth and correspondence pixels. Both stages improve performance.

Depth Format. We compare depth maps to the point clouds recently popularized by DUSt3R [274]. We assume point cloud estimation inherently adopts an over-parameterized pixel-wise intrinsic model, significantly reducing overall SfM performance. Our results further underscore the benefits of dense depth maps from powerful MDE models [190].

7.5 Discussion

Large-scale dense-set SfM evaluations. Recent learning-based methods claim to surpass classical approaches, where such evaluations are typically focused on sparse-set SfM [227, 26, 66, 22]. Of the methods we compare to in our evaluations, FlowMap [227] and AceZero [26] evaluate COLMAP [211] by assessing image rendering quality after training a NeRF. However, the inherent randomness and complexity of NeRF training introduce additional factors and unknowns, making it harder to draw conclusions regarding relative performance on the fundamental SfM task. On

Scene	marepo [41]	DSAC* [25](Full)	DSAC* [25](Tiny)	ACE [22]	MfS
Cubes	71.8%	83.8%	68.7%	**97.0**%	75.1%
Bears	80.7%	82.6%	73.1%	80.7%	**100**%
Winter Sign	0.0%	0.2%	0.3%	1.0%	**9.3**%
Inscription	37.1%	54.1%	41.3%	**49.0**%	28.3%
The Rock	99.8%	**100**%	99.8%	**100**%	**100**%
Tendrils	29.3%	25.1%	19.6%	34.9%	**51.5**%
Map	55.1%	56.7%	53.3%	**56.5**%	45.1%
Square Bench	**70.7**%	69.5%	60.3%	66.7%	58.6%
Statue	0.0%	0.0%	0.0%	0.0%	0.0%
Lawn	34.2%	34.7%	20.0%	35.8%	**85.0**%
Average	47.9%	50.7%	43.6%	52.2%	**55.29**%

Table 7.6 **Camera relocalization** on Wayspots [22] dataset.

the other hand, MASt3R-SfM [66] evaluates SfM performance on the Tanks-and-Temples dataset (T&T) [131] but only on a sub-sampled version. Moreover, since T&T uses COLMAP-generated pseudo-ground truth, such evaluations are inherently biased, as has been highlighted in several studies [213, 23]. In summary, we promote direct comparisons of state-of-the-art feature-matching methods, such as COLMAP [211] in dense-set SfM, with hundreds to thousands of frames, as was recently reported in [186, 270].

Pushing the envelop on camera re-localization. To fully evaluate the efficacy of our MfS approach for camera re-localization, further evaluation on additional scenarios is needed. Note our strong results on ETH3D suggest the approach extends to outdoor settings. Evaluation on object-centric sequences, such as CO3Dv2 [198], would be valuable as learning-based methods typically perform well in these cases. It would be interesting to explore whether monocular depth priors alone can compensate for such specialized approaches, potentially reducing the need for per-scene adaptations as highlighted in recent studies [41].

Implications for 3D and Vision Foundation Models. The success of our optimization-based approach for multi-view tasks leveraging monocular depth priors, as recently demonstrated as well by [308], is similar in spirit to the success of detector-free SfM [105] leveraging dense feature matching to revise the traditional pipeline. Those results highlight the value of dense predictions, supplementing the recent trends utilizing point clouds following DUSt3R [274]. It

Type	Method	IMC Dataset		
		AUC@3°	AUC@5°	AUC@10°
Detector-Based	COLMAP (SIFT+NN)	24.87	34.47	45.94
	SIFT + NN + PixSfM [205]	26.45	35.73	47.24
	D2Net + NN + PixSfM [205]	10.27	13.12	17.25
	R2D2 + NN + PixSfM [205]	32.44	42.55	55.01
	SP + SG + PixSfM [205]	46.30	58.43	71.62
Detector-Free	LoFTR + PixSfM [205]	44.80	57.00	70.43
	DF-SfM [106] + LoFTR	46.9	59.14	72.44
	DF-SfM [106] + AspanTrans.	**47.58**	**59.88**	73.29
	DF-SfM [106] + MatchFormer	46.32	58.50	71.99
Deep-based	VGG-SfM [267]	45.23	58.89	**73.92**
FoundationMDE	MfS (Ours)	45.06	58.40	73.17

Table 7.7 **Structure-from-Motion** on IMC2021 [122] dataset.

would be interesting to further study this gap and explore effective trade-offs through novel network architectures.

