Unsupervised Learning of Camera Pose with Compositional Re-estimation

Seyed Shahabeddin Nabavi
York University
nabaviss@yorku.ca

Mehrdad Hosseinzadeh
University of Manitoba
mehrdad@cs.umanitoba.ca

Ramin Fahimi
Ryerson University
ramin.fahimi@ryerson.ca

Yang Wang
University of Manitoba
ywang@cs.umanitoba.ca

Abstract

We consider the problem of unsupervised camera pose estimation. Given an input video sequence, our goal is to estimate the camera pose (i.e. the camera motion) between consecutive frames. Traditionally, this problem is tackled by placing strict constraints on the transformation vector or by incorporating optical flow through a complex pipeline. We propose an alternative approach that utilizes a compositional re-estimation process for camera pose estimation. Given an input, we first estimate a depth map. Our method then iteratively estimates the camera motion based on the estimated depth map. Our approach significantly improves the predicted camera motion both quantitatively and visually. Furthermore, the re-estimation resolves the problem of out-of-boundaries pixels in a novel and simple way. Another advantage of our approach is that it is adaptable to other camera pose estimation approaches. Experimental analysis on KITTI benchmark dataset demonstrates that our method outperforms existing state-of-the-art approaches in unsupervised camera ego-motion estimation.

1. Introduction

We tackle the problem of visual odometry (VO), where the goal is to estimate the camera poses (e.g. motion) given a number of consecutive frames in a video sequence. This problem plays an important role in many real-world applications, such as self-driving vehicles [2], obstacle avoidance [30], interactive robots [6] and navigation systems [7]. In the presence of a single RGB camera (i.e. monocular), this problem has been explored in [44, 42, 26, 43, 25, 3, 19, 19, 16, 40] from various perspectives and under different assumptions. Our work is particularly inspired by a recent line of work [44, 42, 26] on learning monocular camera pose estimation and depth estimation in an *unsupervised* setting. The only available data in this setting during training are monocular frames and camera intrinsics. The model is learned to map the input pixels to an estimate of camera poses (parameterized as transformation matrices) and scene structures (parameterized as depth maps). During testing, the input to the model is the raw video. We will use the learned model to produce the camera poses of the test video. As a by-product, we will also obtain the predicted depth map on each frame of the test video.

Figure 1: An illustration of the problem of large displacement between two views in pose estimation with the view synthesis formulation. The 3rd row shows three consecutive frames in a video. The 1st row shows the difference between the left and middle frames. The 2nd row shows the difference between the middle and right frames. When the displacement of two views is large, the assumption made by the view synthesis no longer holds. In this paper, we propose an alternative approach that splits the estimation into smaller pieces and re-estimate the transformation through a compositional transformation estimation.

Several previous works (e.g. [44, 42, 26]) have been proposed to estimate the relative camera pose between consecutive frames in a video sequence using a view synthesis for-

mulation. These methods work by predicting the camera poses and the depth maps, then using them to warp nearly frames to a target view using the predicted camera poses and depth maps. The learning objective is defined using the photometric loss between the predicted target view and the ground-truth target view. This view synthesis formulation implicitly makes several assumptions: 1) the scene is static; 2) there is no occlusion/disocclusion between two views; 3) there is no lighting change between two views. These assumptions often fail in applications where there exists a large displacement between the source view and the target view (see Fig. 1).

To address these limitations, we propose a new unsupervised camera pose estimation approach using compositional re-estimation. Our proposed approach is partly inspired by the inverse compositional spatial transformer network [21] being developed for image alignment. The idea of our approach is that instead of estimating the relative pose between two frames in one shot, we consider the relative pose as being composed of a sequence of smaller camera poses. These smaller camera poses are estimated in a recurrent manner. The advantage of this compositional re-estimation is that we can decompose the problem of estimating the camera pose with a large displacement into several smaller ones, where each smaller problem satisfies the assumption made by the view synthesis formulation of unsupervised camera pose estimation.

This paper makes several contributions. We propose a new compositional re-estimation approach that decomposes the camera pose estimation into a sequence of smaller pose estimation problems. Although the idea of compositional re-estimation has been used for image alignment [21], this is the first work using this idea for deep visual odometry. Our model can be trained end-to-end in an unsupervised learning setting. Experimental results show that our method significantly outperforms other state-of-the-art approaches.

2. Related Work

In this section, we review several lines of research closely related to our work.

