2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-7281-6212-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/IROS45743.2020.9341229

Deep Keypoint-Based Camera Pose Estimation with Geometric Constraints

You-Yi Jau*[1] yjau@eng.ucsd.edu
Rui Zhu*[1] rzhu@eng.ucsd.edu
Hao Su[1] haosu@eng.ucsd.edu
Manmohan Chandraker[1] mkchandraker@eng.ucsd.edu

***Abstract*— Estimating relative camera poses from consecutive frames is a fundamental problem in visual odometry (VO) and simultaneous localization and mapping (SLAM), where classic methods consisting of hand-crafted features and sampling-based outlier rejection have been a dominant choice for over a decade. Although multiple works propose to replace these modules with learning-based counterparts, most have not yet been as accurate, robust and generalizable as conventional methods. In this paper, we design an end-to-end trainable framework consisting of learnable modules for detection, feature extraction, matching and outlier rejection, while directly optimizing for the geometric pose objective. We show both quantitatively and qualitatively that pose estimation performance may be achieved on par with the classic pipeline. Moreover, we are able to show by end-to-end training, the key components of the pipeline could be significantly improved, which leads to better generalizability to unseen datasets compared to existing learning-based methods.**

I. Introduction

Camera pose estimation has been the key to Simultaneous Localization and Mapping (SLAM) systems. To this end, multiple methods have been designed to estimate camera poses from input image sequences, or in a simplified setting, to get the relative camera pose from two consecutive frames. Traditionally a robust keypoint detector and feature extractor, *e.g.* SIFT [1], coupled with an outlier rejection framework, *e.g.* RANSAC [2], has dominated the design of camera pose estimation pipeline for decades.

Recently there have been efforts to bring deep networks to each step of the pipeline, specifically keypoint detection [3]–[5], feature extraction [3], [4] and matching [3], [4], [6], as well as outlier rejection [6]–[8]. The potential benefit is being able to handle challenges such as textureless regions by incorporating data-driven priors. However, when combining such components to replace the classic counterparts, conventional SIFT-based camera pose estimation still significantly outperforms them by a considerable margin. This could be attributed to three basic challenges for learning-based systems. First, these learning-based methods have been individually developed for their own purposes, but never been trained and optimized end-to-end for the ultimate purpose of getting better camera poses. Geometric constraints and the final pose estimation objective are not sufficiently incorporated in the pipeline. Second, learning-based methods have over-fitting nature to the domains they are trained on. When the model is applied to a different dataset, the performance is often inconsistent across various datasets compared to SIFT and RANSAC methods. Third, our evaluation shows that existing learning-based feature detectors, which serve at the very beginning of the entire pipeline, are significantly weaker than the hand-crafted feature detectors (*e.g.* SIFT detector). This is because obtaining training samples with accurate keypoints and correspondences, at the level to surpass or just match the subpixel accuracy of SIFT, is tremendously difficult.

In face of these issues when naively putting existing learning-based methods together, we propose the end-to-end trained framework for relative camera pose estimation between two consecutive frames (Fig. 1). Our framework integrates learnable modules for keypoint detection, description and outlier rejection inspired by the geometry-based classic pipeline. The whole framework is trained in an end-to-end fashion with supervision from ground truth camera pose, which is the ultimate goal for pose estimation. Particularly, in facing the third challenge of requiring accurate keypoints for feature detector training, we introduce a `Softargmax` detector head in the pipeline, so that the final pose estimation error could be back-propagated to provide subpixel level supervision.

Experiments show that the end-to-end learning can drastically improve the performance of existing learning-based feature detectors, as well as the entire pose estimation system. We show that our method outperforms existing learning-based pipelines by a large margin, and performs on par with the state-of-the-art SIFT-based methods. We also demonstrate the significant benefit of generalizability to unseen datasets compared to learning-based baseline methods. We evaluate our model on KITTI [9] and ApolloScape [10] datasets and demonstrate not only quantitatively but also qualitatively. That is, by training end-to-end, we are able to obtain relatively balanced keypoint distribution corresponding to appearance and motion patterns in the image pair.

To summarize, our contributions include:

- We propose the keypoint-based camera pose estimation pipeline, which is end-to-end trainable with better robustness and generalizability than the learning-based baselines.
- The pipeline is connected with the novel `Softargmax` bridge, and optimized with geometry-based objectives obtained from correspondences.
- The thorough study on cross-dataset setting is done to evaluate generalization ability, which is critical but not much discussed in the existing works.

We describe our pipeline in detail in Sec. III with the design of the loss functions and training process. We show the quantitative results of pose estimation and qualitative results in Sec. IV. Code will be made available at `https://github.com/eric-yyjau/pytorch-deepFEPE`.

