Self-Supervised 3D Keypoint Learning for Ego-motion Estimation

Jiexiong Tang\textsuperscript{1,2,∗} Rares Ambruș\textsuperscript{1,*} Vitor Guizilini\textsuperscript{1} Sudeep Pillai\textsuperscript{1} Hanme Kim\textsuperscript{1} Adrien Gaidon\textsuperscript{1}
\textsuperscript{1}Toyota Research Institute \textsuperscript{2}KTH Royal Institute of Technology

\{firstname.lastname\}@tri.global \textsuperscript{2}jiexiong@kth.se

Abstract

Generating reliable illumination and viewpoint invariant keypoints is critical for feature-based SLAM and SfM. State-of-the-art learning-based methods often rely on generating training samples by employing homography adaptation to create 2D synthetic views. While such approaches trivially solve data association between views, they cannot effectively learn from real illumination and non-planar 3D scenes. In this work, we propose a fully self-supervised approach towards learning depth-aware keypoints purely from unlabeled videos by incorporating a differentiable pose estimation module that jointly optimizes the keypoints and their depths in a Structure-from-Motion setting. We introduce 3D Multi-View Adaptation, a technique that exploits the temporal context in videos to self-supervise keypoint detection and matching in an end-to-end differentiable manner. Finally, we show how a fully self-supervised keypoint detection and description network can be trivially incorporated as a frontend into a state-of-the-art visual odometry framework that is robust and accurate.\textsuperscript{†}

1. Introduction

Detecting interest points in RGB images and matching them across views is a fundamental capability of many robotic systems. Tasks such as Structure-from-Motion (SfM) [2], Visual Odometry (VO) or Simultaneous Localization and Mapping (SLAM) [9] assume that salient keypoints can be detected and re-identified in diverse settings, which requires strong invariance to lighting, viewpoint changes, scale etc. Until recently, these tasks have mostly relied on hand-engineered keypoint features [33, 44] which have been limited in performance. Deep learning has recently revolutionized many computer vision applications in the supervised setting [24, 49, 28, 47], however, these methods rely on strong supervision in the form of ground-truth labels which are often expensive to acquire. Moreover, supervising interest point detection is unnatural, as a human annotator cannot readily identify salient regions in images as well as key signatures or descriptors, which would allow their re-identification in diverse scenarios. Inspired by recent approaches to keypoint learning [13, 10, 4], we propose a fully self-supervised approach that exploits the temporal context in videos to learn to extract accurate and robust 3D keypoints from a single monocular image (Figure 1).

Our main contribution is a fully self-supervised framework for the learning of depth-aware keypoint detection and description purely from unlabeled videos. We propose a novel framework for the simultaneous learning of keypoint

\textsuperscript{†}Video: https://www.youtube.com/watch?v=4hFhSD8QUPM
\textsuperscript{†}Code: https://github.com/TRI-ML/KP3D

\textsuperscript{∗}Equal contribution. This work was part of an internship stay at TRI.
In [13] the authors propose SuperPoint: a self-supervised and extracts keypoints from the top and bottom quantiles. Our second contribution is the introduction of 3D multi-view adaptation, a novel adaptation technique that exploits the temporal context in videos to further boost the repeatability and matching performance of the keypoint network. For our final contribution, we show how our self-supervised depth-aware keypoint networks can be incorporated as a front-end into a visual odometry framework, enabling robust and accurate ego-motion estimation results. We show that when integrating our method with a state-of-the-art tracking method such as Direct Sparse Odometry (DSO) [15], we achieve long-term tracking results which are especially on par with state-of-the-art stereo-methods such as DVSO [53]. Through extensive experiments and ablative analysis, we show that the proposed self-supervised keypoint learning adaptation achieves state-of-the-art results on challenging benchmarks for keypoint detection, matching and visual odometry.

2. Related Work

Until recently, handcrafted image features such as SIFT [33] or ORB [44] have been the key enabler of feature-based SLAM [36] and SFM applications [1]. State-of-the-art learning-based keypoint detectors and descriptors, however, have increasingly been demonstrating improved performance on challenging benchmarks [13, 10, 45, 4], setting a new standard for keypoint-based applications.

Learning-Based Methods for Keypoint Estimation Rosten and Drummond [42, 43] pioneered the detection of learning-based image features by learning a decision tree over image patches and accurately classifying corner features with real-time considerations. In TILDE [51], the authors introduced piece-wise linear regression models to detect illumination-invariant features. LIFT [54] uses an off-the-shelf SFM algorithm to generate more realistic training data under extreme viewpoint configurations, and learns to describe features that are robust to significant viewpoint and illumination differences. In LF-Net [38], the authors introduced an end-to-end differentiable network which estimates position, scale and orientation of features by jointly optimizing the detector and descriptor in a single module.

More recently, Quad-networks [46] introduced an unsupervised keypoint learning method that learns to rank invariant interest points under diverse image transformations and extracts keypoints from the top and bottom quantiles. In [13] the authors propose SuperPoint: a self-supervised framework aimed at keypoint learning where they used a shared-encoder with detector and descriptor heads to predict interest points and descriptors simultaneously. In their work, the authors introduce Homographic Adaptation, multi-scale homography-based augmentation approach to boosting interest point detection repeatability and cross-domain generalization using synthetic datasets. Building on this work, UnsuperPoint [10] proposed a similar method for efficient keypoint detection and description trained in a fully self-supervised manner without the need for pseudo ground-truth keypoints.

Other works including Self-Improving Visual Odometry algorithm [12], take advantage of classical SFM techniques to classify the stability and repeatability of keypoints based on their re-projection error. However, due to the non-differentiable nature of their method, training these models requires multiple iterations of updates with diminishing improvements to the keypoint model. Most recently, in [4], the authors incorporate an end-to-end differentiable and neurally-guided outlier-rejection mechanism (IO-Net), that explicitly generates an additional proxy supervisory signal for the matching keypoint-pairs. This allows keypoint descriptions to be further refined, as a result of the outlier-rejection network predictions occurring during the two-view matching stage.