Structure from Motion: Simultaneous estimation of structure and motion is a long-standing and fundamental problem in computer vision. Traditional approaches rely on geometric constraints extracted from monocular feed to estimate motion. They commonly start with feature extraction and matching, followed by geometric verification [32, 36, 33]. They are effective and powerful, yet computationally expensive and only focus on salient features. They also need high-quality images, and the results can drift over time due to factors such as low texture, stereo ambiguities, occlusions and complex geometry. Recently, learning-based methods have become popular and raised the bar on the performance [40, 15, 14, 27]. DeepVO [40] performs end-to-end visual odometry. PoseNet [15] learns 6 Degree-of-Freedom (6DOF) pose regression from monocular RGB images. Encoder-decoder style Hourglass networks have also been proposed to perform localization [27]. Tang et al. [35] present BA differentiabl layer to bridge the gap between classic and deep learning methods. They minimize the feature-metric difference of aligned pixels. On the other hand, our focus is on leveraging recurrent architecture in direct method.

Depth Estimation: Increasing availability of single view datasets [10, 29, 20] has made it possible to have significant improvement in depth prediction. Supervised deep networks [4, 22, 23, 8, 41, 18, 17, 1, 37] have achieved a promising performance and a variety of architectures have been proposed. Eigen et al. [4] demonstrate the capability of deep models for single view depth estimation by directly inferring the final depth map from the input image using two scale networks. Liu et al.[22, 23] formulate depth estimation as a continuous conditional random field learning problem. Laina et al. [18] propose the Huber loss and a newly designed up-sampling module. Kumar et al. [17] demonstrate that recurrent neural networks (RNNs) can learn spatiotemporally accurate monocular depth prediction from a video. Supervised techniques are limited due to the difficulty of collecting expensive ground truth information and impractical in applications as they often require data collection process different from the target robotic deployment platform.

Warping-based View Synthesis: Rethinking depth estimation as an image reconstruction task allows to alleviate the need for ground-truth labels. Self-supervised approaches for structure and motion borrow ideas from warping-based view synthesis. The core idea is to supervise depth estimation by treating view-synthesis via rigid structure from motion as a proxy task. Recently, unsupervised single image camera pose estimation and depth estimation techniques have shown remarkable progress [19, 12, 44, 38, 25, 5, 39]. These methods are mostly based on the photometric error which uses a Lambertian assumption. Garg et al. [9] train a network for monocular depth estimation using a reconstruction loss over a stereo pair with Taylor approximation to make the model fully differentiable. Godard et al.[12] further improve the results by introducing symmetric left-right consistency criterion and better stereo loss functions. Zhou et al. [44] propose a temporal reconstruction error that is computed using temporally aligned snippets of monocular images to deal with the limitation of having stereo images. The camera pose is unknown and needs to be estimated together with depth. The learning loss is obtained by combining a depth estimation network with a pose estimation network. This leads to the loss of absolute scale information in their predictions. This is solved by Li et al. [19] who combine both spatial and temporal reconstruction losses to

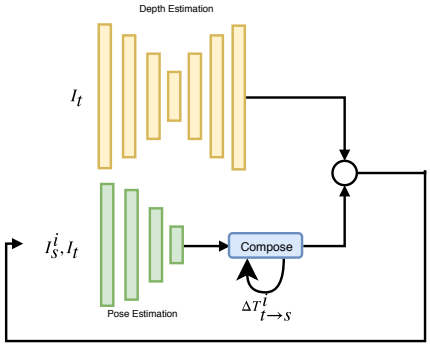

Figure 2: The re-estimation process consists of the pose estimation network, the depth estimation network and compositional variables which keep track of the transformations. The circle indicates the inverse warping process. The recursive arrow shows the warped sources passed to the pose net for the next step.

directly predict the scale-aware depth and pose from stereo images. Proposed by Mahjorian et al. [26], geometric constraints of the scene are enforced by an approximate ICP based loss. On the other hand, Yin et al. [42] jointly learns monocular depth, ego-motion and optical flow from video sequences. To handle occlusion and ambiguities, an adaptive geometric consistency loss is proposed to increase robustness towards outliers and non-Lambertian regions. Geometric features are extracted over the predictions of individual modules and then combined as an image reconstruction loss. Last but not least, Wang et al. [39] address scale ambiguity through a compositional unit which requires Jacobian calculation.

