Fusing the Old with the New: Learning Relative Camera Pose with Geometry-Guided Uncertainty

Bingbing Zhuang[1] Manmohan Chandraker[1,2]
[1]NEC Labs America [2]University of California, San Diego

Abstract

Learning methods for relative camera pose estimation have been developed largely in isolation from classical geometric approaches. The question of how to integrate predictions from deep neural networks (DNNs) and solutions from geometric solvers, such as the 5-point algorithm [37], has as yet remained under-explored. In this paper, we present a novel framework that involves probabilistic fusion between the two families of predictions during network training, with a view to leveraging their complementary benefits in a learnable way. The fusion is achieved by learning the DNN uncertainty under explicit guidance by the geometric uncertainty, thereby learning to take into account the geometric solution in relation to the DNN prediction. Our network features a self-attention graph neural network, which drives the learning by enforcing strong interactions between different correspondences and potentially modeling complex relationships between points. We propose motion parmeterizations suitable for learning and show that our method achieves state-of-the-art performance on the challenging DeMoN [61] and ScanNet [8] datasets. While we focus on relative pose, we envision that our pipeline is broadly applicable for fusing classical geometry and deep learning.

1. Introduction

Estimating the relative pose between two cameras is a fundamental problem in computer vision, which forms the backbone of structure from motion (SFM) methods. Geometric approaches based on the 5-point method [37] and bundle adjustment (BA) [59] are well-studied, while recent methods based on deep neural networks (DNNs) also achieve promising results [61, 7, 56, 65]. But the question of how the two families of methods may be combined to trade-off their relative benefits has as yet remained under-explored, which is the subject of our study in this paper.

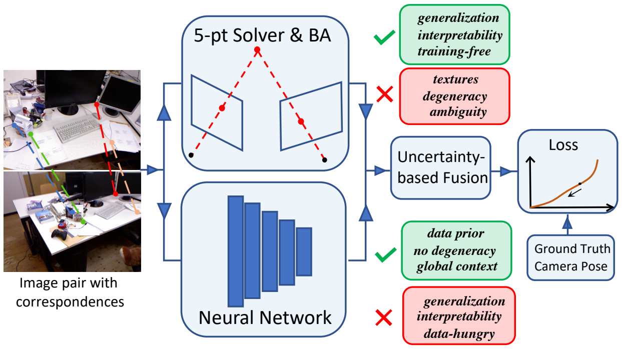


Figure 1. **Geometric-DNN relative pose fusion framework.** The DNN pose prediction is fused with the geometric prediction during training, based on their respective prediction uncertainty.

The behavior of geometric methods [18] is theoretically characterizable under a wide range of camera motions. But such understanding does not always guarantee good performance. While high accuracy is obtained in situations with strong perspective effects, performance may degrade due to lack of correspondences, planar degeneracy and bas-relief ambiguity [9], to name a few. On the other hand, learning-based methods may avoid the above issues by learning sophisticated priors that relate images to camera motion, but can suffer from poor generalization outside the training domain and not be amenable to interpretation.

This paper proposes an uncertainty based probabilistic framework to fuse geometric and DNN predictions, as illustrated in Fig. 1, with the aim of overcoming the limitations of either approach. The underlying intuition is that the geometric solution may be trusted more due to its well-understood rationale if it is highly confident, but the network should play a role in driving the solution closer to the true one in geometrically ill-conditioned scenarios. We obtain geometric uncertainty using the Jacobian of the error functions, serving as an indicator of the quality of the solution, while we design a network to additionally predict the uncertainty associated with camera pose estimation. The uncertainty so obtained may be interpreted as (co)variance of a Gaussian distribution, which allows us to fuse the two predictions using Bayes' rule. We highlight that the geometric solution and the fusion step are both tightly integrated into our end-to-end trainable pipeline, hence enforcing strong interaction between DNNs and the geometric method during training.

Our relative pose learning framework is the first of its kind in terms of forcing the network to give an account of the classical geometric solution along with its uncertainty in a principled way, during training. The network can also be thought of as a means to learn to improve the geometric solution such that the final fused one is closer to the ground truth. Further, the geometric guidance also distinguishes our uncertainty learning from previous works (e.g. [27]) that learn standard aleatoric uncertainty [26]. More importantly, our learned uncertainty may be considered as geometrically calibrated, in the sense that its numerical range can readily match to that of the geometric uncertainty and permits a direct fusion of the two during training.

In terms of network architecture, inspired by SuperGlue [46], we find a self-attention [62] graph neural network (GNN) to be effective at learning from keypoint correspondences. This is probably since self-attention permits strong interactions between correspondences, which is an essential procedure to determine the relative pose. We also illustrate that even both translation direction and rotation lie on a manifold, fusion is still feasible by careful choice of parameterization. We term our uncertainty-aware fusion framework as UA-Fusion. UA-Fusion is extensively validated by achieving state-of-the-art performance, especially in challenging indoor datasets with unconstrained motions.

In summary, our contributions include:

- A principled fusion framework to leverage the best of both classical geometric solvers and DNNs for relative pose estimation.
- A self-attention graph neural network whose attention mechanism drives the learning in our fusion pipeline.
- Superior results on benchmark DeMoN dataset [61] as well as in cross-dataset ScanNet experiments [8].

2. Related Works

Geometric methods Due to its essential role in SFM [48, 75, 77], a wide variety of algorithms have been proposed for the relative pose estimation problem. These could be classified into algebra-based minimal/nonminimal solvers [34, 37, 19, 32, 29] and optimization-based nonlinear methods [28, 67, 5, 12]. In addition, some methods are specially tailored to specific types of camera setup or motion prior [71, 21, 76, 16, 53, 15, 14]. Our framework does not require the geometric methods to be differentiable, and in principle could work with any approaches.

Learning methods Deep learning has recently been applied to different problems in SFM e.g. [45, 47, 58, 55, 78]. In particular, both supervised [61, 7, 56, 65] and unsupervised [73, 64, 35, 3, 72] methods for relative pose estimation are proposed. Despite the promising performance, such methods are largely developed in isolation of geometric methods. Although many works do borrow ideas from geometric methods to guide the network designing, (e.g. the spirit of bundle adjustment in BA-Net [56], LS-Net [7] and DeepSfM [65]), the geometric solution *itself* is not explicitly leveraged in network training as we do. Note that while we validate our concepts by learning the camera pose alone, our idea is amenable to learning with depth as well.