Learning-based Methods for Visual Odometry Self-supervised methods for depth and ego-motion estimation are becoming increasingly popular, as accurate ground-truth measurements rely heavily on expensive and specialized equipment such as LiDAR and Inertial Navigation Systems (INS). One of the earliest works in self-supervised depth estimation [19] used the photometric loss as proxy supervision to learn a monocular depth network from stereo imagery. Zhou et al. [58] extended this self-supervision to the generalized multi-view case, leveraging constraints typically incorporated in SFM to simultaneously learn depth and camera ego-motion from monocular image sequences.

Several works have extended this work further - engineering the loss function to handle errors specifically in handling outliers. However, it has been shown that direct pose estimation (i.e. directly from input images [26]) is prone to over-fitting and benefits from feature sparsification as shown in [3]. Teed and Deng [48] proposed an iterative method to regress dense correspondences from pairs of depth frames and compute the 6-DoF estimate using a PnP [29] algorithm. More recently, the authors of [35] use a model-based pose estimation solution via Perspective-n-Point to recover 6 DoF pose estimates from monocular videos and use the estimate as a form of supervision to enable semi-supervised depth learning from unlabeled videos and LiDAR. Our work borrows a similar concept, however, we take advantage of the model-based PnP solution and the inliers established to outfit a fully differentiable pose estimation module within the 3D keypoint learning frame-
work. [56] uses PnP along with an estimation of the essential matrix to compute the ego-motion, however they rely on estimating the dense flow using multiple frames, while our method focuses on sparse keypoint detection and optimization using a single frame.

3. Self-Supervised 3D Keypoint Learning for Ego-motion Estimation

In this section, we introduce the fully self-supervised framework for monocular depth-aware keypoint learning for the task of ego-motion estimation. Notably, we perform depth-aware keypoint learning purely from watching large volumes of unlabeled videos, without any need for supervision in the form of ground-truth or pseudo ground-truth labels. As a consequence of learning the 2D-to-3D keypoint lifting function from monocular videos, we show that this capability can be additionally used to accurately estimate the ego-motion between temporally adjacent images. We illustrate the proposed monocular SfM-based keypoint learning framework in Figure 2.

3.1. Notation

We formulate monocular depth-aware keypoint learning as follows: Given an input monocular image $I$, we aim to regress keypoint locations $p$, descriptors $f$, and scores $s$ along with dense depth map $D$. Functionally, we define 3 components in our framework that are used to enable depth-aware keypoint learning in an end-to-end differentiable setting: (i) $\text{KeypointNet } f_p: I \rightarrow k = \{p, f, s\}$ that learns to regress $N$ output keypoint locations $p \in \mathbb{R}^{2 \times N}$, descriptors $f \in \mathbb{R}^{256 \times N}$ and scores $s \in \mathbb{R}^N$ given an input image $I \in \mathbb{R}^{3 \times H \times W}$. (ii) $\text{DepthNet } f_D: I \rightarrow D$, that learns to predict the scale-ambiguous dense depth map $D = f_D(I)$, and as a result, provides a mechanism to lift the sparse 2D keypoints $p \in \mathbb{R}^{2 \times N}$ to 3D by directly sampling from the predicted dense depth $D$, $p^3 = \pi^{-1}(p, D_t(p))$. We refer to the resulting 3D keypoints along with their associated descriptors and scores as $k^3$, (iii) A fully-differentiable ego-motion estimator $f_x(I_t, I_s) = f_x(k^3, k_s) = x_{t \rightarrow s} \in \mathbb{SE}(3)$, that predicts the relative 6-DoF rigid-body transformation between the target image $I_t$ and the source image $I_s$. We use $p^*_t$ to denote the relative warped keypoints $p_t$ from the target image $I_t$ to the source image $I_s$ via the transformation $x_{t \rightarrow s}$.

Following [58] we compute the depth at multiple scales during training, however, when referring to the associated sparse depth for a set of descriptors we refer to the depth from the scale with the highest resolution. Notably in the monocular SfM setting, the depth scale is ambiguous up to an unknown scale factor.

3.2. Adaptations for Keypoint Learning

Multi-View Adaptation Following the concept of leveraging known geometric transformations to self-supervise and boost keypoint learning [13], we introduce Multi-View Adaptation - a novel self-supervised adaptation technique that leverages epipolar constraints in two-view camera geometry for robust 3D keypoint learning. Crucially, we generalize the works of [4, 10] and self-supervise 3D keypoint learning that leverage the structured geometry of scenes in unlabeled monocular videos. An overview of the proposed pipeline is illustrated in Figure 2.

In the adaptation step, we are interested in computing the set of corresponding keypoints $p_t \leftrightarrow p^*_t$, i.e. $p_t$ from target image $I_t$ along with warped $p^*_t$ in source image $I_s$. We use predicted keypoint descriptors $k_t$ and $k_s$ in the target and source images to compute $p^3_{MV}$ via reciprocal matching in descriptor space. Given the set of corresponding keypoints $p_t \leftrightarrow p^*_t$, we compute the associated ego-motion $x_{t \rightarrow s}$ (see Section 3.3). Once $x_{t \rightarrow s}$ is known, we compute $p^*_t$ by warping $p_t$ and we induce a combination of dense photometric losses via image-synthesis and sparse geometric losses via re-projection in the monocular two-view setting.

Specifically, we use (i) a dense photometric loss based on the warped projection of $D_t$ in $I_s$ aimed at optimizing the dense depth prediction by the DepthNet; and (ii) a sparse geometric loss aimed at minimizing the re-projection error between corresponding keypoints $k^3_t$ and $k^3_s$ predicted by the KeypointNet.