Compositional and Transformer Networks: Spatial transformer networks [13] are developed to resolve the ambiguity of spatial variations for classification. Jaderberg et al. [13] propose a novel strategy for integrating image warping in neural nets. Inverse compositional spatial transformers [21] further extends this work to remove the boundary artifacts introduced by STNs based on intuitions from the *Lucal & Kanade* algorithm [24] that propagates warp parameters rather than image intensities.

3. Our Approach

The basic components of our method are illustrated in Fig. 2. The input to our model consists of N consecutive frames in a video denoted as $< I_1, I_2, ..., I_N >$. We consider one frame I_t as the target frame (also known as target view or target image) and the remaining frames I_s ($1 \leq s \leq N, s \neq t$) as the source frames (also known as source views or source images). Our model consists of a depth network, a pose estimation network, and a warping module. The depth network produces a per-pixel depth map D_t of the target frame. The pose estimation network learns to iteratively produce camera relative pose $T_{t\to s}^i$ (parameterized as a 6 DoF vector representing the transformation) between the target frame I_t and source frames I_s where i is the index of the iteration. At each iteration, we also maintain a warped source image denoted as I_s^i. This warped source image is obtained by applying the transformation $T_{t\to s}^i$ on the source image I_s. In other words, the pose estimation network takes a target view I_t and N source views I_s^{i-1} at the i-th iteration as its input. It then produces $\Delta T_{t\to s}^i$. This transformation is combined with previous transformations $T_{t\to s}^{i-1}$ from earlier iterations to be used for warping I_s (original source frames) by incorporating the depth map D_t and camera intrinsics K (see Sec. 3.2). Let r be the number of iterations of this re-estimation process. The loss function is defined in the last step of the process where $i = r$. The entire process is explained as an algorithm in the supplementary material.

3.1. Compositional Re-estimation

The goal of the compositional re-estimation module is to estimate the transformation $T_{t\to s}^r \in SE(3)$ from the target frame to a set of source frames. Instead of estimating the transformation in one shot, we use an iterative process that estimates this transformation incrementally. In each iteration i, we estimate an incremental transformation $\Delta T_{t\to s}^i \in SE(3)$. We use $T_{t\to s}^i$ to denote the transformation after the i-th iteration. $T_{t\to s}^i$ can be obtained by adding the effect of $\Delta T_{t\to s}^i \in SE(3)$ to the transformation matrix $T_{t\to s}^{i-1}$ from the previous iteration, i.e.

$$T_{t\to s}^i = \Delta T_{t\to s}^i \oplus T_{t\to s}^{i-1} \tag{1}$$

where $T_{t\to s}^0$ includes rotation, translation. It is initialized by transformation zero and the rotation identity matrix and a row of 0 and 1 to make the matrix squared, here, $\oplus$ denotes a matrix multiplication operator. Let r be the number of this compositional re-estimation steps, $T_{t\to s}^r$ will be used as the final transformation.

The intuition behind this process is that by obtaining $T_{t\to s}^r$ from $\Delta T_{t\to s}^i$ ($i = 1, 2, ..., r$), we allow the model to solve the camera pose estimation problem by splitting it into simpler pieces. Since each step in this process only needs to estimate a small amount of transformation, the assumptions commonly made in camera pose estimation algorithms are more likely to hold. We can unfold this process of compositional re-estimation over time steps as depicted in Fig. 3.

3.2. Warping Module

In each estimation step i, a warped view I_s^i is generated by projecting each pixel p_t in the target view I_t to the corresponding position p_s in the source view (for each source