Geometric uncertainty In addition to the well-behaved camera pose estimation, geometric methods also offer a natural way to measure the uncertainty [11] of the predictions without requiring access to the ground truth. This is achieved by relating the Jacobian of the error landscape to the covariance of Gaussian distributions. The uncertainty so obtained has been extensively studied in photometric computer vision [11]. It has also found applicability in a wide variety of tasks, such as RGB-D SLAM [10], radial distortion model selection [42] in SFM, skeletal images selection for efficient SFM [51], height map fusion [79], 3D reconstruction [17], and camera calibration [39]. Polic et al. recently make efforts [41, 40] towards efficient uncertainty computation in large-scale 3D reconstruction. Despite the prevalence of the geometric uncertainty, it is however not yet fully exploited when it comes to the context of deep learning. Our work makes contributions towards bridging this gap.

Learning uncertainty Uncertainty learning, as a means towards more interpretable deep learning models, has emerged as an important topic recently [26]. Quantifying uncertainty of network predictions is highly desirable in many applications, such as camera relocalization [25], depth uncertainty [4] and photometric uncertainty [68] in SLAM, and optical flow estimation [22]. In the context of SFM, Klodt and Vedaldi [27] utilize uncertainty learning to handle the varying reliability of geometric SFM when applied as reference ground truth to supervise the SFM learning. Our paper is close to but distinct from this work since our goal lies in the fusion between the geometric and learned SFM with both the geometric and learned uncertainty. Laidlow et al. [30, 31] put forth to fuse the depth maps obtained from learning and geometric methods, sharing similar spirit to our pose fusion framework. Yet, their uncertainty is learned independent from the geometric solution, and requires a non-trivial post-processing fusion step. In contrast, our UA-Fusion learns uncertainty by tightly integrating geometric guidance and fusion into network training.

3. Method

Tradeoffs between geometric and learned pose We start by noting intuition from geometric pose estimation, which may guide regimes where interesting trade-offs may be observed for our uncertainty-based fusion.

- *Critical keypoint configurations:* Geometric solvers suffer when correspondences are scarce, for example, in textureless regions. Some common keypoint configurations may lead to ambiguous solutions or ill-posed objectives, for example, when keypoints lie on a plane [18]. We expect that our geometric uncertainty will lend greater importance to DNN priors in such situations.
- *Critical motions:* Bas-relief ambiguity [9, 54] may arise when the camera undergoes sideways motion due to the resemblance between translational and rotational flow under limited field of view, leading to a less accurate pose estimation. Forward motion also poses challenges to geometric SFM, partially due to small feature movement near the focus of expansion and partially due to severe local minima in the least squares error landscape [63, 38].
- *Rotation and translation:* Translation estimates are known to be more sensitive than rotation, leading to issues such as forward motion bias in linear methods if proper normalization is not carried out [37, 57]. Thus, one may expect rotation to be more reliable in the geometric solution, while the DNN may play a more significant role for translation. We will show that our uncertainty-based framework handles this in a principled way.

3.1. Background: Geometric Uncertainty

Geometric solution Formally, we are interested in solving the relative camera pose between two cameras C_1 and C_2 with known intrinsics. We assume C_1 as the reference with pose denoted as $\boldsymbol{P}_1 = [\boldsymbol{I} \;\; \mathbf{0}]$ and wish to estimate the relative camera pose of C_2, denoted as $\boldsymbol{P}_2 = [\boldsymbol{R} \;\; \boldsymbol{t}]$, where $\boldsymbol{R} \in SO(3)$ and $t \in \mathcal{S}^2$ denote the relative rotation and translation direction, respectively. Suppose both cameras are viewing a set of common 3D points $\boldsymbol{X}_i$, $i = 1, 2, ..., n$, each yielding a pair of 2D correspondences $\boldsymbol{x}_i^1$ and $\boldsymbol{x}_i^2$ in the image plane. It is well-known that a minimal set of 5-point correspondences suffices to determine the solution, with Nister's 5-point algorithm [37] being the standard minimal solver. A RANSAC procedure is usually applied to obtain an initial solution, followed by triangulation [18] to obtain 3D points $\boldsymbol{X}_i$. Finally, one could refine the solution by nonlinearly minimizing the re-projection error using bundle adjustment [59],

$$\min_{\boldsymbol{\theta}} \sum_i \|\boldsymbol{x}_i^1 - \pi(\boldsymbol{P}_1, \boldsymbol{X}_i))\|^2 + \|\boldsymbol{x}_i^2 - \pi(\boldsymbol{P}_2, \boldsymbol{X}_i))\|^2, \quad (1)$$

where $\pi()$ denotes the standard perspective projection and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_R, \; \boldsymbol{\theta}_t, \boldsymbol{X}_i, i = 1, 2, ...n\}$. $\boldsymbol{\theta}_R$ and $\boldsymbol{\theta}_t$ represent the parameterizations of rotation and translation; we will come back to their specific choice in the next section.

Geometric uncertainty In order to describe the uncertainty associated with the optimum $\hat{\boldsymbol{\theta}}$ in a probabilistic manner, the distribution of $\boldsymbol{\theta}$ could be approximated locally by a Gaussian distribution $\mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma})$. As a first-order approximation, the information matrix $\boldsymbol{\Lambda}$, i.e. $\boldsymbol{\Sigma}^{-1}$, is computed as:

$$\boldsymbol{\Lambda} = J^{\top}(\hat{\boldsymbol{\theta}}) J(\hat{\boldsymbol{\theta}}), \quad (2)$$

where $J(\hat{\boldsymbol{\theta}})$ denotes the Jacobian of the nonlinear least squares (Eq. 1) at $\hat{\boldsymbol{\theta}}$. We remark that $J(\hat{\boldsymbol{\theta}})$ is of full rank in this paper, implying the absence of gauge ambiguity [11, 24]. This is attributed to the fixed camera pose of C_1 as well as our minimal parameterizations of $(\boldsymbol{R}, \boldsymbol{t})$ to be discussed shortly. Also, we shall conduct fusion on each individual parameter in $\{\boldsymbol{\theta}_R, \; \boldsymbol{\theta}_t\}$ separately due to the discontinuity [74] in representation, and we will directly work on inverse variance for convenience. The inverse variance $1/\sigma_i^2$ of a parameter θ_i in $\{\boldsymbol{\theta}_R, \; \boldsymbol{\theta}_t\}$ may be obtained by Schur complement:

$$1/\sigma_i^2 = \boldsymbol{\Lambda} \setminus \boldsymbol{\Lambda}_{J,J} = \boldsymbol{\Lambda}_{i,i} - \boldsymbol{\Lambda}_{i,J} \boldsymbol{\Lambda}_{J,J}^{-1} \boldsymbol{\Lambda}_{J,i}, \quad (3)$$

where J includes the index to the remaining parameters in $\boldsymbol{\theta}$. This step is also called S-transformation [2] that specifies the gauge of covariance matrix [11, 42]. From the probabilistic point of view, it is in essence the inverse variance of a conditional Gaussian on θ_i given all the other parameters [23]. As the expert reader may have noticed, we omit the keypoint localization uncertainty [11, 39] of $\boldsymbol{x}_i^{1,2}$, for simplicity.