7.6 Conclusion

We introduced a novel "Motion-from-Structure" approach that leverages monocular depth priors, offering notable benefits for various multi-view tasks. Our method achieves state-of-the-art results on challenging datasets like ETH3D [213, 212], while also showing competitive performance on ScanNet [53] and 7-Scenes [220]. We highlight the potential of fully capitalizing on monocular depth priors to advance 3D vision, enabling more efficient and scalable solutions for complex vision tasks. By eliminating the reliance on traditional SfM initialization and improving robustness, our approach paves the way for the future integration of monocular depth estimation in large-scale 3D vision applications.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

We present a robust system that integrates deep monocular and binocular models with optimization techniques to improve structure and motion estimation from images. Our approach enhances accuracy across both small-scale and large-scale image collections. By utilizing a depth prior that remains independent of camera motion, our system ensures reliable performance in challenging scenarios.

In Chapter 4, we introduce a novel monocular 3D prior, the incidence field, to calibrate monocular images. This incidence field provides a pixel-wise parameterization of intrinsic properties that remain invariant to image resizing and cropping. To recover camera intrinsics, we develop a RANSAC-based algorithm that ensures robust estimation. Extensive benchmarking demonstrates the effectiveness of our method in real-world, in-the-wild scenarios. Beyond calibration, we showcase multiple downstream applications that benefit from our approach, highlighting its broader impact on 3D vision tasks.

In Chapter 2 and Chapter 3, we present advancements in depth estimation and geometric matching by addressing key challenges in self-supervised learning and pretraining strategies. First, we introduce a depth estimation framework that explicitly leverages the mutual benefits between self-supervised depth estimation and semantic segmentation. Our approach advances the state-of-the-art, achieving performance comparable to supervised methods while significantly enhancing depth boundary accuracy. Additionally, we explore the benefits of pretraining both the encoder and decoder of a dense geometric matching network using the paired MIM task. By resolving the discrepancy between pretraining and fine-tuning, we improve geometric matching performance by reducing ambiguities in textureless regions and enhancing the representation of local planar surfaces.

In Chapter 5, we decompose two-view Structure-from-Motion (SfM) into three robust subtasks—normalized pose estimation, camera scale estimation, and residual depth estimation—ensuring

resilience to deficient views and improving both pose estimation and video-based depth reconstruction. Building on this, Chapter 6 leverages dense depthmaps and correspondence to achieve SoTA sparse-view pose accuracy, enabling diverse applications such as self-supervised correspondence learning, consistent depth estimation, and sparse-view neural rendering. Extending this further, Chapter 7 generalizes the principles from Chapter 6, demonstrating SoTA performance across various benchmarks in indoor and outdoor scenes, camera relocalization, and Structure-from-Motion tasks. Our approach proves effective for both small and large scale camera pose estimation, showcasing the significant potential of monocular depth estimation in advancing 3D vision.

8.2 Future Work Suggestions

Monocular Depth Estimation. Metric-space monocular depth estimation has become an increasingly important task. Recent studies suggest that camera intrinsics play a crucial role in accurate metric-space depth estimation. Therefore, depth estimation and camera calibration should be jointly conducted, *i.e.*, simultaneously estimating camera intrinsics and depth maps, effectively formulating a monocular SfM approach.

Correspondence Estimation. Geometric matching determines pixel-wise correspondences between two images. Recent studies have proposed various pixel-wise re-parameterizations of camera motion, highlighting the potential of geometric matching to simultaneously learn both image matching priors and camera motion priors.

Camera Calibration. Learning-based camera calibration methods still suffer from limitations due to insufficient camera models. Most datasets are collected using a single camera model, leading to a lack of diversity in available training data. One potential solution is to leverage large-scale EXIF image datasets, where focal length and camera model metadata from EXIF files provide a valuable supervision signal for fine-tuning camera models.

Structure-from-Motion. The SfM pipeline in our approach currently lacks a robust mechanism for enforcing multi-view consistency in depth triangulation. Integrating a learning-based triangulation pipeline could enable the system to benefit from both data-driven learning approaches and traditional optimization-based methods, improving overall reconstruction accuracy.

BIBLIOGRAPHY

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 2011.

[2] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021.