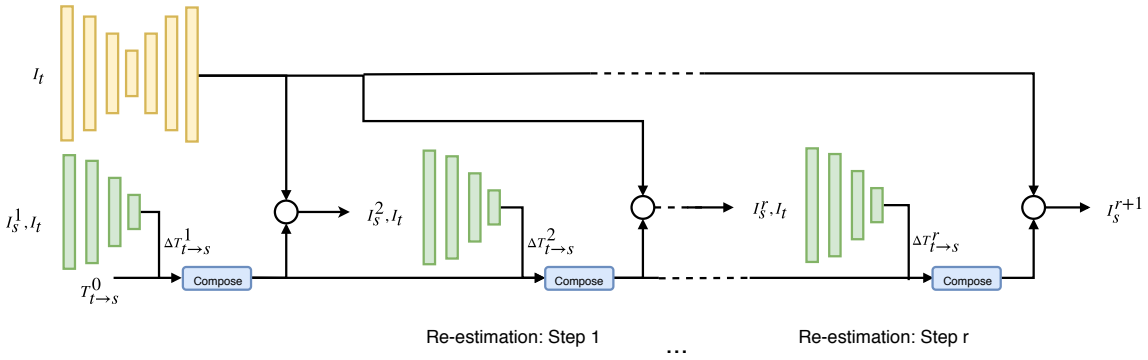

Figure 3: Our process is unfolded over time steps. The pose estimation network (green) estimates $\Delta T_{t\to s}^i$ in every steps by receiving I_s^{i-1} and I_t. $\Delta T_{t\to s}^i$ is then composed to create the final $T_{t\to s}^r$. The loss functions will be calculated only in the last step. Warped source views I_s^{r+1} from transformation $T_{t\to s}^r$ will be used for calculating the loss.

view in I_s^{i-1}) and inversely warp them. This process is done for each estimation step $i \in \{1, ..., r\}$. Since the process is the same throughout these time steps, we explain this warping module in one time step.

As shown in Fig. 4, each pixel $p_t \in I_t$ must be mapped to the corresponding $p_s \in I_s^{i-1}$. This process requires the camera intrinsics K, the estimated depth D_t and transformation $T_{t\to s}^i$ (see Eq. 2). Each $p_s \in I_s^{i-1}$ is warped to position $p_t \in I_t$ to produce I_s^i.

$$p_s \sim K T_{t\to s}^i D_t(p_t) K^{-1} p_t \tag{2}$$

In the above equation, K is a matrix of camera intrinsics and $D_t(p_t)$ is the corresponding depth of p_t and $T_{t\to s}^i \in$ SE(3).

Since some pixels are not mapped to regular grids, we reconstruct the value of p_t with respect to the projection by a weighted sum of pixel neighbourhood through bilinear interpolation (Eq. 3) similar to [44].

$$I_s^i(p_t) = \sum_{i\in t,b,j\in l,r} w^{i,j} I_s^i(p_s^{i,j}) \tag{3}$$

In this equation, t,b,l and r denote top,bottom,left and right.

3.3. Training Losses

Training the re-estimation process requires a supervision signal in the form of a loss function. This loss function consists of four main components.

Photometric Difference ($\mathcal{L}_{ph}$): This loss function plays a vital role in our framework. Like [44, 42, 26], $\mathcal{L}_{ph}$ is an $L1$ loss between the warped source views I_s^{r+1} and the target view:

$$\mathcal{L}_{ph} = \sum_{I\in I_s^{r+1}} \sum_p |I_t(p) - I(p)| \tag{4}$$

where p represent a pixel in an image.

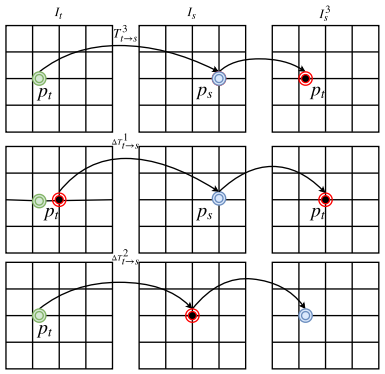

Figure 4: The impact of two steps re-estimation is illustrated. The 2nd and 3rd rows are decompositions of the 1st row. The 1st row shows how transformation $T_{t\to s}^2$ leads to warping $p_s \in I_s$ to $p_t \in I_t$. It consists of 2 steps of estimation. In the first step (2nd row), the pixel p_s is warped to p_t, but the transformation is not exactly correct. The next step (3rd row) corrects the mistake of the previous step by adding a complementary transformation to the previous step. As a result, $T_{t\to s}^2$ is obtained which is a true transformation from the target view to the source view. Note that although we estimate $T_{t\to s}^2$, we inversely warp source views to target view by the inverse of this transformation.

Multi Scale Dissimilarity: This term is known as DSSIM which was firstly used in [42]. It is resilient to outliers as well as being differentiable. It calculates the dissimilarity in multi-scales of the I_s^{r+1} and I_t. We incorporate this term with the photometric loss to form a rich dissimilarity loss. Therefore, we define it as follows:

$$\mathcal{L}_d = \sum_{i=1}^{n} \sum_{I\in I_s^{r+1}} \frac{1 - SSIM(I, I_t)}{2} \tag{5}$$

where n denotes the number of scales in the prediction.

Smoothness: This term keeps sharp details by encouraging disparities to be locally smooth. It mainly contributes to the quality of the disparity map. As most of the work on monocular depth estimation such as [42], we find this term very helpful in our method. We defined this term as $\mathcal{L}_s$.