3.2. Geometric-DNN Relative Pose Fusion

Notations Whenever ambiguity arises, we use subscript g, d, and f to distinguish the prediction from the geometric prediction (g), the DNN prediction prior to fusion (d), and the fused solution (f).

Bayes fusion We conceptually treat the geometric and DNN predictions as measurements from two different sensors and fuse them using Bayes' rule, akin to a Kalman filter. Specifically, given $(\hat{\theta}_{i,g}, 1/\sigma_{i,g}^2)$ and $(\hat{\theta}_{i,d}, 1/\sigma_{i,d}^2)$ as the geometric and DNN prediction, respectively, the posterior distribution of the motion parameter θ_i is [36]

$$P(\theta_i|\hat{\theta}_{i,g}, \sigma_{i,g}^2, \hat{\theta}_{i,d}, \sigma_{i,d}^2) = \mathcal{N}(\theta_i|\hat{\theta}_{i,f}, \sigma_{i,f}^2), \quad (4)$$

$$\hat{\theta}_{i,f} = \frac{1/\sigma_{i,g}^2 \; \hat{\theta}_{i,g} + 1/\sigma_{i,d}^2 \; \hat{\theta}_{i,d}}{1/\sigma_{i,g}^2 + 1/\sigma_{i,d}^2}, \; 1/\sigma_{i,f}^2 = (1/\sigma_{i,g}^2 + 1/\sigma_{i,d}^2). \quad (5)$$

The maximum-a-posterior (MAP) estimation $\hat{\theta}_{i,f}$ is returned as the final fused solution, which receives supervision.

3.2.1 Neural Architecture for Probabilistic Fusion

Architecture Overview As illustrated in Fig. 2, our UA-Fusion framework takes as input an image pair along with the correspondences extracted by a feature matcher, for which we use SuperGlue [46] due to its excellent performance. The two images are stacked and passed to a ResNet [20] architecture to extract the appearance feature. The corresponding keypoint locations are first embeded into a higher-dimensional space by a Multilayer Perceptron (MLP) and

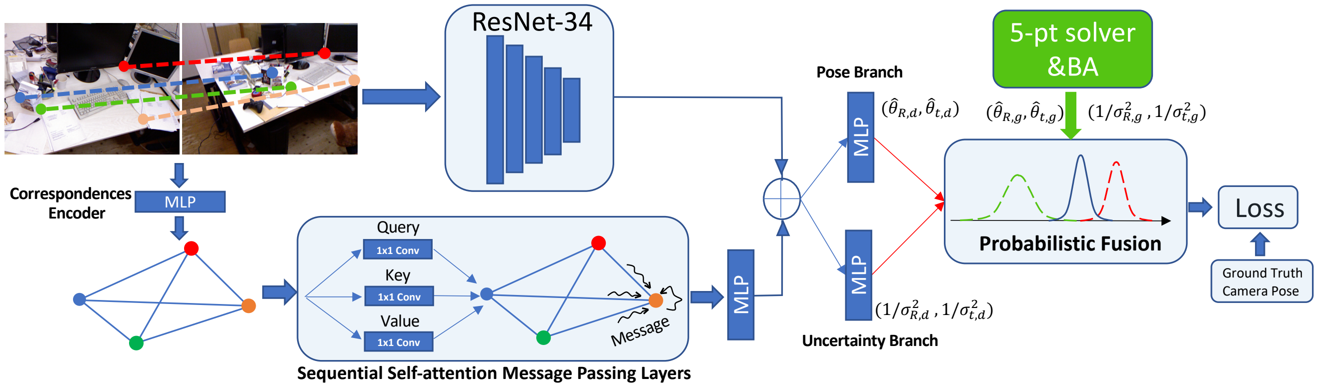

ResNet-34
5-pt solver &BA
Pose Branch
MLP
$(\hat{\theta}_{R,d}, \hat{\theta}_{t,d})$
$(\hat{\theta}_{R,g}, \hat{\theta}_{t,g})$
$(1/\sigma^2_{R,g}, 1/\sigma^2_{t,g})$
Probabilistic Fusion
Loss
Ground Truth Camera Pose
Correspondences Encoder
MLP
Query
1x1 Conv
Key
1x1 Conv
Value
1x1 Conv
Message
MLP
MLP
$(1/\sigma^2_{R,d}, 1/\sigma^2_{t,d})$
Uncertainty Branch
Sequential Self-attention Message Passing Layers

Figure 2. **Overview of our geometric-DNN fusion network**, taking two images with extracted keypoint correspondences as input. The images and correspondences are respectively passed to the ResNet-34 and the self-attention graph network for appearance and geometric feature extraction. The concatenated features are passed to the two-branch MLPs for estimating pose and uncertainty. This solution is then fused with the one from the 5pt&BA method with Jacobian-based uncertainty, yielding the final prediction which receives supervision.

then feed into an attentional graph neural network to extract the geometric feature. Afterwards, the appearance and geometric features are concatenated before being passed to the pose and the uncertainty branches, each using an MLP to predict respectively the mean and inverse variance of the underlying Gaussian distribution of the motion parameters. These are then fused with the geometric solution (5pt&BA) based on uncertainty. Note that the loss is imposed on the final fused output, which induces gradient flows back-propagated through the fusion, hence coupling the geometric and learning module in training.

Intuition for the Architecture While the ResNet offers global appearance context, the graph neural network and geometric feature encode strong geometric cues from any available correspondences to reason about camera motion. Further, correspondences as the sole input to the 5-point solver and bundle adjustment have a more explicit correlation with the uncertainty of geometric solution, allowing the network to decide the extent to which the geometric solution should be trusted in relation to the DNN prediction.