[3] Cuneyt Akinlar and Cihan Topal. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 2011.

[4] Paul Alvarez. Using extended file information (exif) file headers in digital evidence analysis. *IJDE*, 2004.

[5] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 2010.

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

[7] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017.

[8] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022.

[9] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: marginalizing sample consensus. In *CVPR*, 2019.

[10] Olga Barinova, Victor Lempitsky, Elena Tretiak, and Pushmeet Kohli. Geometric image parsing in man-made environments. In *ECCV*, 2010.

[11] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[12] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 100(3), 2005.

[13] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes–a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

[14] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 2008.

[15] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006.

[16] Sevinç Bayram, İsmail Avcıbaş, Bülent Sankur, and Nasir Memon. Image manipulation detection. *Journal of Electronic Imaging*, 2006.

[17] Christian Beder and Richard Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Joint Pattern Recognition Symposium*.

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

[19] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[20] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[21] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5044–5053, 2023.

[22] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated Coordinate Encoding: Learning to Relocalize in Minutes Using RGB and Poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[24] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017.

[25] Eric Brachmann and Carsten Rother. Visual Camera Re-Localization From RGB and RGB-D Images Using DSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), 2022.

[26] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene Coordinate Reconstruction: Posing of Image Collections via Incremental Learning of a Relocalizer. In *ECCV*, 2024.

[27] Gary Bradski and Adrian Kaehler. Opencv. *Dr. Dobb's journal of software tools*, 2000.

[28] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018.

[29] Garrick Brazil and Xiaoming Liu. M3D-RPN: Monocular 3D region proposal network for object detection. In *Proceeding of International Conference on Computer Vision*, 2019.

[30] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *PAMI*, 2010.

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

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

[33] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[34] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[35] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8001–8008, 2019.

[36] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[37] Bo Chen, Tat-Jun Chin, and Nan Li. Bpnp: Further empowering end-to-end learning with back-propagatable geometric optimization. *arXiv: 1909.06043*, 2019.

[38] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, 2022.

[39] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[40] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2624–2632, 2019.

[41] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-Relative Pose Regression for Visual Re-Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[42] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*. Springer, 2022.

[43] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute Pose Regression with Photometric Consistency. In *2021 International Conference on 3D Vision (3DV)*, 2021.

[44] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[45] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *ICCV*, 2019.

[46] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *arXiv preprint arXiv:1810.02695*, 2018.

[47] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. Ls-net: Learning to solve nonlinear least squares for monocular stereo. *ECCV*, 2018.

[48] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[49] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV*, 1999.

[50] David Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011.

[51] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2), 2020.

[52] 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.

[53] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[54] Luanyuan Dai, Xin Liu, Jingtao Wang, Changcai Yang, and Riqing Chen. Learning two-view correspondences and geometry via local neighborhood correlation. *Entropy*, 2021.

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

[56] Bibhash Pran Das, Mrutyunjay Biswal, Abhranta Panigrahi, and Manish Okade. Cnn based image resizing detection and resize factor classification for forensic applications. In *ICORT*, 2021.

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

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

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[60] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022.

[61] Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *ECCV*, 2008.

[62] Konstantinos G Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 2010.

[63] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.

[64] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[65] Elan Dubrofsky. Homography estimation. *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, 2009.

[66] Bardienus Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3R-SfM: a Fully-Integrated Solution for Unconstrained Structure-from-Motion, 2024.

[67] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *CVPR*, 2019.

[68] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *CVPR*, 2023.

[69] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Revisiting robust losses for dense feature matching. *arXiv preprint arXiv:2305.15404*, 2023.

[70] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *CVPR*, 2024.

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

[72] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *PAMI*, 2017.

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

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

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

[76] Torben Fetzer, Gerd Reis, and Didier Stricker. Stable intrinsic auto-calibration from fundamental matrices of devices with uncorrelated camera parameters. In *WACV*, 2020.

[77] 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.

[78] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

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

[80] Simon Fuhrmann, Fabian Langguth, and Michael Goesele. Mve-a multi-view reconstruction environment. In *GCH*, 2014.

[81] Yasutaka Furukawa and Carlos Hernández. Multi-View Stereo: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

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

[83] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

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

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

[86] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time RGB-D camera relocalization. In *ISMAR*, 2013.

[87] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

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

[89] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer, 2016.