Principled Mask: The term of principled mask refers to an attention mechanism which ensures that out of bound pixels do not contribute to the loss function. This term is used in [44, 26]. In our work, this mask only contributes to the last step (r) of estimation. In order to avoid the trivial attention of zero for all pixels, we also use a regularization term ($\mathcal{L}_{reg}(E)$) in [44] in our loss function on the mask. As a result, the final photometric term in our loss function is as follows:

$$\mathcal{L}_{ph} = \sum_{I \in I_s^{r+1}} \sum_{p} E(p) \left| I_t(p) - I(p) \right| \tag{6}$$

where E_s is pixel-wise predicted principled mask for the target and source and p denotes a pixel.

Putting all the pieces together, the final loss function for training our model is then computed as a weighted summation of aforementioned loss functions:

$$\mathcal{L}_{final} = \lambda_{ph}\mathcal{L}_{ph} + \lambda_d\mathcal{L}_d + \lambda_s\mathcal{L}_s + \lambda_e \sum_{i=1}^{n} \mathcal{L}_{reg}(E^i) \tag{7}$$

where λ_{ph}, λ_s, λ_d and λ_e are loss weights. Note that following [44], the final loss is computed over different scales.

Since our method estimates the relative pose in multiple steps in a recurrent manner, the vanishing gradient may become an issue. To overcome this, we use residual connections and memory mechanisms in our model shown in Fig. 3. The depth estimation network has residual connections to every differentiable warping module to alleviate the vanishing gradient problem. On the other hand, *compose* $\in SE(3)$ is a variable which preserves the compositional transformation for the warping module. This variable is updated at each step so that the warping module always has access to the most updated version of transformations.

3.4. Model Architecture

Pose Estimation Network: The pose estimation network is an encoder. Each layer is a convolution followed by a ReLU activation for non-linearity. The inputs to the encoder are I_t, I_s^i. The encoder outputs n 6DOF vectors corresponding to each source view to represent camera relative poses $\Delta T_{t\to s}^i$ from target view I_t to source views I_s^i.

In the last step of the re-estimation process, this network behaves differently, and it outputs $\Delta T_{t\to s}^r$ and an attention mask denoted as E^r. This attention mask is generated using a sequence of deconvolution (convTranspose) followed by sigmoid. This attention mask is used to exclude out of boundary pixels [26]. Note that it is acceptable that some pixels may not contribute to the loss function because they are not in target view. However, one step estimation excludes some pixels that are supposed to be in the target but are warped out of boundary due to the wrong estimation. Since we estimate the pose in multiple steps, the out of boundary pixels of ours and previous methods are different.

Depth Estimation Network: The depth estimation network outputs the disparity map of I_t. Pixel-level depth estimation provides a rich source of information to resolve scale ambiguity of camera motion estimation [43]. In order to be consistent with both [42] and [43], we report the results of using both VGG-based and ResNet50-based depth estimation networks.

4. Experiment

We evaluate the performance of the proposed method on two complementary tasks: camera pose estimation and depth estimation. Our experiments on these tasks demonstrate that the proposed formulation leads to state-of-the-art performance for estimating the camera pose while obtaining comparable results for estimating the target frame's depth.

In the following, we first describe the implementation details of training and give details of the benchmark dataset used in the experiments. Then we present both quantitative and qualitative results. We also investigate the impact of the re-estimation process on the performance by performing ablation studies.

4.1. Dataset and Training Details

Dataset: We evaluate our pose estimation network on the KITTI Odometry benchmark [11]. KITTI Odometry contains 22 sequences of frames recorded in street scenes from the egocentric view of the camera. Among the 22 sequences, IMU/GPS ground truth information of the first 11 sequences (seq. 00 to seq. 10) is publicly available. For the pose estimation task, we use the same training/validation splits used in [44, 42, 26, 43]. For pose estimation, we train the networks on seq. 00 to seq. 08 in the official odometry benchmark of KITTI dataset. Sequence 09 and sequence 10 are reserved for evaluating the performance of camera pose estimation. Besides, we provide qualitative outputs of our approach on sequences 11 and 15, though the ground truth is not available on these sequences. For depth estimation, we use 40k frames for training and 4k for validation in order to be consistent with previous work. We evaluate the depth estimation on the split provided by Eigen et al. [4]. It consists of 697 frames for which the depth ground truth is obtained by projecting the Velodyne laser scanned points into the image plane.