Self-Attention Graph Neural Network As the network input, we stack all the correspondences $(\boldsymbol{x}_i^1, \boldsymbol{x}_i^2)$ between the two views as $\boldsymbol{x}^{12} \in R^{n\times 4}$, which is subsequently passed to an MLP for embedding, yielding $\boldsymbol{f}^{(0)} \in R^{n\times d}$, with $d = 128$ being the feature dimension. Next, $\boldsymbol{f}^{(0)}$ is passed to four sequential message passing layers to propagate information between all pairs of correspondences, with structure similar to SuperGlue [46]. The message passing is best represented as a graph, with each node containing a pair of correspondence and edges connecting different pairs of correspondences. Specifically, in the l-th layer, the feature vector $\boldsymbol{f}_i^l$ associated with the correspondence pair i is updated as:

$$\boldsymbol{f}_i^{l+1} = \boldsymbol{f}_i^l + \text{MLP}([\boldsymbol{f}_i^l, \boldsymbol{m}_i^l]), \tag{6}$$

where [. , .] indicates concatenation and $\boldsymbol{m}_i^l$ denotes the message aggregated from all the correspondences based on the self-attention mechanism [62]. As per the standard procedure, we define the query ($\boldsymbol{Q}^l$), key ($\boldsymbol{K}^l$) and value ($\boldsymbol{V}^l$) as linear projections of $\boldsymbol{f}^l$, each with their own learnable parameters shared across all the correspondences. The message $\boldsymbol{m}^l$ is then computed as

$$\boldsymbol{m}^l = \text{softmax}(\frac{\boldsymbol{Q}^l {\boldsymbol{K}^l}^\top}{\sqrt{d}})\boldsymbol{V}^l, \tag{7}$$

The softmax is performed row-wise and $\boldsymbol{m}_i^l$ is row i of $\boldsymbol{m}^l$. The output of the last layer $\boldsymbol{f}_i^4$ is passed to an MLP and average pooling to be concatenated with the ResNet feature.

Why self-attention? First of all, the self-attention encourages interactions between all correspondences, which mimics the knowledge from classical geometry that all pairs of correspondences ($n >= 5$) together contribute to the relative pose determination, necessitating the interactions between points. More importantly, the strong representation capability of self-attention facilitates the learning of complex relationships among different pairs of correspondences. A straightforward example is the spatial relation. It is known in classical SFM [18] that two pairs of correspondences far from each other and widely spread in the image plane provide stronger signals for motion estimation; it effectively makes full use of the perspective effect in the field of view and prevents the degradation of perspective to affine camera model [18, 50]. Conversely, two pairs of correspondences near to each other typically contribute weaker extra cues compared to either pair alone. Hence, different pairs of correspondences are not on equal footing and should be treated differently. Indeed, we empirically observe a strong correlation between the attention and the spatial distance. In a nutshell, the self-attention enforces extensive interactions and permits learning more complex and abstract relationships among different pairs of correspondences, which we

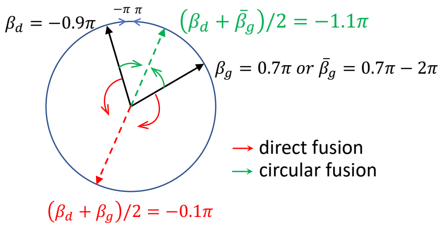

$\beta_d = -0.9\pi$
$-\pi$ π
$(\beta_d + \bar{\beta}_g)/2 = -1.1\pi$
$\beta_g = 0.7\pi$ or $\bar{\beta}_g = 0.7\pi - 2\pi$
direct fusion
circular fusion
$(\beta_d + \beta_g)/2 = -0.1\pi$

Figure 3. **Toy-example illustration** of the direct and circular fusion on β, wherein the fusion is simply defined as averaging.

empirically observe to play a significant role in terms of performance. More analyses will follow in the experiments.

3.2.2 Motion parameterization

It is crucial to choose the proper motion parameterizaton for the above network, which we discuss now.

Translation We consider the properties of two distinct parameterizations for training and fusion:

$$\boldsymbol{t}(t_x, t_y, t_z) = \frac{(t_x, t_y, t_z)^\top}{\|(t_x, t_y, t_z)^\top\|}, \tag{8}$$

$$\boldsymbol{t}(\alpha, \beta) = (\cos\alpha, \sin\alpha\cos\beta, \sin\alpha\sin\beta), \tag{9}$$

where $\alpha \in [0, \pi]$ and $\beta \in [-\pi, \pi]$ are constraints for uniqueness. Since the unit-norm vector $\boldsymbol{t}$ itself is not a convenient quantity for fusion as it lies on the $\mathcal{S}^2$ manifold, we seek to fuse the parameters (t_x, t_y, t_z) or (α, β). As the scale of (t_x, t_y, t_z) is indeterminate, causing gauge ambiguity and a rank-deficient Jacobian, we opt for (α, β) as the fusion entity, that is, $\boldsymbol{\theta}_t = \{\alpha, \beta\}$. However, due to its circular nature, the wrap-around of β at $\pm\pi$ leads to discontinuity in the representation, which leads to training difficulties if the network predicts β directly. To address this issue, we design the network to output (t_x, t_y, t_z) followed by unit-normalization, then we extract (α, β) and proceed to fusion.

Circular Fusion While the fusion of α remains straightforward, the circular nature slightly complicates the fusion of β. Ideally, a meaningful fusion is obtained only when $|\beta_d - \beta_g| < \pi$, which could be achieved by letting $\bar{\beta}_g = \beta_g + 2k\pi$ with $k \in \{-1, 0, 1\}$. This is illustrated by the toy example in Fig. 3, where depending upon the specific values of β_g and β_d, a direct fusion of the two might yield a solution far from both when $|\beta_d - \beta_g| > \pi$. This is, however, addressed by fusing β_d and $\bar{\beta}_g$ instead. We term this procedure as circular fusion and $\bar{\beta}_g$ as β_g's circular nearest neighbor to β_d.

Rotation We consider two minimal 3-parameter representation of rotation—angle-axis representation and Euler angles. The network is designed to regress the angle directly. Although this also faces discontinuity at $\pm\pi$ [74], it barely poses a problem since rotations between two views are often far from $\pm\pi$ (such strong rotation will quickly diminish the overlapping field of view). We observe similar performance from the two representations, but opt for Euler angels since its fusion of yaw-pitch-roll angles, denoted as $\boldsymbol{\theta}_R = \{\phi_y, \phi_p, \phi_r\}$, has a clearer geometric meaning.

Loss One could impose the loss directly on the fused angles $\{\boldsymbol{\theta}_{R,f}, \boldsymbol{\theta}_{t,f}\}$ with a circular loss [33], or convert them back to $(\boldsymbol{R}(\boldsymbol{\theta}_{R,f}), \boldsymbol{t}(\boldsymbol{\theta}_{t,f}))$ and impose loss on it. All the combinations perform similarly in our experiments; we choose the following loss for marginally better performance,

$$L(\boldsymbol{\theta}_{R,f}, \boldsymbol{\theta}_{t,f}) = |\boldsymbol{t}(\boldsymbol{\theta}_{t,f}) - \boldsymbol{t}^*|_1 + w|\boldsymbol{\theta}_{R,f} - \bar{\boldsymbol{\theta}}^*_R|_1, \tag{10}$$

where $*$ indicates the ground truth and $\bar{\boldsymbol{\theta}}^*_R$ is $\boldsymbol{\theta}^*_R$'s circular nearest neighbour to $\boldsymbol{\theta}_{R,f}$. $w = 1.0$ is a weight.