[90] 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.

[91] Alexander Grabner, Peter M Roth, and Vincent Lepetit. Gp2c: Geometric projection parameter consensus for joint 3d pose and focal length estimation in the wild. In *ICCV*, 2019.

[92] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeuriPS*, 2020.

[93] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, Chengzhou Tang, Zilong Dong, and Ping Tan. Dro: Deep recurrent optimizer for video to depth. *IEEE Robotics and Automation Letters*, 2023.

[94] 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.

[95] Vitor Guizilini, Rareș Ambruș, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *CVPR*, 2022.

[96] Faiza Gul, Wan Rahiman, and Syed Sahal Nazli Alhady. A comprehensive study for robot navigation techniques. *Cogent Engineering*, 2019.

[97] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–500, 2018.

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

[99] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[100] Richard I Hartley. In defense of the eight-point algorithm. *PAMI*, 1997.

[101] Richard I. Hartley. Kruppa's equations derived from the fundamental matrix. *PAMI*, 1997.

[102] Richard I. Hartley and Peter Sturm. Triangulation. In Václav Hlaváč and Radim Šára, editors, *Computer Analysis of Images and Patterns*. Springer Berlin Heidelberg, 1995.

[103] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[104] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[105] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-Free Structure from Motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[106] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. *CVPR*, 2024.

[107] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, 2005.

[108] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007.

[109] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *CVPR*, 2018.

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

[111] Masa Hu, Garrick Brazil, Nanxiang Li, Liu Ren, and Xiaoming Liu. Camera self-calibration using human faces. In *FG*, 2023.

[112] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[113] Zhaoyang Huang, Xiaokun Pan, Runsen Xu, Yan Xu, Guofeng Zhang, Hongsheng Li, et al. Life: Lighting invariant flow estimation. *arXiv preprint arXiv:2104.03097*, 2021.

[114] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[115] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[116] Sergio Izquierdo and Javier Civera. Sfm-ttr: Using structure from motion for test-time refinement of single-view depth networks. In *CVPR*, 2023.

[117] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 690–706, 2018.

[118] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014.

[119] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *CVPR*, 2021.

[120] 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.

[121] Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Matzen, Matthew Sticha, and David Fouhey. Perspective fields for single image camera calibration. 2022.

[122] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

[123] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[124] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018.

[125] Fadi Khatib, Yuval Margalit, Meirav Galun, and Ronen Basri. Grelpose: Generalizable end-to-end relative camera pose regression. *arXiv preprint arXiv:2211.14950*, 2022.

[126] JooSeuk Kim and Clayton D. Scott. Robust kernel density estimation. *J. Mach. Learn. Res.*, 13(1):2529–2565, sep 2012.

[127] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

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

[129] Josef Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1983.

[130] 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.

[131] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4), 2017.

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

[133] Jana Košecká and Wei Zhang. Video compass. In *ECCV*, 2002.

[134] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. Comoda: Continuous monocular depth adaptation using past experiences. In *WACV*, 2021.

[135] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6647–6655, 2017.

[136] Hyunjoon Lee, Eli Shechtman, Jue Wang, and Seungyong Lee. Automatic upright adjustment of photographs with robust camera calibration. *PAMI*, 2013.

[137] 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.

[138] Jinwoo Lee, Hyunsung Go, Hyunjoon Lee, Sunghyun Cho, Minhyuk Sung, and Junho Kim. Ctrl-c: Camera calibration transformer with line-classification. In *ICCV*, 2021.

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

[140] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024.

[141] Haoang Li, Ji Zhao, Jean-Charles Bazin, Wen Chen, Zhe Liu, and Yun-Hui Liu. Quasi-globally optimal and efficient vanishing point estimation in manhattan world. In *ICCV*, 2019.

[142] Hongdong Li. A practical algorithm for l triangulation with outliers. In *CVPR*, 2007.

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

[144] Jianan Li, Xuemei Xie, Qingzhe Pan, Yuhan Cao, Zhifu Zhao, and Guangming Shi. Sgm-net: Skeleton-guided multimodal network for action recognition. *PR*, 2020.

[145] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[146] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.

[147] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[148] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.

[149] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[150] Tony Lindeberg. Scale invariant feature transform. 2012.

[151] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2010.