Homography Adaptation Following [4, 10], the KeypointNet is additionally trained on image pairs $(I_t, I_s)$ related through a known homography transformation $H$ which warps pixels from the source image to the target image. As a result, the training image pairs are generated by randomly sampling $H$ from a set of predefined homographies. For every warped keypoint in $H(p_t)$, we compute the corresponding keypoint from $p_s$ based on Euclidean distance, and denote the resulting set as $p^h_{HA}$. The resulting...
correspondence set is then directly used in self-supervising the keypoints by imposing a loss on the consistency of known keypoint pair matches. Figure 3 to compare the two adaptation techniques.

3.3. Pose Estimation from 3D Keypoints

Having computed correspondences \( p_t \in I_t \) and \( p_{MV}^s \in I_s \), we utilize a robust estimator to compute the 6-DoF rigid-body pose transformation \( x_{t \rightarrow s} \) between the target and source views.

**Pose Estimation via Perspective-n-Point** By lifting the 2D keypoints from the target image to 3D with the associated depth \( D_t \), we use the PnP algorithm [29] to compute the initial relative pose transformation \( x_{t \rightarrow s}^0 = (R_0 t_0) \) to geometrically match the keypoints in the target image to those in the source image. Specifically, we minimize:

\[
E_{pnp}(x_{t \rightarrow s}) = \sum \left\| p_{MV}^s - \pi(R p^d_t + t) \right\|,
\]

where \( \pi(\cdot) \) is the standard pinhole camera projection model used the project the warped points \( (R p^d_t + t) \) on to the source image \( I_s \).

The estimated relative pose \( x_{t \rightarrow s}^0 \) is obtained by minimizing the residual error in Equation (1) using the Gaussian-Newton (GN) method (see supplementary material) with RANSAC to ensure robustness to outliers. This step allows us to compute the pose robustly, however, this makes the pose no longer differentiable with respect to the 3D keypoints used to estimate it. To alleviate this limitation, we address how the resulting pose estimate can be used as an initial-guess to an end-to-end differentiable pose estimation module within the proposed self-supervised 3D keypoint learning framework.

**Differentiable Pose Estimation from 3D Keypoints** Inspired by recent monocular direct methods that perform frame-to-keyframe tracking[16], we show that by calculating the re-projected source 3D keypoints \( d_s \) from the target keypoints \( p^d_t \) via the initial pose estimate \( x_{t \rightarrow s}^0 = (R_0 t_0) \), a 3D residual can be formulated to recover the pose in closed-form for the established inlier set in PnP:

\[
r(R, t) = \left\| p^s_p - R p^d_t + t \right\|,
\]

where \( p^s_p = \pi^{-1}(p^s, D_t(p^s)) \),

\[
p^d_t = \pi^{-1}(p^d_t, d_s),
\]

\[
d_s = \left\| R_0 \cdot p^d_t + t_0 \right\|.
\]

The 3D residual above can be effectively minimized by estimating the rotation and translation separately using a closed-form solution on the established inlier set. We first estimate the rotation by subtracting the means of the points and minimizing the eq. (3) by solving an SVD in closed-form (otherwise known as the Orthogonal Procrustes problem [57]):

\[
r(R) = \left\| \tilde{p}^d_s - R \tilde{p}^d_t \right\|_2,
\]

where \( \tilde{p}_t = p^d_t - \bar{p}^d_t \),

\[
U, s, V = \text{SVD}(\sum (\tilde{p}_t^d)^T (\tilde{p}_t^d)),
\]

\[
\hat{R} = V U^T.
\]

Once the rotation \( R \) is computed, the translation \( t \) can be directly recovered by minimizing:

\[
\hat{t} = \tilde{p}_s^d - R \tilde{p}_t^d.
\]

Thus, the gradients for the pose rotation and translation can be effectively propagated with respect to the lifted 3D keypoint locations, making the overall pose estimation fully-differentiable. The differentiable pose estimated using the 2D keypoints from the source image and 3D keypoints from the target image tightly couples keypoint and depth estimation, thereby allowing both predictions to be further optimized using overall keypoint learning objective.

3.4. Keypoint Learning Objective

In this work, we self-supervise the learning of depth-aware keypoints in a fully end-to-end differentiable manner
using a combination of photometric and geometric losses. We optimize both the KeypointNet and DepthNet jointly using the following losses:

**Keypoint Loss** Based on the descriptor-matched correspondences \( p_i \leftrightarrow p_i^* \) and the 3D adapted keypoints \( p_i \leftrightarrow p_i^{MV} \), we define a loss term that enforces geometric consistency between the 2D keypoints in the source view and the 3D keypoints in the target view of the same scene:

\[
L_p = \sum ||p_i^* - p_i^{MV}||_2 .
\]  

(6)

**Descriptor Loss** Following [4], we use nested hardest sample mining to self-supervise the keypoint descriptors between the two views. Given anchor descriptors \( f_s \) from the target frame and their associated positive descriptors in the \( f_s = f_i^{MV} \) in the source frame, we define the triplet loss:

\[
L_f = \sum \max(0, ||f_s + m|| - ||f_s + m||) ,
\]  

(7)

where \( f_s \) is the hardest descriptor sample mined from \( f_s \) with margin \( m \).

**Score Loss** The score loss is introduced to identify reliable and repeatable keypoints in the matching process. In particular, we want to ensure that (i) the feature-pairs have consistent scores across matching views; and (ii) the network learns to predict high scores for good keypoints with low geometric error and strong repeatability. This objective is achieved by minimizing the squared distance between scores for each matched keypoint-pair, and minimizing or maximizing the average score of a matched keypoint-pair if the distance between the paired keypoints is greater or less than the average distance respectively:

\[
L_s = \sum \left[ \frac{(s_s + s_i^{MV})}{2} \cdot (d(p_i^*, p_i^{MV}) - d) + (s_i - s_i^{MV})^2 \right],
\]  

(8)

where \( s_i \) and \( s_i^{MV} \) are the scores of the source and target frames respectively, and \( d \) is the average reprojection error of associated points in the current frame, given by \( d = \sum d(p_i^*, p_i^{MV}) \). Here, \( d \) refers to the 2D Euclidean distance in feature space between the \( L \) matching keypoints.