Implementation details We make use of OpenCV for the 5-point solver with RANSAC, and Ceres [1] for BA and Jacobian computation. More discussions on the framework and network training are in the supplementary.

4. Experiments

4.1. Dataset

DeMoN Dataset [61]. This dataset contains a large amount of data including indoor, outdoor, and synthetic data, extracted from SUN3D [66], RGB-D [52], MVS [60, 13, 49, 48], and ShapeNet [6]. The diversity of both scenes and motions makes the dataset challenging for SFM.

ScanNet [8]. To evaluate the generalization ability, we also test our network on the ScanNet data. We use the testing data extracted by [56], that consists of 2000 pairs of images with accurate ground truth pose. The sequences are captured indoors by hand-held cameras, making it particularly hard.

4.2. Results on DeMoN

We first compare UA-Fusion with the state-of-the-art relative pose learning methods including DeMoN [61], LS-Net [7], BA-Net [56] and DeepSfM [65], in Tab. 1. The error of translation (resp. rotation) is measured as the angle between the prediction and the ground truth. First, although SuperGlue followed by 5pt&BA produces competitive results compared to methods that learn pose directly, it can be seen that our fusion of the geometric and DNN prediction further boosts the performance, achieving the overall best results. We also test UA-Fusion with SIFT as the feature matcher, under which case ours again improves over the pure geometric method, further validating its merits. In addition, we test two more methods that focus on learning correspondences, LGC-Net [69] and NM-Net [70], by passing their output to 5pt&BA. We observe that the obtained accuracies lag behind our fusion method.

	MVS		Scenes11		RGB-D		Sun3D	
	Rot.	Tran.	Rot.	Tran.	Rot.	Tran.	Rot.	Tran.
DeMoN [61]	5.156	14.447	0.809	8.918	2.641	20.585	1.801	18.811
LS-Net [7]	4.653	11.221	4.653	8.210	**1.010**	22.110	1.521	14.347
BA-Net [56]	3.499	11.238	3.499	10.370	2.459	14.900	1.729	13.260
DeepSfM [65]	2.824	9.881	0.403	5.828	1.862	14.570	1.704	13.107
LGC-Net [69]	2.753	3.548	0.977	4.861	2.014	16.426	1.386	14.118
NM-Net [70]	6.628	12.595	15.717	31.477	13.444	34.212	4.393	21.091
SIFT+5pt&BA	1.313	2.555	2.062	8.125	6.895	29.457	2.516	21.925
UA-Fusion-SIFT	**1.203**	**2.403**	0.525	4.322	2.274	15.570	1.960	17.340
SuperGlue+5pt&BA	2.884	5.024	0.390	3.872	1.829	16.330	**1.255**	12.200
UA-Fusion	2.502	4.506	**0.388**	**3.001**	1.480	**10.520**	1.340	**11.830**

Table 1. **Quantitative comparison with state-of-the-art methods on DeMoN datasets.** Both translation (Tran.) and Rotation (Rot) errors are measured in degree ($\circ$). The lowest error in each column is bolded.

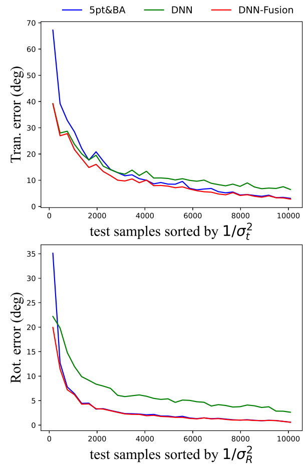


Figure 4. **Error comparison** between geometric and DNN predictions.

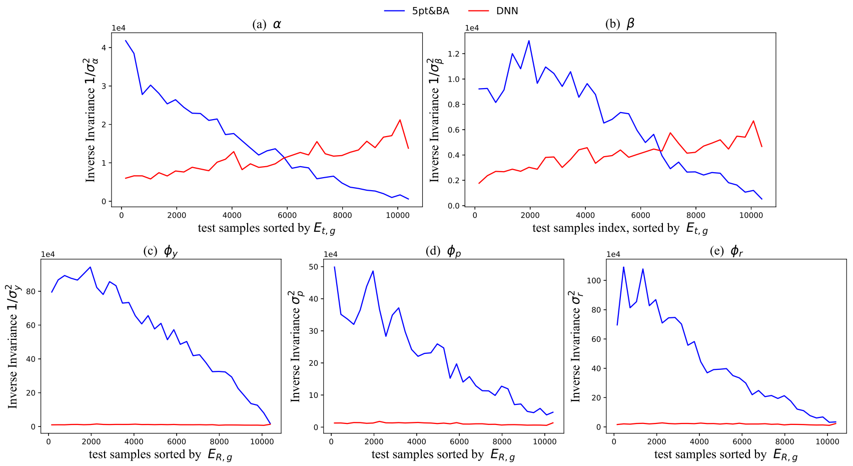


Figure 5. Comparison between the **geometric and DNN uncertainty** for each parameter in $\{\boldsymbol{\theta_t}, \boldsymbol{\theta_R}\}$.

4.3. Analysis: Geometric vs. DNN

Since the DeMoN test set contains only 354 pairs of images, we train a model with a few training sequences ($\sim$10k pairs) left for testing, in order for statistically meaningful analysis. We denote the translation and rotation error of the geometric solution ("5pt&BA"), the plain DNN predition prior to fusion ("DNN"), and the final fused solution ("DNN-Fusion") as $(E_{t,g}, E_{R,g})$, $(E_{t,d}, E_{R,d})$ and $(E_{t,f}, E_{R,f})$.

Geometric error vs. DNN error We first study the accuracy against the uncertainty of the geometrical solution. We sort the test data by the total inverse variance in translation $1/\sigma_t^2 = 1/\sigma_\alpha^2 + 1/\sigma_\beta^2$, and plot the sorted $E_{t,g}$, $E_{t,d}$ and $E_{t,f}$ in the top row of Fig. 4. The curves are smoothed by averaging over every 300 consecutive image pairs. First observe the overall trend that $E_{t,g}$ decreases while $1/\sigma_t^2$ increases, revealing the correlation between geoemtrical error and uncertainty. In addition, the accuracy gap between $E_{t,g}$ and $E_{t,f}$ is increasingly large with decreasing $1/\sigma_t^2$. This indicates a more significant role of DNNs in geometrically uncertain scenarios; conversely, the geometrical solution is mostly retained if it is highly confident. Further observe the superiority of "DNN-Fusion" over "DNN", which is expected as only "DNN-Fusion" receives direct supervision. Another potential reason is that "DNN" likely has to contain bias to compensate the error/bias in "5pt&BA" during fusion. The same curve is plotted for the rotation error in the bottom row of Fig. 4. We observe that the geometrical solution transfers to the final output in most cases, probably due to the higher stability of the rotation estimation compared to the translation; this viewpoint has been conveyed by Nistér [37].