*Equal contribution
[1] The authors are with University of California, San Diego

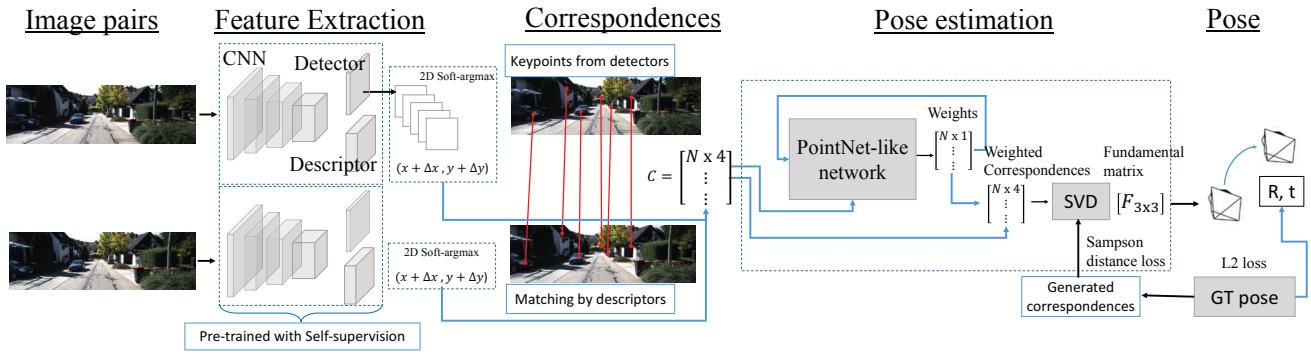

Fig. 1: **Overview of the system.** A pair of images is fed into the pipeline to predict the relative camera pose. Feature extraction predicts detection heatmaps and descriptors for finding sparse correspondences. Local 2D `Softargmax` is used as a bridge to get subpixel prediction with gradients. Matrix $\mathbf{C}$ of size $N \times 4$ is formed from correspondences. $\mathbf{C}$ is the input for pose estimation, where the PointNet-like network predicts weights for all correspondences. Weighted correspondences are passed through SVD to find fundamental matrix F, which is further decomposed into poses. Ground truth poses (GT poses) are used to compute L2 loss between rotation and translation (**pose-loss**). Correspondences generated from GT poses are used to compute fundamental matrix loss (**F-loss**). See more details in Sec. III.

II. Related Work

Geometry-based visual odometry Visual odometry (VO) is a well-established field [11]–[13], which estimates camera motion between image frames. This line of research can be separated into two main groups, feature-based methods and direct methods. For feature-based methods, *e.g.* [14], [15], sparse keypoints for images are detected and described in order to form a set of correspondences. The correspondences are then used for pose estimation using 8-point algorithm [16], PnP [17], or optimized jointly with pose using bundle adjustment [18]. Due to the presence of localization noise and outliers, RANSAC [2] is a popular choice for outlier rejection. However, the method struggles in case of textureless or repetitive patterns, where keypoints are noisy and difficult to match, as it only uses sparse features across the image and strives to find a good subset of them. This issue motivated the direct methods [12], [19], which maximizes photometric consistency over all pixel pairs. However, the method suffers in dynamic scenes or challenging lighting environments. Methods combining sparse feature-based methods and direct methods are proposed in recent years, *e.g.* [11], [13], [20], and in addition, loop closure and bundle adjustment (BA) have been applied to extend VO to simultaneous localization and mapping (SLAM).

Learning-based visual odometry Deep learning for VO has developed rapidly in recent years, *e.g.* [21]–[29], taking advantage of convolutional neural networks (CNN) for better adaptation to specific domains. For the monocular camera setting, CNN-SLAM [21] claims that learned depth prediction helps in textureless regions and corrects the scale. PoseNet [24] utilizes CNN to predict a 6 DoF global pose and claims the ability to adapt to a new sequence with fine-tuning. To take advantage of temporal information, DeepVO [22], [30] proposes to use recurrent neural networks (RNN) to predict poses along the sequence. The most recent work is [31] which brings learnable memory and refinement modules into the framework. For the sparse feature-based category, some work has been done using learning-based methods, *e.g.* [3], [32]–[34]. In [32], the feature descriptor of a two-layer shallow network is combined with the SLAM pipeline. In addition, SuperPoint [3], [33], which is a learned feature extractor, is combined with BA to update the stability score for each point. However, learned feature extractor is employed in an off-the-shelf manner, which may not be optimal from an end-to-end perspective.

Learning-based feature extraction and matching Feature extraction, which consists of keypoint detection and description, has been utilized in a variety of vision problems. Traditionally, detectors [1], [35], [36] and descriptors [1], [36] are mostly designed by heuristics. SIFT [1], which utilizes the Gaussian feature pyramid and descriptor histograms, has achieved success over the past decade. In recent years, deep learning has been utilized to build up feature extractors, *e.g.* [3]–[5]. To our knowledge, LIFT [5] is the first end-to-end pipeline, which consists of a SIFT-like procedure with sliding window detection and is trained on ground truth generated from SIFT and SfM [37]. LF-net [4] optimizes keypoints and correspondences with gradients using ground truth camera poses and depth. SuperPoint [3] proposes a self-supervised pipeline to train detectors and descriptors at the same time and beats SIFT in HPatches [38] evaluation, with some follow-up works [39], [40]. However, all of the feature extractors are not optimized in together with the overall VO system, leading to suboptimal performance. Also, their evaluation metrics, *i.e.* matching score, does not necessarily reflect the performance of pose estimation in a VO task.

Learning-based camera pose estimation Learning-based methods for camera pose estimation have been gaining attention in recent years. Following direct methods, [23], [41]–[46] take advantage of geometric constraints of 3D structure, and jointly estimate depth and pose in an unsupervised manner using photometric consistency. Poursaeed *et al.* [47] uses Siamese networks [48] to regress the fundamental matrix between left and right views through the sequence. For feature-based pipeline, [6] makes a differentiable sampling-based version of RANSAC, while [8], [49] utilize PointNet-like architecture [50] to weigh each input correspondence and