We define similar keypoint, descriptor and score loss terms in the Homography Adaptation (HA) case using the \( p_i^{HA} \) correspondences.

**Photometric Loss** In addition to the geometric losses, we impose a dense photometric loss to learn dense depth in the DepthNet. Following [18, 58, 25], we warp the depth from the target from \( D_t \) along via the predicted ego-motion estimate \( x_{ego} \) to the source frame \( I_s \), and impose a structural similarity (SSIM) loss [52] between the synthesized target image \( \hat{I}_t \) and the original target image \( I_t \). The resulting dense photometric loss is regularized with an L1 pixel-wise loss term (See Appendix for more details):

\[
L_D(I_t, \hat{I}_t) = \alpha \frac{1 - SSIM(I_t, \hat{I}_t)}{2} + (1 - \alpha) |I_t - \hat{I}_t| .
\]  

(9)

To account for parallax errors and the presence of dynamic objects in videos, we compute the pixel-wise minimum between the set of synthesized source images \( I_s \in I_s \) (i.e. context images) and the target image \( I_t \) [21].

\[
L_D(I_t, I_s) = \min_{I_s} L_D(I_t, \hat{I}_t) .
\]  

(10)

In addition, we mask out static pixels by removing those which have a warped photometric loss \( L_D(I_t, \hat{I}_t) \) higher than their corresponding unwarped photometric loss \( L_D(I_t, I_s) \), calculated using the original source image without view-synthesis [21]. This has the effect of removing pixels with non-changing appearances, including static frames and dynamic objects with no relative motion.

\[
M_D = \min_{I_s} L_D(I_t, I_s) > \min_{I_s} L_D(I_t, \hat{I}_t) .
\]  

(11)

**Depth Smoothness Loss** In order to regularize the depth in texture-less low-image gradient regions, we also incorporate an edge-aware term similar to [19]:

\[
L_{sm} = |\delta_x D_t|e^{-|\delta_x I_t|} + |\delta_y D_t|e^{-|\delta_y I_t|} .
\]  

(12)

**Depth Consistency** Recall that the depth we regress \( D_t \) is scale-ambiguous. While recovering scale-consistent depth is not a strict requirement for the proposed framework to learn 3D keypoints, scale-consistency has been shown to be crucial for tasks that involve accurate ego-motion estimation [7, 22]. To this end, we incorporate a depth consistency term that discourages scale-drift between dense depth predictions in adjacent frames:

\[
L_c = \frac{||D_t(p_t) - D_s(p_i^{MV})||}{D_t(p_t) + D_s(p_i^{MV})} .
\]  

(13)

Note that \( L_c \) is a sparse loss defined based on the correspondences \( p_i \leftrightarrow p_i^{MV} \).

**Overall Objective** The overall objective used to simultaneously optimize the KeypointNet and DepthNet becomes:

\[
L = \alpha (L_p + L_f + L_s) + \beta (L_D + L_{sm}) ,
\]  

(14)

where \( \alpha \) and \( \beta \) are weights used to balance the depth and keypoint losses, and they are chosen as 0.2 and 1.0 respectively during the training.
4. Deep Semi-Direct Sparse Odometry

In this section, we will explain how the fully self-supervised depth-aware keypoint network can be incorporated as the front-end into a visual SLAM framework. We show that integrating our method into a state-of-the-art monocular visual SLAM framework such as DSO [15], we are able to achieve long-term tracking results which are especially on par with stereo methods such as DVSO [53] or ORB-SLAM2 [37]. Unlike other monocular visual odometry approaches, the superior keypoint matching and stable 3D lifting performance of our proposed method allows us to bootstrap the tracking system, rejecting false matches and outliers and avoiding significant scale-drift as demonstrated in Figures 5 and 1.

Figure 4 shows the whole pipeline of our Deep Semi-Direct Sparse Odometry (DS-DSO) system which is built on top of the windowed sparse direct bundle adjustment formulation of DSO. As illustrated, we improve depth-initialization of keyframes in the original DSO implementation by using the depth estimated through our proposed self-supervised 3D keypoint network. In addition, we modify the hand-engineered direct semi-dense tracking component to our proposed sparse and robust learned keypoint-based method introduced in this work.

5. Experiments

5.1. Datasets

We evaluate our system on the KITTI [17] dataset. We follow the training protocol of [53] and train on KITTI odometry sequences 01, 02, 06, 08, 09 and 10, and evaluate on sequences 00, 03, 04, 05 and 07. We report $r_{rel}$ - average translational RMSE drift (%) on trajectories of length 100-800m, and $r_{rot}$ - average rotational RMSE drift (deg /100m) on trajectories of length 100-800m. To evaluate the performance of our DepthNet we use the Eigen [14] test splits which consists of 697 images with associated depth (we note that the eigen test split does not overlap the KITTI odometry sequence we use for training).

To evaluate the performance of our keypoint detector and descriptor we use the HPatches [5] dataset. HPatches consists of a set of 116 image sequences (illumination and viewpoint), each sequence containing a source image and five target images, for a total of 580 image pairs. We quantify detector performance through the Repeatability and Localization Error metrics and descriptor performance through the Correctness and Matching Score metrics (the exact definition of these metrics can be found in the Appendix). For a fair comparison, we evaluate the results generated without applying Non-Maxima Suppression (NMS). Following related work [4, 10, 13], we pre-train our KeypointNet on the COCO [32] dataset, which contains 118k training images. We note that pretraining on COCO is completely self-supervised, using Homography Adaptation (more details in the following section).

5.2. Implementation Details

We implement our networks in PyTorch [40] using the ADAM optimizer [27]. We use $1e - 4$ as the learning rate and train KeypointNet and DepthNet jointly for 50 epochs with a batch size of 8. We implement KeypointNet following [4], with the mention that we use an ImageNet pre-trained ResNet-18 backbone, which we found to perform much better than the reference architecture used. We follow [18] and implement DepthNet using an ImageNet [11] pretrained ResNet-18 backbone along with a depth decoder that outputs inverse depth at 4 scales. However, at test-time, only the highest resolution scale is used for 2D-to-3D keypoint lifting.