Geometric uncertainty vs. DNN uncertainty Next, we visualize the uncertainty by plotting $(1/\sigma_\alpha^2, 1/\sigma_\beta^2)$ sorted by $E_{t,g}$. As shown in Fig. 5(a)(b), $(1/\sigma_{\alpha,d}^2, 1/\sigma_{\beta,d}^2)$ are higher than $(1/\sigma_{\alpha,g}^2, 1/\sigma_{\beta,g}^2)$ for cases with larger $E_{t,g}$, and vice verse. This implies the desired capability of "DNN" to dominate the final solution when "5pt&BA" tends to fail. Similarly, we present the same plot for rotation in Fig. 5(c)-(e). Comparing the range of $1/\sigma_{R,g}^2$ and $1/\sigma_{t,g}^2$, one observes the higher confidence/stabability in the geometric rotation

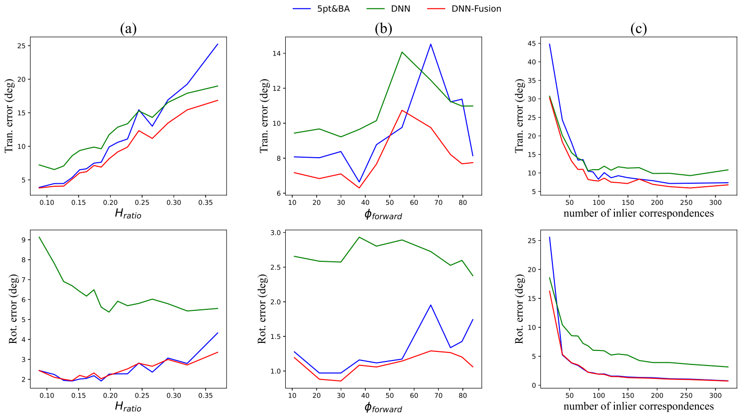


Figure 6. Comparisons of errors against (a) **homography degeneracy**, (b) **forward and sideway motion**, and (c) **correspondences**.

ScanNet	Rot.	Tran.
DeMoN [61]	3.791	31.626
LSD-SLAM [44]	4.409	34.360
BA-Net [56]	1.587	31.005
DeepSfM [65]	1.588	30.613
SuperGlue+5pt&BA (indoor)	1.776	20.936
UA-Fusion (indoor)	**0.814**	**16.517**
SuperGlue+5pt&BA (outdoor)	2.519	20.470
UA-Fusion (outdoor)	0.824	17.495

Table 2. **Quantitative comparisons on ScanNet**. Both indoor and outdoor SuperGlue model are tested.

estimate than the translation. This also makes it dominate in most cases when fused with the DNN prediction, as evidenced by the curves. We also note that the smoothed curves reveal the overall trend but may overly obscure the DNN's impact on the rotation, thus we provide in the supplementary more analyses in this regard.

Homography degeneracy Motion estimation may be unstable with correspondences that could be related by a homography. Characterizations of the algorithm's stability in this aspect is essential. To this end, we fit a homography to correspondences in each image pair; the ratio of inliers, H_{ratio}, is used to indicate closeness to degeneracy. We then sort image pairs by H_{ratio} and plot the error curves in Fig. 6(a). As can be seen, "DNN-Fusion" mostly keeps the geometric solution in well-conditioned cases with lower H_{ratio}, while alleviates the degradation under higher H_{ratio}.

Forward vs. Sideway motion We first select those image pairs with nearly pure translational motion, and compute the angle between translation direction and z-axis, denoted as $\phi_{forward}$. $\phi_{forward} = 0°$ and $90°$ indicate pure forward and sideway motion, respectively. We plot the errors sorted by $\phi_{forward}$ in Fig. 6(b). While forward motion does not cause serious issues in our tested data, one observes increasing errors while approaching pure sideway motion, which is liable to the bas-relif ambiguity. Yet, the performance degradation is alleviated in our fused solution.

Correspondence Finally, we plot in Fig. 6(c) the errors against the number of inlier correspondences found by "5pt&BA". Clearly, DNNs contribute more when the geometric accuracy drops due to lack of correspondences.

Qualitative Results We present here three qualitative examples. Fig. 8(a) represents a geometrically challenging case since all the correspondences nearly lie in a frontal plane with small depth variations, leading to weak perspective effect and potential degeneracy. Similarly, Fig. 8(b) shows an ill-conditioned instance with sparse correspondences concentrated on a small region. Fig. 8(c) demonstrates a case with nearly lateral motion along the vertical direction, which is liable to bas-relief ambiguity. As can be seen, the geometric method may deteriorate significantly when confronted with such challenges, especially the translation, whereas our fusion network returns more reasonable solutions.

4.4. Self-attention

A comprehensive understanding on what is learned by the attention module is challenging. However, as discussed in Sec. 3.2.1, we could study its potential relation to the spatial distance between any two pairs of correspondences. To this end, for a pair of correspondences, we collect its attentions to all the other pairs of correspondences in the last self-attention layer, and then compute its spatial distance[1] to other correspondences at different percentiles of the attention set. This way we can compare spatial distance against varying attention. Averaging over all the points in the entire testing set, we obtained the curve shown in Fig. 7. First, in line with the expectation, the overall trend indicates strong correlation between attention and spatial distance. In addition, it also reveals the overall smaller spatial distance with increasingly higher attentions. Referring to Sec. 3.2.1, we reckon that this is due to the increasing difficulty of extracting additional pose-related information from two points closer to each other; such difficulty enforces stronger interactions between those points in order to make contributions to the final camera pose estimation. More discussions are in the supplementary due to lack of space.

4.5. Baselines & Ablation Study

We provide ablation studies to elucidate the impact of important factors by comparison with several baselines.

Aleatoric Uncertainty: We replace our geometry-guided uncertainty with the plain aleatoric uncertainty [26, 27, 30],

[1] Denoting a pair of correspondences as $(\boldsymbol{x}_i^1, \boldsymbol{x}_i^2)$ and $(\boldsymbol{x}_j^1, \boldsymbol{x}_j^2)$, the spatial distance is defined as$(\|(\boldsymbol{x}_i^1 - \boldsymbol{x}_j^1)\| + \|(\boldsymbol{x}_i^2 - \boldsymbol{x}_j^2)\|)/2$.