[152] Jinjiang Liu and Xueliang Zhang. Drc-net: Densely connected recurrent convolutional neural network for speech dereverberation. In *ICASSP*, 2022.

[153] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023.

[154] Yaojie Liu and Xiaoming Liu. Spoof trace disentanglement for generic face anti-spoofing. *PAMI*, 2022.

[155] Yaojie Liu, Joel Stehouwer, Amin Jourabloo, and Xiaoming Liu. Deep tree learning for zero-shot face anti-spoofing. In *CVPR*, 2019.

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

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

[158] Manolis IA Lourakis and Rachid Deriche. *Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondences to 3D measurements*. PhD thesis, INRIA, 1999.

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

[160] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[161] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3D holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018.

[162] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ToG*, 2020.

[163] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 155–163, 2018.

[164] Yi Ma, Stefano Soatto, Jana Košecká, and Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer, 2004.

[165] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *CVPR*, 2018.

[166] 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.

[167] Ishit Mehta, Parikshit Sakurikar, and PJ Narayanan. Structured adversarial training for unsupervised monocular depth estimation. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pages 314–323, 2018.

[168] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. In *WACV*, 2019.

[169] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *ACIVS*, 2017.

[170] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[171] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

[172] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*. PMLR, Nov 2022.

[173] Arsalan Mousavian, Hamed Pirsiavash, and Jana Košecká. Joint semantic segmentation and depth estimation with deep convolutional networks. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pages 611–619, 2016.

[174] 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.

[175] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 2017.

[176] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (SIGGRAPH)*, 41(4):1–15, 2022.

[177] Yosuke Nakagawa, Hideaki Uchiyama, Hajime Nagahara, and Rin-Ichiro Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In *3DV*, 2015.

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

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

[180] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022.

[181] David Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 2004.

[182] Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*, 2021.

[183] Carl Olsson, Anders Eriksson, and Richard Hartley. Outlier removal using duality. In *CVPR*, 2010.

[184] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. In *NeurIPS*, 2018.

[185] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, 2018.

[186] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024.

[187] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[188] Cosimo Patruno, Roberto Marani, Grazia Cicirelli, Ettore Stella, and Tiziana D'Orazio. People re-identification using skeleton standard posture and color descriptors from rgb-d data. *Pattern Recognition*, 2019.

[189] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. idisc: Internal discretization for monocular depth estimation. In *CVPR*, 2023.

[190] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[191] Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 9250–9256, 2019.

[192] Simon Placht, Peter Fürsattel, Etienne Assoumou Mengue, Hannes Hofmann, Christian Schaller, Michael Balda, and Elli Angelopoulou. Rochade: Robust checkerboard advanced detection for camera calibration. In *ECCV*.

[193] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pages 324–333, 2018.

[194] Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Geometry meets semantics for semi-supervised monocular depth estimation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 298–313, 2018.

[195] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *PAMI*.

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

[197] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, 2019.

[198] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[199] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeuriPS*, 2019.

[200] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

[201] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, 2020.

[202] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011.

[203] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[204] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[205] Paul-Edouard Sarlin, Philipp Lindenberger, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. *PAMI*, 2023.

[206] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning Robust Camera Localization From Pixels To Pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[207] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9), 2016.

[208] Grant Schindler and Frank Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *CVPR*, 2004.

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

[210] Johannes L. Schonberger and Jan-Michael Frahm. Structure-From-Motion Revisited. In *CVPR*, 2016.

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

[212] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle Adjusted Direct RGB-D SLAM. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[213] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[214] Mohammad Shafiei, Sai Bi, Zhengqin Li, Aidas Liaudanskas, Rodrigo Ortiz-Cayon, and Ravi Ramamoorthi. Learning neural transmittance for efficient rendering of reflectance fields. 2021.

[215] Mengyi Shan, Brian Curless, Ira Kemelmacher-Shlizerman, and Steve Seitz. Animating street view. In *SIGGRAPH Asia 2023 Conference Papers*, 2023.

[216] Yoli Shavit, Ron Ferens, and Yosi Keller. Coarse-to-fine multi-scene pose regression with transformers. *IEEE transactions on pattern analysis and machine intelligence*, 45(12):14222–14233, 2023.

[217] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. In *ECCV*, 2020.

[218] Xiaoming Liu Shengjie Zhu. Lighteddepth: ideo depth estimation in light of limited inference view angles. In *CVPR*, 2023.