We train on snippets of 3 images $(I_{t-Dt}, I_t, I_{t+Dt})$, for $\Delta t \in [1, 2, 4]$ with target image $I_t$ and images $(I_{t-Dt}, I_{t+Dt}) \in I_S$ as context images (otherwise referred to as source images). Using the pair of target and source images generated via 3D Multi-View Adaptation, we compute the losses as defined in Section 3.4. The dense photometric loss is computed over the context $I_S$ as shown in Equation 11. Additionally, starting from the target image $I_t$, we also perform Homography Adaptation similar to [4], e.g. translation, rotation, scaling, cropping and symmetric perspective transform. Additionally, we apply per-pixel
Gaussian noise, color jitter and Gaussian blur to the images for additional robustness to image lighting.

**Pretraining** We pre-train KeypointNet on COCO using Homography Adaptation for 50 epochs using a learning rate of 5e−4 which is later halved after 40 epochs. We refer to this as our baseline KeypointNet, and evaluate its performance in Table 1. To speed up convergence, we pretrain our DepthNet on the KITTI training sequences (e.g. 01, 02, 06, 08, 09 and 10) using the method described in [18]. We train for 200 epochs with a learning rate of 1e−4 which is decayed every 40 epochs. We refer to this as our baseline DepthNet, and we evaluate its performance in the experiments below.

### 5.3. Keypoint Detector and Descriptor Performance

Table 1 shows the performance of our keypoints and descriptors on HPatches [5]. We note that our baseline method, trained on COCO using Homography Adaptation, outperforms all classical as well as learning-based methods in terms of keypoint robustness (Repeatability and Localization Error) and descriptor matching performance (Correctness and Maching Score). As seen in the table, we show further improvements when training using the proposed 3d multi-view adaptation method. In addition to the superior VO results reported in Table 3, our method allows us to train a state-of-the-art keypoint detector with associated descriptor that can robustly detect correspondences in challenging situations. We refer the reader to the supplementary materials for additional qualitative results.

### 5.4. Visual Odometry Performance

We summarize our results and comparisons of the visual odometry performance with state-of-the-art methods in Table 3. Our method outperforms all other monocular-trained methods, as well as all other stereo-trained methods except for DVSO [53]. However, we emphasize that while DVSO is trained from a wide-baseline stereo setup which provides a very strong prior for outlier rejection, our system is trained in a fully self-supervised manner purely relying on monocular videos - a significantly harder problem. The experimental results indicate that our depth-aware keypoints provide superior matching performance that even rivals state-of-the-art methods trained on stereo imagery.

In addition, we report frame-to-frame trajectory estimation results using the PnP formulation described in Section 3.3. Notably, our frame-to-frame (F2F) method outperforms all other monocular methods except for DF-VO [56], which heavily relies on optical-flow and RANSAC-based essential matrix and hand-engineered scale-factor recovery. Comparing our F2F estimation results with PnP-trained DF-VO [56] (DF-VO PnP in Table 3), we attribute the superior performance to the direct optimization of sparse 2D-3D keypoints, as opposed to [56] which relies purely on dense optical flow. We show qualitative results of our method in Figure 5, noting that our DS-DSO results accurately follow the ground truth trajectory with minimal scale drift.

### 5.5. Ablation Study

We summarize our ablative analysis in Table 2. Our baseline - KeypointNet pre-trained on COCO and DepthNet trained on KITTI, but the two are not optimized together - shows superior results compared to most monocular methods (see Table 3), thus motivating our approach of combining keypoints and depth in a self-supervised learning framework. We notice a significant improvement when training the two networks together (Row 2: Ours – TR, DP). Adding the the differential pose estimation (Row 5: Ours – TR) further improves the performance of our system for the rrel metric; we note that the rrel metric does not improve, mostly due to an error in Sequence 04 (please refer to the supplementary material for detailed results for each version of our method on all the KITTI odometry sequence). We further ablate the KeypointNet (Row 3: Ours – TR, KPN trained) - i.e. we estimate the ego-motion using the DepthNetwork after training together with the KeypointNet, but we use the original KeypointNet trained only on COCO. We perform a similar experiment ablating the trained DepthNet (Row 4: Ours – TR, DepthNet trained).

In both cases we note a performance drop for both the

---

**Table 1. Keypoint and descriptor performance on HPatches [5].** Repeatability and Localization Error measure keypoint performance while Correctness (pixel threshold 3) and Matching score measure descriptor performance. Higher is better for all metrics except Localization Error.

| Method               | Rep | Loc | Cor-1 | M.Score | Rep | Loc | Cor-1 | M.Score |
|----------------------|-----|-----|-------|---------|-----|-----|-------|---------|
| SIFT [33]            | 0.552 | 1.429 | 0.422 | 0.218   | 0.529 | 1.450 | 0.607 | 0.204   |
| SURF [6]             | 0.491 | 1.150 | 0.702 | 0.255   | 0.466 | 1.244 | 0.745 | 0.230   |
| BRISK [19]           | 0.566 | 1.073 | 0.767 | 0.290   | 0.516 | 1.211 | 1.207 | 0.655   |
| SIFT [33]            | 0.451 | 0.855 | 0.845 | 0.364   | 0.421 | 1.011 | 0.833 | 0.265   |
| LSUNet [19]          | 0.451 | 0.855 | 0.845 | 0.364   | 0.421 | 1.011 | 0.833 | 0.265   |
| LSUNetNet [30]       | 0.538 | 1.084 | 0.726 | 0.296   | 0.523 | 1.183 | 0.745 | 0.241   |
| SuperPoint [17]      | 0.615 | 1.010 | 0.833 | 0.318   | 0.590 | 1.212 | 0.734 | 0.241   |
| UnsupPoint [19]      | 0.645 | 0.832 | 0.455 | 0.424   | 0.612 | 0.991 | 0.643 | 0.383   |