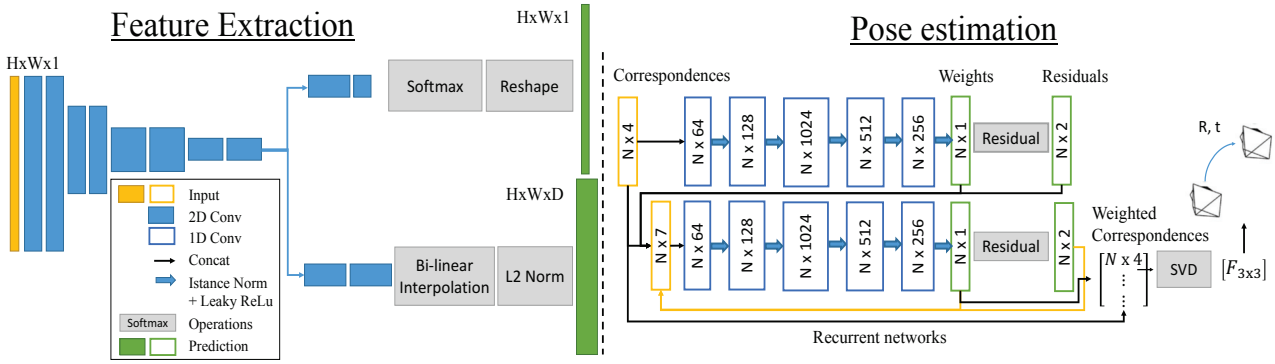

Fig. 2: **Network structure of our feature extraction (FE) and pose estimation(PE) modules.** FE module [3] has VGG-like structure, with gray images as input, and detection and description heatmaps as output. PE module has N correspondences ($\mathbf{C}$) as input, with several layers of 1D convolution to initialize weights and compute residuals. The weights, residuals and correspondences are fed into the RNN with the same structure for D iterations (D=5). From the final correspondences with weights, the fundamental matrix is estimated.

subsequently solve for the camera pose. These methods retain the mathematical and geometric constraints from classic methods, therefore can be more generalizable than direct prediction from image appearance.

III. Method

A. Overview

We propose a deep feature-based camera pose estimation pipeline called `DeepFEPE` (Deep learning-based Feature Extraction and Pose Estimation), which takes two frames as input and estimates the relative camera pose. The pipeline mainly consists of two learning-based modules, for feature extraction and pose estimation respectively, as shown in Fig. 1.

Instead of naive concatenation of the modules, careful designs are made for training `DeepFEPE` end-to-end, which includes `Softargmax` [51] detector head and geometry-embedded loss function. The `Softargmax` detector head equips the feature detection with sub-pixel accuracy, and enables gradients from pose estimation to flow back through the point coordinates. For loss function, we not only regress a fundamental matrix, but also directly constrain the decomposed poses by enforcing a geometry inspired L2 loss on the estimated rotation and translation, which leads to better prediction and generalization ability as shown in Sec. IV. We include more details for `DeepFEPE` and the network structures in Fig. 2.

Notation We refer to the pair of images as $I, I' \in \mathbb{R}^{H\times W}$, the transformation matrix from frame i to j as $T_{ij} = [R|t]$, where $R \in \mathbb{R}^{3\times 3}$ is the rotation matrix and $t \in \mathbb{R}^{3\times 1}$ is the translation vector. We refer to a point in 2D image coordinates as $p \in \mathbb{R}^2$, where $p = [u, v]$.

B. Feature Extraction (FE)

We use learning-based feature extraction (FE), namely SuperPoint [3], in our pipeline. SuperPoint is chosen as our base component because it is trained with self-supervision and demonstrated top performance for homography estimation in HPatches dataset [38]. Similar to traditional feature extractors, *e.g.* SIFT, SuperPoint serves as both the detector and descriptor, with the input gray image $I \in \mathbb{R}^{H\times W\times 1}$, and output keypoint heatmap $H_{det} \in \mathbb{R}^{H\times W\times 1}$ and descriptor $H_{desc} \in \mathbb{R}^{H\times W\times D}$. SuperPoint consists of a fully-convolutional neural network with a shared encoder and two decoder heads as the detector and descriptor respectively, as shown in Fig. 2.

1) `Softargmax` Detector Head: To overcome the challenge of training end-to-end, we propose detector head with 2D `Softargmax`. In the original Superpoint, non-maximum suppression (NMS) is applied to the output of keypoint decoder H_{det} to get sparse keypoints. However, the direct output from NMS only has pixel-wise accuracy and is non-differentiable. Inspired by [4], we apply `Softargmax` on the 5×5 patches extracted from the neighbors of each keypoint after NMS. The final coordinate of each keypoint can be expressed as

$$(u', v') = (u_0, v_0) + (\delta u, \delta v), \tag{1}$$

where in a given 2D patch,

$$\delta u = \frac{\sum_j \sum_i e^{f(u_i,v_j)} i}{\sum_j \sum_i e^{f(u_i,v_j)}}, \delta v = \frac{\sum_j \sum_i e^{f(u_i,v_j)} j}{\sum_j \sum_i e^{f(u_i,v_j)}}. \tag{2}$$

$f(u, v)$ denotes the pixel value of the heatmap at position (u, v), and i, j denotes the relative directions in x, y-axis with respect to the center pixel (u_0, v_0). The integer-level keypoint (u_0, v_0) is therefore updated to (u', v') with subpixel accuracy.

The output of the `Softargmax` enables flow of gradients from the latter module to the front, in order to refine the coordinates for subpixel accuracy. To pre-train the FE module with `Softargmax`, we convolve the ground truth 2D detection heatmap with a Gaussian kernel σ_{fe}. The label of each keypoint is represented as a discrete Gaussian distribution on a 2D image.