[219] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[220] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013.

[221] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020.

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

[223] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CR, New York, 1986.

[224] Gilles Simon, Antoine Fond, and Marie-Odile Berger. A-contrario horizon-first vanishing point detection using second-order grouping laws. In *ECCV*, 2018.

[225] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[226] Saurabh Singh and Abhinav Shrivastava. EvalNorm: Estimating batch normalization statistics for evaluation. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3633–3641, 2019.

[227] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. *arXiv preprint arXiv:2404.15259*, 2024.

[228] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *Siggraph*. 2006.

[229] Jaime Spencer, Chris Russell, Simon Hadfield, and Richard Bowden. Kick back & relax: Learning to reconstruct the world by watching slowtv. In *ICCV*, 2023.

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

[231] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[232] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012.

[233] Peter Sturm. Multi-view geometry for general camera models. In *CVPR*, 2005.

[234] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021.

[235] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[236] J Sung, H Koppula, B Selman, and A Saxena. Cornell activity datasets: Cad-60 & cad-120, 2014.

[237] Chengzhou Tang and Ping Tan. BA-net: Dense bundle adjustment networks. In *ICLR*, 2019.

[238] Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. NeuMap: Neural Coordinate Mapping by Auto-Transdecoder for Camera Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[239] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *ICLR*, 2022.

[240] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *ICLR*, 2020.

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

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

[243] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 2016.

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

[245] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. Pseudo rgb-d for self-improving monocular slam and depth prediction. In *ECCV*, 2020.

[246] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *CVIU*, 2000.

[247] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9799–9809, 2019.

[248] Prune Truong, Martin Danelljan, Luc V Gool, and Radu Timofte. Gocor: Bringing globally optimized correspondence volumes into your neural network. In *NeuriPs*, 2020.

[249] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *CVPR*, 2020.

[250] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *arXiv preprint arXiv:2109.13912*, 2021.

[251] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *arXiv preprint arXiv:2109.13912*, 2023.

[252] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021.

[253] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, 2023.

[254] Tinne Tuytelaars, Krystian Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 2008.

[255] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In *NeuriPS*, 2020.

[256] Tluuldcus Ucicr. Feature-based image metamorphosis. *Computer graphics*, 26:2, 1992.

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

[258] 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.

[259] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[260] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[261] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010.

[262] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *PAMI*, 2008.

[263] Jayakorn Vongkulbhisal, Ricardo Cabral, Fernando De la Torre, and João P Costeira. Motion from structure (mfs): Searching for 3d objects in cluttered point trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[264] 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.

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

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

[267] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *CVPR*, 2024.

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

[269] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, 2015.

[270] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025.

[271] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022.

[272] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, Yi Zhao, Giorgos Tolias, and Juho Kannala. Hscnet++: Hierarchical scene coordinate classification and regression for visual localization with transformer. *International Journal of Computer Vision*, 132(7), Jul 2024.

[273] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. *arXiv preprint arXiv:2312.14132*, 2023.

[274] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[275] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *IROS*, 2020.

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

[277] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[278] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2162–2171, 2019.

[279] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *CVPR*, 2021.

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

[281] Horst Wildenauer and Allan Hanbury. Robust camera self-calibration from monocular images of manhattan worlds. In *CVPR*, 2012.

[282] Olivia Wiles, Sebastien Ehrhardt, and Andrew Zisserman. Co-attention for conditioned image matching. In *CVPR*, 2021.

[283] Scott Workman, Connor Greenwell, Menghua Zhai, Ryan Baltenberger, and Nathan Jacobs. Deepfocal: A method for direct focal length estimation. In *ICIP*, 2015.

[284] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wil. In *BMVC*, 2016.

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

[286] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013.

[287] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2011.

[288] Xin Wu, Hao Zhao, Shunkai Li, Yingdian Cao, and Hongbin Zha. Sc-wls: Towards interpretable feed-forward camera re-localization. In *European Conference on Computer Vision*. Springer, 2022.

[289] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013.

[290] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *NeuriPs*, 2021.

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

[292] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022.

[293] Yiliang Xu, Sangmin Oh, and Anthony Hoogs. A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. In *CVPR*, 2013.

[294] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *CVPR*, 2021.

[295] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024.

[296] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *ECCV*, 2018.