Training Details: The training procedure is performed in an end-to-end fashion by jointly learning camera pose and depth estimation at the same time. Monocular frames are resized to 128 $\times$ 416 and the network is optimized by an

improved variation of Adam optimizer [31]. The optimizer parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is adjusted at $2e^{-4}$ and loss weights are set to be $\lambda_{ph} = 0.15$, $\lambda_d = 0.85$, $\lambda_s = 0.1$ and $\lambda_e = 0.1$. In all of our experiments, we use a batch size of 4 and set the input sequence to be 3 frames for training.

Network Architecture: The pose estimation network consists of 7 convolution layers followed by ReLU. The last convolution is a 1×1 convolution to produce 6 DoF vectors. This 6 DoF vector corresponds to 3 Euler angles and 3-D translation which are then converted to SE(3) format for composition. In the last step of the re-estimation, the decoder of pose estimation is activated to produce the principled masks. In order to compare the depth estimation with previous work, we have experimented with using both VGG and ResNet50 as the backbone architecture in the depth estimation network. The VGG-based network is used in [44], while the ResNet50-based network is used in [42].

4.2. Monocular Pose Estimation

As discussed before, the input to the pose estimation network is a sequence of 3 consecutive frames. We follow [44] to split the long sequences into chunks of 3 frame. The middle frame in each chunk is considered as the target frame and the other two frames as source frames. Since our work is a monocular-based system, the frames are obtained from one camera in training and testing. In [26, 42, 44], the pose estimation network generates the camera pose vector in one step. In contrast, our approach uses the re-estimation process through composition. As a result, we achieve camera poses in a step-by-step fashion (see Sec. 3.1). The performance of pose estimation is measured by the absolute trajectory error (ATE) over 3 and 5 frames snippets. Table 1 compares the result of our method with other approaches. It is noteworthy that our method does not use any external supervision signal during training. Instead, it leverages a re-estimation process which leads to a better estimation of the camera pose. Also, note that our model even outperforms other baselines that use auxiliary information. For example, ORB-SLAM [28] benefits from loop closure techniques and GeoNet [42] utilizes the optical flow information in training. In contrast, our model does not use any of this auxiliary information. In order to evaluate the global consistency of the proposed method, we also evaluate ATE on the full trajectory which is described in [34] as another measurement. Table 2 shows the comparison with ORB-SLAM [28] without loop closure and SFMLearner [44].

4.3. Monocular Depth Estimation

We follow [44, 42] in setting up the training and testing sets for the depth estimation task. More specifically, we first filter out all the testing sequence frames and frames with a very small optical flow (with magnitude less than 1) from the training set. In the end, we obtain 44540 sequences. We use 40109 of them for training and the remaining 4431 for evaluation. Note that for the task of depth estimation, the input in the training and testing phases consists of only one frame (i.e. the target frame, I_t).

Similar to previous work, we multiple the predicted depth map by a scalar scale s defined as $s = \text{median}(D_{GT})/\text{median}(D_{predict})$ [44].

For a fair comparison, we compare with other monocular depth estimation approaches that use VGG and ResNet as the backbone architectures separately. Since the maximum depth in the KITTI dataset is 80 meters, we also limit the distance to 80 meters. The results are shown in Table 3.

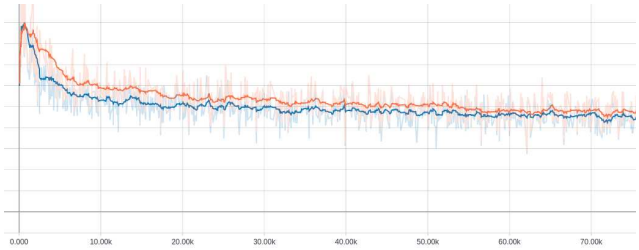

Figure 5: Dissimilarity loss (photometric loss + DSSIM loss) over training epochs. The loss of our approach (blue) is lower than that of the network without the re-estimation (orange) throughout the epochs. This shows that by using the re-estimation process, our model generates images that are more similar to the target frame.