2) Descriptor Sparse Loss: To pre-train an efficient FE, we adopt sparse descriptor loss instead of dense loss. Original dense loss [3] collects loss from all possible correspondences between two sets of descriptors in low resolution output, which creates a total of $(H_c \times W_c)^2$ of positive and negative pairs. Instead, we sparsely sample N positive pairs, and M negative pairs collected from each positive pair,

forming $M \times N$ pairs of sampled correspondences. The loss function is the mean contrastive loss as described in [3].

3) Output of Feature Extractor: We obtain correspondences for pose estimation from the sparse keypoints and their descriptors. To get the keypoints, we apply non-maximum suppression (NMS) and a threshold on the heatmap to filter out redundant candidates. The descriptors are sampled from H_{desc} using bi-linear interpolation. With two sets of keypoints and descriptors, 2-way nearest neighbor matching is applied to form N correspondences, forming an $N \times 4$ matrix as input for pose estimation.

C. Pose Estimation (PE)

Pose estimation takes correspondences as input to solve for the fundamental matrix. Instead of using a fully connected layer to regress fundamental matrix or pose directly as in [29], [41], [47], we embed geometric constraints, *i.e.* sparse correspondences, into camera pose estimation. To create a differentiable pipeline in replacement of RANSAC for pose estimation from noisy correspondences, we build upon the Deep Fundamental Matrix Estimation (DeepF) [8], and propose a geometry-based loss to train `DeepFEPE`.

1) Existing Objective for Learning Fundamental Matrix: DeepF [8] formulates fundamental matrix estimation as a weighted least squares problem. The weights on the correspondences indicate the confidence of matching pairs, and are predicted using a neural network model with the PointNet-like structure. Then, weights and points are applied to solve for the fundamental matrix. Residuals of the prediction, as defined in [8], are obtained from the mean Sampson distance [52] of the input correspondences. The correspondences, weights, residuals are fed into the model recurrently to refine the weights. To be more specific, the residuals $r(\mathbf{p_i}, \mathbf{F})$ are calculated as following:

$$r(\mathbf{p_i}, \mathbf{F}) = |\hat{\mathbf{p}}_\mathbf{i}^T \mathbf{F} \hat{\mathbf{p}}_\mathbf{i}'|(\frac{1}{\|\mathbf{F}^T \hat{\mathbf{p}}_\mathbf{i}\|_2} + \frac{1}{\|\mathbf{F} \hat{\mathbf{p}}_\mathbf{i}'\|_2}), \tag{3}$$

where $\mathbf{p} = (u, v, 1)$ and $\mathbf{p}' = (u', v', 1)$ denote a pair of correspondences in the homogeneous coordinates.

Following [8], the loss is defined as epipolar distances from *virtual* points on a grid, to their corresponding epipolar lines, which are generated from ground truth fundamental matrix. It is abbreviated as **F-loss** in the following sections.

2) Geometry-based Pose Loss: Due to the fact that a good estimation in epipolar space does not guarantee better pose estimation, we propose a geometry-based loss function by enforcing a loss between estimated poses and ground truth poses. The estimated fundamental matrix is converted into the essential matrix using the calibration matrix and further decomposed into 2 sets of rotation and 2 translation matrices. By picking the one camera pose where all points are in front of both cameras (which gives lowest error among all 4 possible combinations of poses), we obtain the rotation in quaternions [53] and translation vectors, and compute L2 loss as our geometry-based loss. Then, loss terms $\mathcal{L}_{rot}$ and $\mathcal{L}_{trans}$ are collected from the pose with minimum L2 loss.

$$\mathcal{L}_{rot} = min(\|q(\mathbf{R}_{est\_i}) - q(\mathbf{R}_{gt})\|_2), i = [1, 2], \tag{4}$$

$$\mathcal{L}_{trans} = min(\|\mathbf{t}_{est\_i} - \mathbf{t}_{gt}\|_2), i = [1, 2], \tag{5}$$

where R, t are decomposed from the essential matrix, and $q(.)$ converts the rotation matrix into a quaternion vector. The final loss is followed,

$$\begin{aligned}\mathcal{L}_{pose} =& min(\mathcal{L}_{rot}(\mathbf{R}_{est}, \mathbf{R}_{gt}), c_r) + \\ & \lambda_{rt} * min(\mathcal{L}_{trans}(\mathbf{t}_{est}, \mathbf{t}_{gt}), c_t),\end{aligned} \tag{6}$$

where c_r and c_t are clamping constants for losses to prevent gradient explosion. The geometry-based loss is abbreviated as **pose-loss** in the following sections.

D. Training Process

After initializing both FE and PE modules respectively, we train the pipeline end-to-end. Our FE module is trained using a self-supervised method [3]. The keypoint detector is initialized by synthetic data, which can be used to generate pseudo ground truth for detectors on any dataset with single image homography adaptation (HA). Homography warping pairs are generated on-the-fly for descriptor training [3]. We put a Gaussian filter on the ground truth heatmap to enable prediction with `Softargmax`, where $\sigma_{fe} = 0.2$. For descriptor sparse loss, we have $H_c = H/8$, $W_c = W/8$, $N = 600$, and $M = 100$. NMS window size is set to be $w = 4$. The model is trained with 200k iterations on synthetic datasets, and 50k iterations on real images.

With the correspondences from the pre-trained FE, we initialize the PE module using **F-loss**. The network has RNN structure with iteration D=5. The training converges at around 20k iterations. For training with the **pose-loss**, We set $c_r = 0.1$, $c_t = 0.5$ and $\lambda_{rt} = 0.1$.

When connecting the entire pipeline, the gradients from **pose-loss** flow back through the Pose Estimation(PE) module to update the prediction of weights, as well as the Feature Extraction(FE) module to update the locations of keypoints. The pipeline and supervision is shown in Fig. 1.