[297] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *arXiv preprint arXiv:1711.03665*, 2017.

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

[299] 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.

[300] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020.

[301] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.

[302] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016.

[303] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *ICCV*, 2023.

[304] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6044–6053, 2019.

[305] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1983–1992, 2018.

[306] Qichao Ying, Hang Zhou, Zhenxing Qian, Sheng Li, and Xinpeng Zhang. Robust image protection countering cropping manipulation. *arXiv preprint arXiv:2206.02405*, 2022.

[307] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *CVPR*, 2023.

[308] Yifan Yu, Shaohui Liu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. Relative Pose Estimation through Affine Corrections of Monocular Depth Priors. *arXiv preprint arXiv:2501.05446*, 2025.

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

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

[311] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *CVPR*, 2016.

[312] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*, 2018.

[313] Hui Zhang, K Wong Kwan-yee, and Guoqiang Zhang. Camera calibration from images of spheres. *PAMI*, 2007.

[314] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *ICCV*, 2019.

[315] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[316] Yueqiang Zhang, Langming Zhou, Haibo Liu, and Yang Shang. A flexible online camera calibration using line segments. *Journal of Sensors*.

[317] Zhengyou Zhang. A flexible new technique for camera calibration. *PAMI*, 2000.

[318] Zhengyou Zhang. Camera calibration with one-dimensional objects. *PAMI*, 2004.

[319] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[320] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4106–4115, 2019.

[321] Zhoutong Zhang, Forrester Cole, Richard Tucker, William T Freeman, and Tali Dekel. Consistent depth of moving objects in video. *TOG*, 2021.

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

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

[324] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022.

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

[326] Zhengming Zhou and Qiulei Dong. Two-in-one depth: Bridging the gap between monocular and binocular self-supervised depth estimation. In *ICCV*, 2023.

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

[328] Shengjie Zhu, Abhinav Kumar, Masa Hu, and Xiaoming Liu. Tame a wild camera: In-the-wild monocular camera calibration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[329] Shengjie Zhu, Abhinav Kumar, Masa Hu, and Xiaoming Liu. Tame a wild camera: in-the-wild monocular camera calibration. *Advances in Neural Information Processing Systems*, 36, 2024.

[330] Shengjie Zhu and Xiaoming Liu. Lighteddepth: Video depth estimation in light of limited inference view angles. In *CVPR*, 2023.

[331] Shengjie Zhu and Xiaoming Liu. Pmatch: Paired masked image modeling for dense geometric matching. In *CVPR*, 2023.

[332] Shengjie Zhu and Xiaoming Liu. Revisit self-supervised depth estimation with local structure-from-motion. In *ECCV*, 2024.

[333] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8856–8865, 2019.

[334] 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.

APPENDIX

A chronological list of all peer-reviewed and under-review publications completed during the Ph.D. program at MSU is provided below.

- *Shengjie Zhu*, Ahmed Abdelkader, Mark J. Matthews, Xiaoming Liu, and Wen-Sheng Chu. "Motion-from-Structure: Leveraging Monocular Depth Priors for Multi-View Tasks." Manuscript under review at International Conference on Computer Vision. 2025.
- *Shengjie Zhu*, and Xiaoming Liu. "Revisit Self-supervised Depth Estimation with Local Structure-from-Motion." Proceedings of the European Conference on Computer Vision. 2024.
- *Shengjie Zhu**, Girish Chandar Ganesan*, Abhinav Kumar, and Xiaoming Liu. "Remove Projective LiDAR Depthmap Artifacts via Exploiting Epipolar Geometry." Proceedings of the European Conference on Computer Vision. 2024.
- *Shengjie Zhu*, Abhinav Kumar, Masa Hu, and Xiaoming Liu. "Tame a wild camera: in-the-wild monocular camera calibration." Proceedings of the Advances in Neural Information Processing Systems. 2023.
- *Shengjie Zhu*, and Xiaoming Liu. "PMatch: Paired Masked Image Modeling for Dense Geometric Matching." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
- *Shengjie Zhu*, and Xiaoming Liu. "LightedDepth: Video Depth Estimation in light of Limited Inference View Angles." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
- *Shengjie Zhu*, Garrick Brazil, and Xiaoming Liu. "The Edge of Depth: Explicit Constraints between Segmentation and Depth." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.