**Table 2. Ablative analysis.** We ablate from our method: TR - the tracking component (Section 4), DP - the differentiable pose component (Section 3.3), KPN trained - the trained version of the KeypointNet (i.e. using only the baseline), DN trained - the trained version of the DepthNet. All rrel results are obtained by performing a Sim(3) alignment step [23].
Table 3. Comparison of vision-based trajectory estimation with state-of-the-art methods. The Type column indicates the data used at train type. Note: All methods are evaluated on monocular data. Our results are obtained after performing a single Sim(3) alignment step [23] wrt. the ground truth trajectories. Bold text denotes the best method trained on monocular data; "*" denotes the best overall method. † and ‡ represent test and respectively train seq. for our method, as well as for [41, 53]. [58, 55, 34, 31, 59, 56] are trained on Sequences 00-08 and tested on Sequences 09 and 10. The numbers for [37] are reported from [31]. ‡ - the numbers of [18] are based on our own implementation.

6. Conclusion

In this paper, we propose a fully self-supervised framework for depth-aware keypoint learning from unlabeled monocular video, by incorporating a novel differentiable pose estimation module that simultaneously optimizes the keypoints and their depths in a structure-from-motion setting. Unlike existing learned keypoint methods that employ simple homography adaptation, we introduce multi-view adaptation that exploits the temporal context in videos to further boost the repeatability and matching performance of our proposed keypoint network. The resulting 3D keypoints and associated descriptors exhibit superior performance compared to all other traditional and learned methods, and is also able to learn from realistic non-planar 3D
Figure 5. Qualitative trajectory estimation results on the KITTI Odometry Seq. 03, 04, 05 and 07. We compare trajectory estimation results obtained via hand-engineered keypoint matching methods against our depth-aware learned keypoint matching, with a common visual odometry backend such as DSO. As illustrated in the figure, our self-supervised method is able to accurately and robustly track stable keypoints for the task of long-term trajectory estimation.

*Supplementary Materials*

**A. Architecture Diagram**

**ResNet18-DepthNet** we provide a detailed description of our DepthNet architecture in Table 4, and note that we follow [20] and use a ResNet18 encoder followed by a decoder which outputs inverse depth at 4 scales.

**ResNet18 KeypointNet** Table 5 details the network architecture of our KeypointNet. We follow [4] but change the network encoder and use a ResNet18 architecture instead, which we found to perform better.

**B. Dense Depth Evaluation**

We perform a qualitative evaluation of our DepthNet on the KITTI dataset, specifically on the Eigen [14] test split, and report the numbers in Table 6. We also include the numbers reported by [21] and note that our DepthNet baseline numbers are on par with those of [21] (note that this corresponds to row 1. Baseline of Table 2 in the main text). Table 6 also shows our numbers after fine-tuning the DepthNet and KeypointNet through the proposed Multi-View Adaptation method (note that this corresponds to row 5. Ours – TR of Table 2 in the main text). We note a slight decrease in the Abs Rel and Sq Rel metrics, but otherwise the numbers are within error margin with respect to our baseline. These results provide an important sanity check: as the main focus of this work is sparse, depth-aware keypoint learning, we don’t expect to see much variation when performing dense depth evaluation. We mention that sparsely evaluating the depth using the keypoints regressed by our method is not feasible using the depth available in the KITTI dataset: even using the denser depth maps provided by [50], only about 10% of our keypoints have valid depths in the ground truth maps, which amounts to a very small number of points (< 50) per image.

**C. Structural Similarity (SSIM) loss**

We define the SSIM loss [52] as:

\[
\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}
\]

where \(\mu_x, \mu_y\) are the means of the images \(x, y\), \(\sigma_x^2, \sigma_y^2\) are the variances of the images \(x, y\), and \(\sigma_{xy}\) is the covariance of \(x, y\), \(C_1 = (K_1L)^2\) and \(C_2 = (K_2L)^2\), with \(L\) being the dynamic range of the pixel values, and \(K_1, K_2\) are small positive constants.
### Table 5. KeypointNet diagram.

| Layer Description | K | Output Tensor Dim. |
|-------------------|---|-------------------|
| #0 Input RGB image | | 3×H×W |
| **ResidualBlock** | | |
| Conv2d + BatchNorm + ReLU | 3 | 64×H/2×W/2 |
| Conv2d + BatchNorm | | 3 |
| **KeyPoint Encoder** | | |
| #1 Conv2d (S2) + BatchNorm + ReLU | 7 | 64×H/2×W/2 |
| #2 Conv2d + BatchNorm + ReLU | 3 | 64×H/2×W/2 |
| #3 ResidualBlock (#2) x2 | | - |
| #4 Max. Pooling (×1/2) | 3 | 64×H/4×W/4 |
| #5 ResidualBlock (#3 + S2) x2 | | - |
| #6 Max. Pooling (×1/2) | 3 | 128×H/4×W/4 |
| #7 ResidualBlock (#4 + #3) x2 | | - |
| #8 Max. Pooling (×1/2) | 3 | 256×H/8×W/8 |
| #9 ResidualBlock (#5 + #4) x2 | | - |
| #10 Conv2d + BatchNorm + LReLU (#9) | 3 | 256×H/16×W/16 |
| #11 Conv2d + Upsample (#10) | | 3×H×8×W |
| #12 Conv2d + BatchNorm + LReLU | 3 | 256×H/8×W/8 |
| #13 Conv2d + Upsample(#7 #12) | | 3×H×4×W |
| #14 Conv2d + BatchNorm + LReLU | 3 | 128×H/4×W/4 |
| #15 Conv2d + Upsample (#5 #14) | | 3×H×2×W |
| #16 Conv2d + BatchNorm + LReLU | 3 | 64×H/2×W/2 |
| **KeyPoint Decoder** | | |
| #17 Conv2d + BatchNorm + LReLU (#16) | 3 | 64×H/2×W/2 |
| #18 Conv2d + Sigmoid | | 3×H×8×W |
| #19 Conv2d + Tan. Harmonic | | 3×H×4×W |
| **Score Head** | | |
| #20 Conv2d + BatchNorm + LReLU (#19) | 3 | 256×H/8×W/8 |
| #21 Conv2d + BatchNorm + LReLU (#12) | | 3×H×4×W/8 |
| **Location Head** | | |
| #22 Conv2d + BatchNorm + LReLU (#19) | 3 | 256×H/8×W/8 |
| #23 Conv2d + BatchNorm + LReLU (#12) | | 3×H×4×W/8 |
| **Descriptor Head** | | |
| #24 Conv2d + BatchNorm + LReLU (#16) | 3 | 64×H/2×W/2 |
| #25 Conv2d | | 3×H×4×W/8 |