IV. Experiments

We evaluate `DeepFEPE` using pose estimation error, and compare with previous approaches. Acronyms and symbols for the approaches are defined in Tab. I. Different methods are evaluated on KITTI dataset [54], and further on ApolloScape dataset [10] to show the generalization ability to unseen data. To be noted, we evaluate for relative pose estimation with existing baselines as in Sec. IV-B, and for visual odometry as in Sec. IV-C. We demonstrate significant improvement quantitatively against baseline learning-based methods after end-to-end training, as shown in Tab. II and Tab. IV. To give an insight into the improvement of optimizing SuperPoint from **pose-loss**, we evaluate the epipolar error of correspondences quantitatively in Tab. VII and visualize the change of keypoint distribution during training in Fig. 5.

A. Datasets

We extract all pairs of consecutive frames, *i.e.* with time difference 1, for training and testing.

KITTI dataset We train and evaluate our pipeline using KITTI odometry sequences, with ground truth 6 DoF poses obtained from IMU/GPS. There are 11 sequences in total, where sequences 00-08 are used for training (16k samples) and 09-10 are used for testing (2,710 samples).

ApolloScape dataset The dataset is collected in driving scenarios, with ground truth 6 DoF poses collected from

Model References		Feature extraction		Pose estimation		Loss		Training
Categories	Symbols	Sift (Si)	Superpoint (Sp)	RANSAC (Ran)	DeepF (Df)	F-loss (f)	Pose-loss (p)	End-to-end (end)
SIFT + RANSAC (Si-base)	Si-Ran	✓		✓				
Superpoint + RANSAC (Sp-base)	Sp-Ran		✓	✓				
Baseline with Sift + DeepF (Si-models)	Si-Df-f	✓			✓	✓		
	Si-Df-p	✓			✓		✓	
	Si-Df-fp	✓			✓	✓	✓	
Ours - no end-to-end training (Sp-models)	Sp-Df-f		✓		✓	✓		
	Sp-Df-p		✓		✓		✓	
Ours - with end-to-end training (DeepFEPE)	Sp-Df-f-end		✓		✓	✓		✓
	Sp-Df-p-end		✓		✓		✓	✓
	Sp-Df-fp-end		✓		✓	✓	✓	✓

TABLE I: **The reference table for modules and losses trained for experiments.** The table lists all the baselines used in Sec. IV-B. Baselines and our models are referred to by symbols, as they consist of different FE or PE modules trained using different losses.

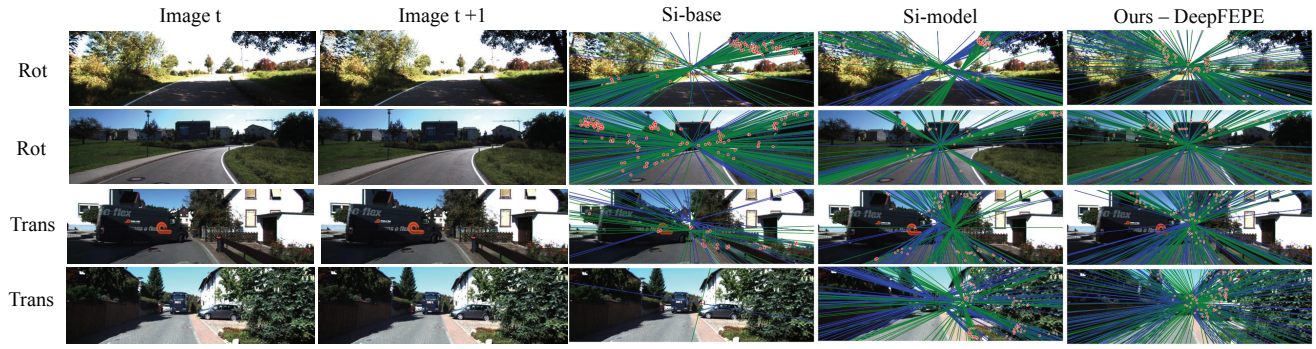

Fig. 3: **Pose estimation comparison.** The first two columns show the image pairs. The last three columns compare Si-base, Si-model and DeepFEPE. We show 2 examples for rotation and 2 for translation dominated pairs. The blue lines are plotted from ground truth fundamental matrix F_{gt}, as the green lines are from estimated F_{est}. Red dots are the keypoints from correspondences with high weights. For Si-base, the correspondences are selected by RANSAC, where keypoints around vanishing points are usually rejected. However, Si-model utilizes all correspondences to solve for the fundamental matrix, which leads to better quantitative results after training. The distribution of points in DeepFEPE is more balanced than for others, leading to more accurate pose estimation.

GPS/IMU. It includes different view angles of the camera, and lighting variations from KITTI dataset, and is used for generalization testing. We only use testing split in Road 11 for cross-dataset setting (5.8k samples).

B. Relative Pose Estimation

We evaluate the performance of DeepFEPE from the estimated rotation and translation, as in [8]. With the transformation matrix, we calculate the error by composing the inverse of the ground truth matrix with our estimation. Then, we extract the angle from the composed rotation matrix as the error term.

$$\mathbf{R}_{rel} = \mathbf{R}_{est} * \mathbf{R}_{gt}^{T}, \quad (7)$$

$$\delta\theta = \|Rog(\mathbf{R}_{rel})\|_2, \quad (8)$$

where $Rog(.) \in \mathbb{R}^{3\times3} \rightarrow \mathbb{R}^{3\times1}$ converts a rotation matrix to a Rodrigues' rotation vector. The length of the resulting vector represents the error in angle. We measure translation error by angular error between the estimated translation and the ground truth vector. Due to scale ambiguity, we set the translation vector to be unit vector.

$$\delta\mathbf{t} = cos^{-1}\left(\frac{\mathbf{t}_{est} \cdot \mathbf{t}_{gt}}{\|\mathbf{t}_{est}\|\|\mathbf{t}_{gt}\|}\right) \quad (9)$$

The equation computes the angle between the estimated translation vector and ground truth vector. With the rotation and translation error for each pair of images throughout the sequence, we compute the inlier ratio, from 0% to 100%, under different thresholds. Mean and median of the error are also computed in degrees.