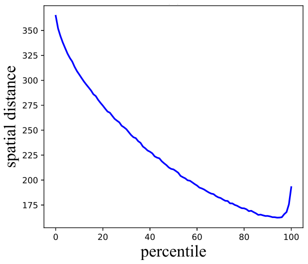
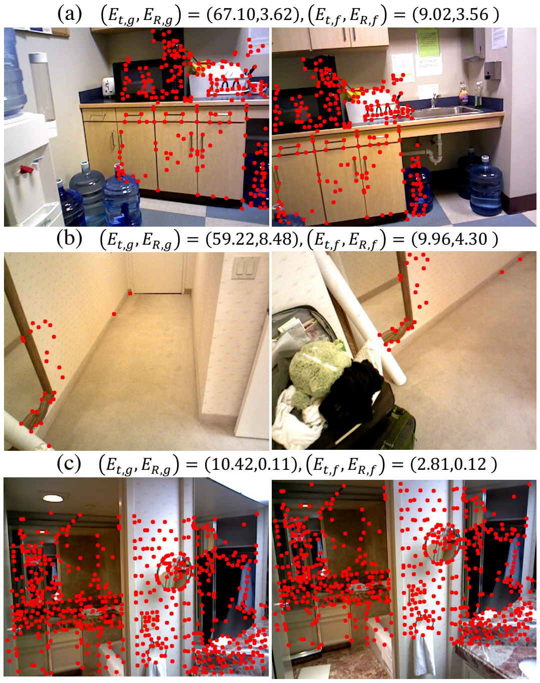


Figure 7. Study of spatial distances (pixels) against **attentions**.

	MVS		Scenes11		RGB-D		Sun3D		All	
	Rot.	Tran.	Rot.	Tran.	Rot.	Tran.	Rot.	Tran.	Rot.	Tran.
Aleatoric Uncertainty	2.890	4.954	**0.375**	2.956	1.813	13.650	1.273	11.750	1.371	7.694
w/o Uncertainty	4.608	7.233	1.034	5.924	4.121	12.400	2.479	13.870	2.712	9.200
Median Uncertainty	3.638	4.841	0.764	4.262	2.080	10.960	1.669	13.200	1.790	7.838
w/o Attention	2.956	6.051	0.410	3.237	1.678	12.65	1.321	12.14	1.380	7.870
Discontinuity	2.910	**3.984**	0.384	3.268	1.667	12.160	2.298	12.310	1.573	7.452
w/o ResNet	2.519	4.539	0.409	**2.805**	1.742	13.550	1.261	**11.630**	1.306	7.506
w/o GNN	3.035	6.080	0.399	3.266	1.826	14.220	**1.255**	13.550	1.406	8.589
PointNet	2.833	5.665	0.409	2.850	1.652	13.220	1.270	13.490	1.333	8.080
UA-Fusion	**2.502**	4.506	0.388	3.001	**1.480**	**10.520**	1.340	11.830	**1.246**	**6.888**

Table 3. **Ablation study** of our methods under different configurations. We also compute the averaged error over all the four scenes, as shown in the "All" column. The lowest error in each column is bolded.



(a) $(E_{t,g}, E_{R,g})$ = (67.10,3.62), $(E_{t,f}, E_{R,f})$ = (9.02,3.56)

(b) $(E_{t,g}, E_{R,g})$ = (59.22,8.48), $(E_{t,f}, E_{R,f})$ = (9.96,4.30)

(c) $(E_{t,g}, E_{R,g})$ = (10.42,0.11), $(E_{t,f}, E_{R,f})$ = (2.81,0.12)

Figure 8. **Qualitative results** with the geometric and DNN errors on top of each image pair. Red dots mark the correspondences.

and leverage fusion as a post-processing step.

w/o Uncertainty: We remove the uncertainty head and the fusion step, only regressing the camera pose.

Median Uncertainty: Instead of predicting uncertainty by the network, we simply assign each σ_d^{-1} a constant value—the median of σ_g^{-1}'s in the entire training set. This way, the DNN prediction would dominate the final solution on half of the data with highest geometric uncertainty.

w/o Attention: We remove the self-attention mechanism and use plain MLPs to compute the messages.

Discontinuity: In this case, the network directly predicts (α, β) instead of (t_x, t_y, t_z), as discussed in Sec. 3.2.2.

w/o ResNet and **w/o GNN**: Here, we remove either ResNet or self-attention GNN from the network, i.e. relying on the geometric feature or appearance feature alone.

PointNet: We replace the self-attention GNN with PointNet [43], which has a similar number of parameters.

As shown in Tab. 3, all the baseline methods lead to a drop in the accuracy. This substantiates the effectiveness of our geometry-guided uncertainty, self-attention mechanism, motion prameterization, and network design. In particular, removing uncertainty and fusion causes the most significant degradation. We present more analysis on the aleatoric uncertainty in the supplementary due to lack of space.

4.6. Results on ScanNet

To demonstrate the generalization capability of our network trained on the DeMoN datasets, we also conduct cross-validation experiments on the ScanNet dataset. Since the indoor model of SuperGlue is trained on ScanNet as well and its training set may have intersection with the testing set here, we instead apply its outdoor model to obtain correspondences. As a reference, we also report the results with the indoor model. As can been in Tab. 2, UA-Fusion achieves the best accuracy compared to prior arts. More importantly, we observe similar behavior in accuracy and uncertainty as analyzed in Sec. 4.3; this is detailed in the supplementary.

Computational performance: Our network inference takes about 0.01s per image pair, on an RTX2080 GPU. Computing geometric uncertainty takes around 0.025s, where the block sparsity of the Jacobian is leveraged for efficiency.

5. Conclusion

In this paper, we propose a new relative pose estimation framework capable of reaping benefits from both the classical geometric methods and deep learning approaches. At the crux of our method is learning the uncertainty explicitly governed by the geometric solution, which permits fusion in a probabilistic manner during training. Despite being specific to the relative pose problem, we envision that our idea of learning with geometric uncertainty guidance followed by fusion in a deep network is broadly applicable.