Table 5 details the architecture of KeypointNet. *Upsample* is a nearest-neighbor interpolation operation that doubles the spatial dimensions of the input tensor. @ denotes feature concatenation for skip connections.

### D. Pose Estimation

Recall that we aim to minimize:

\[
E_{mp}^P(x) = \sum |p_{IM}^M - \pi(R \cdot p_d^I + t)|
\]

where \( R \in \mathbb{R}^{3×3} \) is the rotation matrix and \( t \in \mathbb{R}^3 \) is the translation vector. They together compose a rigid body transform \( \exp(\hat{x}) \in SE(3) \), which is defined by \( x = [\omega^T, t^T]^T \in \mathbb{se}(3) \). \( x \) is a member of the Lie algebra and is mapped to the Lie group \( SE(3) \) through the matrix exponential \( \exp(\cdot) \):

\[
\exp(\hat{x}) = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} [\omega]^× & t \end{bmatrix}
\]

where \( [\omega]^× \) is the skew-symmetric matrix of \( \omega \).

The estimated relative pose can be obtained by optimizing the residual error in Equation (16). The Gaussian-Newton (GN) method is used to solve this non-linear least square problem. GN calculates \( x \) iteratively as follows:

\[
x^{(n+1)} = x^{(n)} - (J_x^T J_x)^{-1} J_x^T r(x^{(n)})
\]

where \( J_x \) is the Jacobian matrix with respect to the residual measurements. RANSAC is performed to achieve a robust estimation, rejecting outliers with three major types against the ego-motion assumption: false-positive matching pairs, dynamic objects or points with wrong depth estimations.

### E. Detailed Results for the Pose Ablation Study

Table 7 provides detailed results on all the KITTI odometry sequences for each entry of our ablation study (Table 2 of the main text). We note that (i) the proposed contributions - Multi-View Adaptation (row 2 vs row 1) and Differentiable Pose (row 5 vs row 1) consistently improve over the baseline; and that (ii) by swapping out the KeypointNet or DepthNet trained using the proposed Multi-View Adaptation with their baseline counterparts (rows 3 and 4) results in worse performance for both the \( t_{rel} \) and \( r_{rel} \) metrics.

### F. Qualitative Results on HPatches

#### G. Keypoint Detector and Descriptor Evaluation Metrics

We follow [13] and use the **Repeatability** and **Localization Error** metrics to estimate keypoint performance and **Homography Accuracy** and **Matching Score** metrics to estimate descriptor performance. We note that for all metrics we used a distance threshold of 3. For the Homography estimation, consistent with other reported methods, we used 300 keypoints with the highest scores. Similarly, for the frame to frame tracking we selected 480 keypoints to estimate the relative pose.

**Repeatability** is computed as the ratio of correctly associated keypoints after warping onto the target frame. We consider a warped keypoint correctly associated if the nearest keypoint in the target frame (based on Euclidean distance) is below a certain threshold.

**Localization Error** is computed as the average Euclidean distance between warped and associated keypoints.

**Homography Accuracy** To compute the homography between two images we perform reciprocal descriptor
Table 6. **Quantitative performance comparison of depth estimation on the KITTI dataset** for reported depths of up to 80m. For Abs Rel, Sq Rel, RMSE and \( \text{RMSE}_{\log} \) lower is better, and for \( \delta < 1.25 \), \( \delta < 1.25^2 \) and \( \delta < 1.25^3 \) higher is better. All networks have been pre-trained on ImageNet [11] pretraining. We evaluate on the annotated KITTI depth maps from [50]. At test-time, the scale for all the methods is corrected using the median ground-truth depth from the LiDAR.

| Method                  | Abs Rel | Sq Rel | RMSE    | \( \text{RMSE}_{\log} \) | \( \delta < 1.25 \) | \( \delta < 1.25^2 \) | \( \delta < 1.25^3 \) |
|-------------------------|---------|--------|---------|--------------------------|---------------------|----------------------|----------------------|
| Monodepth2 [21]         | 0.090   | 0.545  | 3.942   | 0.137                    | 0.914               | 0.983                | 0.995                |
| DepthNet baseline       | 0.089   | 0.543  | 3.968   | 0.136                    | 0.916               | 0.982                | 0.995                |
| DepthNet finetuned       | 0.094   | 0.572  | 3.805   | 0.138                    | 0.912               | 0.981                | 0.994                |

Table 7. **Detailed results of our Pose Ablative Analysis.** Note: our results are obtained after performing a single Sim(3) alignment step [23] wrt. the ground truth trajectories.