We compare different models as follows. **(1) Si-base** (classic baseline models): Correspondences from SIFT are fed into RANSAC for pose estimation. **(2) Si-models** (SIFT and DeepF models): SIFT correspondences are used to estimate pose using deep fundamental matrix [8], which is the current state-of-the-art relative pose estimation pipeline. **(3) Sp-base** (SuperPoint with RANSAC): SuperPoint is pre-trained and then connected to RANSAC for pose estimation. **(4) Sp-models** (SuperPoint and DeepF models): SuperPoint is pre-trained on the given dataset and then frozen to train DeepF models. **(5)** DeepFEPE (Our method with end-to-end training): feature extraction (FE) and pose estimation (PE)

are trained jointly using **F-loss** or **pose-loss**. The reference table of the models above are shown in Tab. I, with symbols representing different training combinations. The models are trained on KITTI and evaluated on both KITTI and ApolloScape datasets.

Tab. II compares the learning-based baseline (Sp-base) with our DeepFEPE model, which shows significant improvement w.r.t. rotation and translation error. Looking into the rotation error, the pre-trained SuperPoint [3] performs poorly with RANSAC pose estimation (0.217 degrees median error), whereas the DeepF [8] module improves that to 0.078 degrees. Our DeepFEPE further improves the rotation median error to 0.041 degrees, with translation median error from 2.1 (Sp-Ran) to 0.5 degrees.

KITTI Models	KITTI dataset - error(deg.) inlier ratio↑, mean↓, median↓					
	Rotation (deg.)			Translation (deg.)		
	0.1↑	Mean.↓	Med.↓	2.0↑	Mean.↓	Med.↓
Base(Sp-Ran)	0.189	0.641	0.217	0.481	5.798	2.103
Sp-Df-f	0.633	0.100	0.078	0.830	1.476	0.846
Sp-Df-p	0.875	0.130	0.047	0.887	1.719	0.539
Ours(Sp-Df-f-end)	0.915	0.053	0.042	0.905	1.662	**0.489**
Ours(Sp-Df-p-end)	**0.932**	**0.050**	**0.041**	0.905	1.600	0.503
Ours(Sp-Df-fp-end)	0.910	0.054	0.048	**0.917**	**1.062**	0.504

TABLE II: **Comparison of pose estimation for learning-based KITTI models on KITTI dataset.** The set of models are trained on KITTI with learning-based feature extraction (FE). (Refer to Tab. I for acronyms.)

KITTI Models	KITTI dataset - error(deg.) inlier ratio↑, mean↓, median↓					
	Rotation (deg.)			Translation (deg.)		
	0.1↑	Mean.↓	Med.↓	2.0↑	Mean.↓	Med.↓
Base(Si-Ran)	0.818	0.391	0.056	0.899	1.895	0.639
Si-Df-f	0.938	**0.051**	0.041	0.914	1.699	**0.484**
Si-Df-p	0.901	0.059	0.044	0.903	1.472	0.513
Si-Df-fp	**0.947**	0.111	**0.038**	0.916	1.741	0.484
Ours(Sp-Df-fp-end)	0.910	0.054	0.048	**0.917**	**1.062**	0.504

TABLE III: **Comparison of pose estimation for SIFT-based KITTI models on KITTI dataset.** The table compares our DeepFEPE model with Si-base and Si-models for pose estimation. (Refer to Tab. I for acronyms.)

KITTI Models	Apollo dataset - error(deg.) inlier ratio↑, mean↓, median↓					
	Rotation (deg.)			Translation (deg.)		
	0.1↑	Mean.↓	Med.↓	2.0↑	Mean.↓	Med.↓
Base(Sp-Ran)	0.407	0.205	0.118	0.583	5.645	1.670
Sp-Df-f	0.725	0.126	0.068	0.754	2.074	1.155
Sp-Df-p	0.730	0.124	0.067	0.827	1.905	0.974
Ours(Sp-Df-f-end)	0.841	0.100	0.051	0.910	**1.122**	**0.589**
Ours(Sp-Df-p-end)	0.686	0.152	0.071	0.747	2.652	1.068
Ours(Sp-Df-fp-end)	**0.864**	**0.092**	**0.051**	**0.924**	1.275	0.659

TABLE IV: **Comparison of pose estimation for learning-based KITTI models on Apollo dataset.** The table compares the learning-based approaches in a cross-dataset setting.