Method	seq. 9	seq. 10
ORB-SLAM [28]	0.014 ± 0.008	0.012 ± 0.011
SFMLearner [44]	0.016 ± 0.009	0.013 ± 0.009
GeoNet [42]	0.012 ± 0.007	0.012 ± 0.009
3D ICP (3 frames)[26]	0.013 ± 0.010	0.012 ± 0.011
EPC++(mono) [25]	0.013 ± 0.007	0.012 ± 0.008
Ours (2 steps)	$\mathbf{0.009 \pm 0.005}$	$\mathbf{0.009 \pm 0.007}$

Table 1: Quantitative results for the camera pose estimation task. We compare our model with existing state-of-the-art approaches. Following prior work, we report the mean and standard deviation for Absolute Trajectory Error (ATE) over 3 and 5 snippets of sequence 9 and sequence 10 of KITTI odometry benchmark.

Method	seq. 09	seq. 10
ORB-SLAM[28]	54.94	26.99
SFMLearner [44]	31.21	28.36
Ours (2 steps)	**28.38**	**10.25**

Table 2: Odometry evaluation on KITTI odometry benchmark sequence 09 and sequence 10. The error refers to the translational ATE error over full trajectories.

Figure 6: Qualitative examples of depth estimation for one step (middle) and two steps (right) depth estimation through depth estimation network. Note that the only difference between them is the compositional re-estimation.

Although the results are comparable on the depth estimation task, our model does not outperform state-of-the-art on monocular depth estimation. This is expected since the re-estimation does not directly affect the depth estimation because it does not re-estimate the predicted depth map. This also confirms that the improvement of our method on camera pose estimation (see Table 1 and Table 2) is due to the compositional re-estimation.

4.4. Ablation Study

In order to further investigate the relative contribution of each module in our model, we perform two additional ablation studies. In the first experiment, we remove the re-estimation process in our model and train the rest of the network. We then measure the performance on the evaluation set. To do so, we set the maximum step (r) to 1 to assess the relative contribution of one step re-estimation process. Table 4 (2nd row) shows that removing this process profoundly impacts the overall performance. The estimation accuracy drops on seq. 09 is particularly significant. This might be due to the fact that seq. 9 is more complex than seq. 10 and requires more refinement for estimating the camera pose. In the second experiment, we investigate the impact of larger displacement on the optimal number of steps. Therefore, the number of input frame is also set to be five. As it is shown in table 5 and 6, the best performance on three frame snippets input is acquired by two steps estimation. However, since the displacement between source frames and target frame is larger in five frame snippets scenario, the best performance is acheived by three steps estimation.

Another important aspect of our method is that it leads to better image reconstruction. In Fig. 5, we visualize the re-construction loss (photometric and DSSIM) over training epochs to show how our method is better at re-construction than the baseline after a few epochs. We can see a noticeable gap between the loss of our model and the model without the re-estimation process.

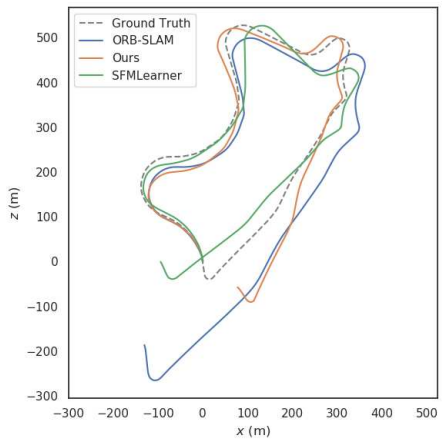

Figure 7: Full trajectories of our method (solid orange), SFMLearner [44] (solid blue), ORB-SLAM [28] (solid green) on the sequence 9 of KITTI Visual Odometry benchmark. Ground truth is shown in the dotted gray line.

4.5. Qualitative Experiment

We provide qualitative examples for camera ego-motion estimation as the main contribution of this paper. We visualize the full trajectories on sequence 9 and 10 (Fig. 7 and 8, respectively). Compared with [44], our trajectories are visually better and closer to ground truth. To further demonstrate the impact of the re-estimation process, we also show the performance of our method on official test sequences

Method	Supervised	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Cap 80m								
Eigen et al. [4] Coarse	Depth	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. [4] Fine	Depth	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [23]	Depth	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Godard et al. [12]	Pose	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Zhou et al. [44]	No	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou et al. [44] updated	No	0.183	1.595	6.709	0.270	0.734	0.902	0.959
GeoNet [42]	No	0.164	1.303	6.090	0.247	0.765	0.919	0.968
ICP [26]	No	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Ours VGG (2 steps)	No	0.170	1.384	6.247	0.255	0.758	0.913	0.962
Godard et al. [12]	Pose	0.124	1.076	5.311	0.219	0.847	0.942	0.973
GeoNet [42]	No	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Ours ResNet (2 steps)	No	0.160	1.195	5.916	0.245	0.774	0.917	0.964

Table 3: Quantitative results on the depth estimation task. We compare our model with other state-of-the-art monocular depth estimation approaches. Depth estimation is trained on the KITTI dataset. Evaluation is performed using the training/test split in [4]. "Depth" and "Pose" indicate using the ground truth depth and pose as supervision during training.