| Method                  | Type | 01° | 02° | 06° | 08° | 09° | 10° | 00° | 03° | 04° | 05° | 07° | Train | Test |
|-------------------------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|------|
| Baseline                | Mono | 18.96 | 3.35 | 2.16 | 3.80 | 3.15 | 5.06 | 3.50 | 3.64 | 2.33 | 3.25 | 3.00 | 6.08  | 3.14 |
| Ours – TR, DP           | Mono | 20.17 | 3.37 | 2.15 | 3.01 | 2.61 | 5.39 | 2.89 | 3.10 | 2.88 | 3.09 | 2.66 | 6.12  | 2.92 |
| Ours – TR, KPN trained  | Mono | 19.09 | 3.30 | 2.23 | 3.16 | 2.84 | 5.03 | 2.83 | 3.29 | 2.02 | 3.69 | 2.58 | 5.94  | 2.88 |
| Ours – TR, DN trained   | Mono | 15.53 | 3.37 | 1.84 | 3.63 | 2.83 | 5.06 | 3.56 | 3.06 | 2.11 | 3.33 | 2.34 | 5.38  | 2.88 |
| Ours – TR               | Mono | 17.79 | 3.15 | 1.88 | 3.06 | 2.69 | 5.12 | 2.76 | 3.02 | 1.93 | 3.30 | 2.41 | 5.61  | 2.68 |
| Ours                    | Mono | 4.70  | 3.62 | 0.92 | 2.46 | 2.31 | 5.24 | 1.83 | 1.21 | 0.76 | 1.84 | 0.54 | 3.21  | 1.24 |

Figure 6. Qualitative matching results of our method on the HPatches dataset [5].

Matching and we used OpenCVs findHomography method with RANSAC, with a maximum of 5000 iterations and error threshold 3. To compute the Homography Accuracy we compare the estimated homography with the ground truth homography. Specifically we warp the image corners of the original image onto the target image using both the estimated homography and the ground truth homography, and we compute the average distance between the two sets of warped image corners, noting whether the average distance is below a certain threshold.

**Matching Score** is computed as the ratio between successful keypoint associations between the two images, with the association being performed using Euclidean distance in descriptor space.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 2

[2] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *European conference on computer vision*, pages 29–42. Springer, 2010. 1

[3] Rares Ambrus, Vitor Guizilini, Jie Li, Sudeep Pillai, and Adrien Gaidon. Two stream networks for self-supervised
ego-motion estimation. In Proceedings of the 3rd International Conference on Robot Learning (CoRL), 2019. 2, 8

[4] Anonymous. Neural outlier rejection for self-supervised key-
point learning. In Submitted to International Conference on Learning Representations, 2020. under review. 1, 2, 3, 5, 6, 7, 9

[5] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5173–5182, 2017. 6, 7, 11

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In European conference on computer vision, pages 404–417. Springer, 2006. 7

[7] 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, 2019. 5

[8] 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. arXiv preprint arXiv:1908.10553, 2019. 8

[9] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics, 32(6):1309–1332, 2016. 1

[10] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. arXiv preprint arXiv:1907.04011, 2019. 1, 2, 3, 6, 7

[11] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009. 6, 11

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Self-improving visual odometry daniel. arXiv preprint arXiv:1812.03245, 2018. 2

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 224–236, 2018. 1, 2, 3, 6, 7, 10

[14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In NIPS, pages 2366–2374, 2014. 6, 9

[15] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence, 40(3):611–625, 2017. 2, 6

[16] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. IEEE TPAMI, 2018. 4

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013. 6

[18] Clément Godard, Oisin Mac Aodha, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. arXiv preprint arXiv:1806.01260, 2018. 5, 6, 7, 8

[19] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In CVPR, volume 2, page 7, 2017. 2, 5

[20] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In CVPR, 2017. 9

[21] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. arXiv:1806.01260, 2018. 5, 9, 11

[22] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras, 2019. 5

[23] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017. 7, 8, 11

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 1

[25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In NIPS, 2015. 5

[26] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE international conference on computer vision, pages 2938–2946, 2015. 2

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 6

[28] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6399–6408, 2019. 1

[29] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EpnP: An accurate o (n) solution to the PnP problem. IJCV, 2009. 2, 4

[30] Stefan Leutenegger, Margarita Chi, and Roland Siegwart. Brisk: Binary robust invariant scalable keypoints. In 2011 IEEE international conference on computer vision (ICCV), pages 2548–2555. Ieee, 2011. 7

[31] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. UnDeepVO: Monocular visual odometry through unsupervised deep learning. arXiv preprint arXiv:1709.06841, 2017. 8

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 6

[33] David G Lowe et al. Object recognition from local scale-invariant features. In iccv, volume 99, pages 1150–1157, 1999. 1, 2, 7

[34] Chenzhu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts+: Joint learning of geometry and motion with 3d holistic understanding. arXiv preprint arXiv:1810.06125, 2018. 8
[35] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. *arXiv preprint arXiv:1807.00275*, 2018. 2

[36] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015. 2

[37] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 6, 8

[38] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In *NIPS*, 2018. 2

[39] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *Advances in Neural Information Processing Systems*, pages 6234–6244, 2018. 7

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[41] Sudeep Pillai, Rares Ambrus, and Adrien Gaidon. SuperDepth: Self-Supervised, Super-Resolved Monocular Depth Estimation. In *ICRA*, 2019. 8

[42] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 430–443, 2006. 2

[43] Edward Rosten, Reid B. Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):105–119, 2010. 2

[44] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011. 1, 2, 7

[45] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dumczyk. From coarse to fine: Robust hierarchical localization at large scale. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[46] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: Unsupervised learning to rank for interest point detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[47] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019. 1

[48] Zachary Teed and Jia Deng. Deevp2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 2

[49] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019. 1

[50] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 9, 11

[51] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 5, 9

[53] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry. In *ECCV*, 2018. 2, 6, 7, 8

[54] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned Invariant Feature Transform. In *ECCV*, 2016. 2

[55] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*, 2018. 8

[56] Huangying Zhan, Chamara Saroj Weerasekera, Jiawang Bian, and Ian Reid. Visual odometry revisited: What should be learnt? *arXiv preprint arXiv:1909.09803*, 2019. 3, 7, 8

[57] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000. 4

[58] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 2, 3, 5, 8

[59] Alex Zihao Zhu, Wenxin Liu, Ziyun Wang, Vijay Kumar, and Kostas Daniilidis. Robustness meets deep learning: An end-to-end hybrid pipeline for unsupervised learning of egomotion. *arXiv preprint arXiv:1812.08351*, 2018. 8