In terms of other baselines, we compare DeepFEPE with Si-models and Si-base in Tab. III. DeepFEPE achieves better mean translation and rotation error compared to Si-base, and comparable performance with Si-models. The table

KITTI Models	Apollo dataset - error(deg.) inlier ratio↑, mean↓, median↓					
	Rotation (deg.)			Translation (deg.)		
	0.1↑	Mean.↓	Med.↓	2.0↑	Mean.↓	Med.↓
Base(Si-Ran)	**0.922**	0.157	**0.037**	**0.979**	**0.788**	0.388
Si-Df-f	0.845	0.172	0.043	0.895	2.452	0.389
Si-Df-p	0.727	0.333	0.056	0.760	4.918	0.658
Si-Df-fp	0.840	0.148	0.044	0.911	2.103	**0.369**
Ours(Sp-Df-fp-end)	0.864	**0.092**	0.051	0.924	1.275	0.659

TABLE V: **Comparison of pose estimation for SIFT-based KITTI models on Apollo dataset.** The table compares our DeepFEPE with other baseline methods in a cross-dataset setting.

demonstrates that the DeepFEPE model sets up the new state-of-the-art for learning-based relative pose estimation against DeepF. The qualitative results are shown in Fig. 3 and Fig. 4, comparing Si-base, Si-model and DeepFEPE. Pose estimation is visualized by comparing the epipolar lines projected from ground truth and estimated fundamental matrices. If the estimated one is close to ground truth, the vanishing point should match that of ground truth. Keypoints with high weights predicted by Pose Estimation (PE) are also plotted for reasoning the relation of point distribution and pose estimation.

Due to the fact that learning-based methods are biased towards the training data, we evaluate the models trained from KITTI on the ApolloScape dataset. The results demonstrate that our model retains generalization ability and is less prone to overfitting. From Tab. IV, we compare DeepFEPE models with Sp-base models and observe the benefit from end-to-end training with lower rotation and translation error. Without end-to-end training, the Sp-base models degrade significantly (in Tab. II) and are won over by end-to-end models by a large margin. Compared to other baselines in Tab. V, we observe that the Si-base demonstrates the highest accuracy, and DeepFEPE achieves better mean rotation and translation error over Si-models.

To further examine the benefit of geometry-based loss, we can look into Tab. II, with 3 models trained on either **F-loss**, **pose-loss** or both. We can observe the model trained using both losses achieves significantly better mean translation error. We believe this is because the geometric information incorporated in **pose-loss** encourages the keypoint distribution in FE to be pose-aware. The keypoints are updated to balance between good localization accuracy and matching w.r.t. the pose estimation. The change of keypoint distribution is observed from Fig. 5. This shows the potential of having a robust and optimized feature extractor with end-to-end training. As observed from the figure, keypoints close to the vanishing point are reduced after the end-to-end training. It is because these points are good for matching but may incur high triangulation errors when solving for camera pose, due to their little motion from frame to frame. On the other hand, points near the border of the image see a noticeable increase. These points may not be robustly matched with conventional descriptors because of large motion and in some cases motion blur. On the contrary, our method is able to reveal these points which provide a wider baseline for more accurate camera pose estimation.

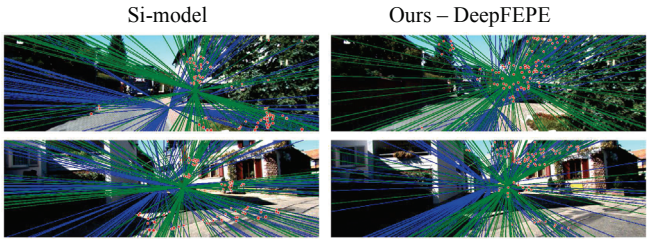

Fig. 4: **Failure cases of pose estimation.** The failure cases include over-exposed and textureless scenes. The challenging views result in noisy correspondences and wrong predictions. (Lines and dots are plotted as in Fig. 3.)

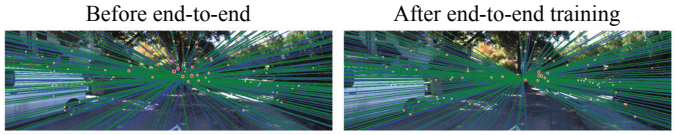

Fig. 5: **Change of keypoint distribution after end-to-end training.** To show the qualitative results of feature extractor, we freeze the pose estimation module and update only the feature extractor. (Lines and dots are plotted as in Fig. 3.)

C. Visual Odometry Evaluation

To evaluate the prediction over the whole sequence, we test the models on KITTI sequences 09, 10. We set the relative translation vectors to be unit vectors and align the trajectory with the ground truth using Sim(3) Umeyama alignment [9], [28][1]. We compute the relative translation and rotation error, as shown in Tab. VI. Our `DeepFEPE` model significantly improves the SuperPoint baseline and works comparable with the Si-base pipeline.

Methods	Seq. 09		Seq. 10	
	$t_{err}(\%)$	$r_{err}(\deg/100m)$	t_{err}	r_{err}
Base(Si-Ran)	8.842	0.512	11.508	1.447
Base(Sp-Ran)	11.005	3.324	40.021	29.599
Si-Df-fp	9.706	0.889	11.206	1.546
Ours(Sp-Df-fp-end)	8.639	0.664	11.719	0.945

TABLE VI: **Visual odometry results on KITTI.**

	KITTI - epiploar dists (n pixels), num of matches: mean/ med.					
KITTI Models	0.2	0.5	1.0	2.0	Mean	Med.
Sp-Ran	0.080	0.195	0.361	0.581	541.546	533.000
Sp-Df-f-end	0.107	0.258	0.460	0.685	719.986	703.000
Sp-Df-p-end	0.096	0.232	0.421	0.643	626.343	611.000
Sp-Df-fp-end	0.105	0.254	0.453	0.677	669.170	654.000

TABLE VII: **Superpoint evaluation.**

D. SuperPoint Correspondence Estimation

To understand how the Feature Extraction (FE) module is updated after training, we collect quantitative results using Sampson distance and demonstrate keypoint distribution qualitatively. For each pair of correspondences, we calculate the Sampson distance from Eq. (3), which indicates whether the pair of points lies close to each other's epipolar line. We show the inlier ratio w.r.t. different distance value (unit: pixel) from 0.2 to 2, as well as the number of correspondences in Tab. VII. The results show an increase of inlier ratio up to 10% with Sampson distance below 1px on KITTI dataset. The number of correspondences also increases by around 20%. The result shows that the end-to-end training improves the individual module as well. The model trained on **F-loss** has the best result under this metric, as the **F-loss** minimizes the energy in the epipolar space.