References

[1] Sameer Agarwal, Keir Mierle, et al. Ceres solver. 2012. 5

[2] Willem Baarda. S-transformations and criterion matrices. *Netherlands geodetic commission*, 1973. 3

[3] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, 2019. 2

[4] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *CVPR*, 2018. 2

[5] Jesus Briales, Laurent Kneip, and Javier Gonzalez-Jimenez. A certifiably globally optimal solution to the non-minimal relative pose problem. In *CVPR*, 2018. 2

[6] 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. 5

[7] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. Learning to solve nonlinear least squares for monocular stereo. In *ECCV*, 2018. 1, 2, 5, 6

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1, 2, 5

[9] Kostas Daniilidis, Minas E Spetsakis, and Y Aloimonos. Understanding noise sensitivity in structure from motion. *Visual navigation: from biological systems to unmanned ground vehicles*, 1997. 1, 3

[10] Alejandro Fontan, Javier Civera, and Rudolph Triebel. Information-driven direct rgb-d odometry. In *CVPR*, 2020. 2

[11] Wolfgang Förstner and Bernhard P Wrobel. *Photogrammetric computer vision*. Springer, 2016. 2, 3

[12] Johan Fredriksson, Viktor Larsson, Carl Olsson, and Fredrik Kahl. Optimal relative pose with unknown correspondences. In *CVPR*, 2016. 2

[13] Simon Fuhrmann, Fabian Langguth, and Michael Goesele. Mve-a multi-view reconstruction environment. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage (GCH)*, 2014. 5

[14] Banglei Guan, Ji Zhao, Daniel Barath, and Friedrich Fraundorfer. Relative pose estimation for multi-camera systems from affine correspondences. *arXiv preprint arXiv:2007.10700*, 2020. 2

[15] Banglei Guan, Ji Zhao, Zhang Li, Fang Sun, and Friedrich Fraundorfer. Minimal solutions for relative pose with a single affine correspondence. In *CVPR*, 2020. 2

[16] Levente Hajder and Daniel Barath. Relative planar motion for vehicle-mounted cameras from a single affine correspondence. In *ICRA*, 2020. 2

[17] Sebastian Haner and Anders Heyden. Covariance propagation and next best view planning for 3d reconstruction. In *ECCV*, 2012. 2

[18] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 3, 4

[19] Richard I Hartley. In defense of the eight-point algorithm. *PAMI*, 1997. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[21] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *CVPR*, 2013. 2

[22] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *ECCV*, 2018. 2

[23] Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*. Prentice hall Upper Saddle River, NJ, 2002. 3

[24] Ken-ichi Kanatani and Daniel D Morris. Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 2001. 3

[25] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 2

[26] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017. 2, 7

[27] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *ECCV*, 2018. 2, 7

[28] Laurent Kneip and Simon Lynen. Direct optimization of frame-to-frame rotation. In *ICCV*, 2013. 2

[29] Zuzana Kukelova. Algebraic methods in computer vision. 2013. 2

[30] Tristan Laidlow, Jan Czarnowski, and Stefan Leutenegger. Deepfusion: real-time dense 3d reconstruction for monocular slam using single-view depth and gradient predictions. In *ICRA*, 2019. 2, 7

[31] Tristan Laidlow, Jan Czarnowski, Andrea Nicastro, Ronald Clark, and Stefan Leutenegger. Towards the probabilistic fusion of learned priors into standard pipelines for 3d reconstruction. In *ICRA*, 2020. 2

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

[33] Quewei Li, Jie Guo, Yang Fei, Qinyu Tang, Wenxiu Sun, Jin Zeng, and Yanwen Guo. Deep surface normal estimation on the 2-sphere with confidence guided semantic attention. In *ECCV*, 2020. 5

[34] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981. 2

[35] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *CVPR*, 2018. 2

[36] Kevin P Murphy. *Machine learning: a probabilistic perspective*. 2012. 3

[37] David Nistér. An efficient solution to the five-point relative pose problem. *PAMI*, 2004. 1, 2, 3, 6

[38] John Oliensis. The least-squares error for structure from infinitesimal motion. *IJCV*, 2005. 3

[39] Songyou Peng and Peter Sturm. Calibration wizard: A guidance system for camera calibration based on modelling geometric and corner uncertainty. In *ICCV*, 2019. 2, 3

[40] Michal Polic, Wolfgang Forstner, and Tomas Pajdla. Fast and accurate camera covariance computation for large 3d reconstruction. In *ECCV*, 2018. 2

[41] Michal Polic and Tomas Pajdla. Camera uncertainty computation in large 3d reconstruction. In *3DV*, 2017. 2

[42] Michal Polic, Stanislav Steidl, Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. Uncertainty based camera model selection. In *CVPR*, 2020. 2, 3

[43] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 8

[44] Jakob Quewei, Li, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014. 7

[45] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 2

[46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 3, 4

[47] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3302–3312, 2019. 2

[48] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2, 5

[49] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 5

[50] Larry S Shapiro, Andrew Zisserman, and Michael Brady. 3d motion recovery via affine epipolar geometry. *IJCV*, 1995. 4

[51] Noah Snavely, Steven M Seitz, and Richard Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*, 2008. 2

[52] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 5

[53] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *3DV*, 2014. 2

[54] Richard Szeliski and Sing Bing Kang. Shape ambiguities in structure from motion. *PAMI*, 1997. 3

[55] Feitong Tan, Hao Zhu, Zhaopeng Cui, Siyu Zhu, Marc Pollefeys, and Ping Tan. Self-supervised human depth estimation from monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 650–659, 2020. 2

[56] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment networks. In *ICLR*, 2018. 1, 2, 5, 6, 7

[57] Tina Yu Tian, Carlo Tomasi, and David J Heeger. Comparison of approaches to egomotion computation. In *CVPR*, 1996. 3

[58] 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. 2

[59] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*. Springer, 1999. 1, 3

[60] Benjamin Ummenhofer and Thomas Brox. Global, dense multiscale reconstruction for a billion points. In *ICCV*, 2015. 5

[61] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, 2017. 1, 2, 5, 6, 7

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 4

[63] Andrea Vedaldi, Gregorio Guidi, and Stefano Soatto. Moving forward in structure from motion. In *CVPR*, 2007. 3

[64] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018. 2

[65] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *ECCV*, 2020. 1, 2, 5, 6, 7

[66] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013. 5

[67] Jiaolong Yang, Hongdong Li, and Yunde Jia. Optimal essential matrix estimation via inlier-set maximization. In *ECCV*, 2014. 2

[68] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*, 2020. 2

[69] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 5, 6

[70] Chen Zhao, Zhiguo Cao, Chi Li, Xin Li, and Jiaqi Yang. Nm-net: Mining reliable neighbors for robust feature correspondences. In *CVPR*, 2019. 5, 6

[71] Ji Zhao, Wanting Xu, and Laurent Kneip. A certifiably globally optimal solution to generalized essential matrix estimation. In *CVPR*, 2020. 2

[72] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Moving indoor: Unsupervised video depth learning in challenging environments. In *ICCV*, 2019. 2

[73] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2

[74] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 3, 5

[75] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *CVPR*, 2018. 2

[76] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *ICCV*, 2017. 2

[77] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *CVPR*, 2018. 2

[78] Bingbing Zhuang, Quoc-Huy Tran, Gim Hee Lee, Loong Fah Cheong, and Manmohan Chandraker. Degeneracy in self-calibration revisited and a deep learning solution for uncalibrated slam. In *IROS*, 2019. 2

[79] Jacek Zienkiewicz, Andrew Davison, and Stefan Leutenegger. Real-time height map fusion using differentiable rendering. In *IROS*, 2016. 2