[1]https://github.com/Huangying-Zhan/kitti-odom-eval

V. Conclusion

In this paper we propose an end-to-end trainable pipeline for estimating camera poses from input image pairs. We demonstrate that our performance is on par with classic methods, and superior generalization ability to unseen data compared with existing baselines. Both qualitative and quantitative results are included in the paper to support the claim. We provide further insights into the benefits that end-to-end training brings into keypoint detection, feature extraction and pose estimation. Future work of this paper may include sequential input or keyframes with long-term temporal cues. Experiments on other datasets with different motion patterns than the driving datasets in the paper can also be explored.

ACKNOWLEDGMENT

We thank Ishit Mehta for helpful discussions. This work is supported by NSF CAREER Award 1751365.

References

[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[2] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[3] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[4] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning Local Features from Images," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 6237–6247.

[5] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 467–483.

[6] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.

[7] E. Brachmann and C. Rother, "Learning less is more-6d camera localization via 3d surface regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4654–4662.

[8] R. Ranftl and V. Koltun, "Deep Fundamental Matrix Estimation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, vol. 11205, pp. 292–309.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[10] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape Open Dataset for Autonomous Driving and its Application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[11] J. Engel, J. Sturm, and D. Cremers, "Semi-dense Visual Odometry for a Monocular Camera," in *2013 IEEE International Conference on Computer Vision*. Sydney, Australia: IEEE, Dec. 2013, pp. 1449–1456.

[12] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2320–2327, iSSN: 2380-7504, 1550-5499, 1550-5499.

[13] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 15–22.

[14] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[16] R. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, June 1997.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, Feb. 2009.

[18] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle Adjustment — A Modern Synthesis," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Berlin Heidelberg, 2000, pp. 298–372.

[19] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Computer Vision – ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 834–849.

[20] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, March 2018.

[21] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 6565–6574.

[22] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2043–2050, May 2017.

[23] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7286–7291, iSSN: 2577-087X.

[24] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[25] C. Tang and P. Tan, "BA-Net: Dense Bundle Adjustment Network," *arXiv:1806.04807 [cs]*, June 2018.

[26] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 822–838.

[27] Y. Li, Y. Ushiku, and T. Harada, "Pose graph optimization for unsupervised monocular visual odometry," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 5439–5445.

[28] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 35–45.

[29] V. Balntas, S. Li, and V. Prisacariu, "RelocNet: Continuous Metric Learning Relocalisation Using Neural Nets," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, vol. 11218, pp. 782–799.

[30] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

[31] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha, "Beyond tracking: Selecting memory and refining poses for deep visual odometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8575–8583.

[32] R. Kang, J. Shi, X. Li, Y. Liu, and X. Liu, "DF-SLAM: A Deep-Learning Enhanced Visual SLAM System based on Deep Local Features," *arXiv:1901.07223 [cs]*, Jan. 2019.

[33] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Self-Improving Visual Odometry," *arXiv:1812.03245 [cs]*, Dec. 2018.

[34] H. Zhan, C. S. Weerasekera, J. Bian, and I. D. Reid, "Visual odometry revisited: What should be learnt?" *CoRR*, vol. abs/1909.09803, 2019.

[35] M. Trajković and M. Hedley, "Fast corner detection," *Image and Vision Computing*, vol. 16, no. 2, pp. 75–87, Feb. 1998.

[36] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571, iSSN: 2380-7504, 1550-5499, 1550-5499.

[37] C. Wu, "Towards Linear-Time Incremental Structure from Motion," in *2013 International Conference on 3D Vision - 3DV 2013*, June 2013, pp. 127–134, iSSN: 1550-6185.

[38] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[39] J. Tang, R. Ambrus, V. Guizilini, S. Pillai, H. Kim, and A. Gaidon, "Self-Supervised 3D Keypoint Learning for Ego-motion Estimation," *arXiv:1912.03426 [cs]*, Dec. 2019.

[40] P. H. Christiansen, M. F. Kragh, Y. Brodskiy, and H. Karstoft, "UnsuperPoint: End-to-end Unsupervised Interest Point Detector and Descriptor," *arXiv:1907.04011 [cs]*, July 2019.

[41] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 6612–6619.

[42] Z. Yin and J. Shi, "GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose," *arXiv:1803.02276 [cs]*, Mar. 2018.

[43] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2022–2030.

[44] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3828–3838.

[45] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 240–12 249.

[46] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[47] O. Poursaeed, G. Yang, A. Prakash, Q. Fang, H. Jiang, B. Hariharan, and S. Belongie, "Deep fundamental matrix estimation without correspondences," in *The European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[48] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[49] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[50] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[51] O. Chapelle and M. Wu, "Gradient descent optimization of smoothed information retrieval metrics," *Information Retrieval*, vol. 13, no. 3, pp. 216–235, June 2010.

[52] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo, "On determining the fundamental matrix : analysis of different methods and experimental results," INRIA, Research Report RR-1894, 1993.

[53] F. Zhang, "Quaternions and matrices of quaternions," *Linear Algebra and its Applications*, vol. 251, pp. 21 – 57, 1997.

[54] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sept. 2013.