Method	seq. 9	seq. 10
ours (2 steps)	**0.009 ± 0.005**	**0.009 ± 0.007**
w/o re-estimation	0.011 ± 0.006	0.009 ± 0.007

Table 4: Results of ablation study of the proposed method on the pose estimation task. The 1st row shows the result of the network using the re-estimation process for 2 steps. The 2nd row shows the performance when removing it.

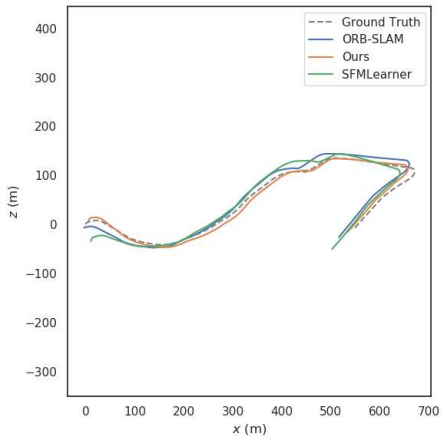

Figure 8: Full trajectories of our method(solid orange), SFMLearner [44] (solid blue), ORB-SLAM [28] (solid green) on the sequence 10 of KITTI Visual Odometry benchmark. Ground truth is shown in the dotted gray line.

(seq. 11 and seq. 15) of KITTI in supplementary material. In addition to this, we demonstrate the impact of the re-estimation process on depth estimation network in Fig.

	1 step	2 steps	3 steps
3 frames	0.011 ± 0.006	0.009 ± 0.005	0.009 ± 0.006
5 frames	0.015 ± 0.007	0.014 ± 0.007	0.013 ±0.007

Table 5: The role of the re-estimation process for 3 frame snippets and 5 frame snippets inputs on sequence 9 of KITTI odometry benchmark.

	1 step	2 steps	3 steps
3 frames	0.009 ±0.007	0.009 ± 0.007	0.009 ±0.009
5 frames	0.014 ± 0.008	0.013 ± 0.008	0.013 ± 0.007

Table 6: The role of the re-estimation process for 3 frame snippets and 5 frame snippets inputs on sequence 10 of KITTI odometry benchmark.

6.

5. Conclusion and Future work

In this paper, we have proposed a novel technique for learning to estimate camera ego motion step by step in an unsupervised deep visual odometry framework. Instead of estimating the camera pose in one pass, our method estimates the camera pose in an iterative fashion. Our method provides a new approach to address the problem of large displacement in consecutive frames. Experimental results on benchmark dataset show that our proposed method outperforms existing state-of-the-art approaches in camera pose estimation.

6. Acknowledgments

This work was supported by a grant from NSERC. We thank NVIDIA for donating some of the GPUs used in this work.

References

[1] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of IEEE International Conference on Computer Vision ICCV*, pages 2722–2730, 2015.

[3] G. Costante and T. A. Ciarfuglia. Ls-vo: Learning dense optical subspace for robust visual odometry estimation. *IEEE Robotics and Automation Letters*, 3:1735–1742, 2018.

[4] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of Advances in neural information processing systems (NeurIPS)*, 2014.

[5] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5515–5524, 2016.

[6] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3-4):143–166, 2003.

[7] F. Fraundorfer, C. Engels, and D. Nistér. Topological mapping, localization and navigation using image collections. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3872–3877. IEEE, 2007.

[8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2002–2011, 2018.

[9] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231–1237, 2013.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR*, pages 3354–3361, 2012.

[12] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, 2017.

[13] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Proceedings of Advances in neural information processing systems (NeurIPS)*, pages 2017–2025, 2015.

[14] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016.

[15] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

[16] K. R. Konda and R. Memisevic. Learning visual odometry with a convolutional network. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 1, pages 486–490, 2015.

[17] A. C. Kumar, S. M. Bhandarkar, and P. Mukta. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR workshop)*, pages 396–3968, 2018.

[18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vision (3DV)*, pages 239–248, 2016.

[19] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291, 2018.

[20] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050, 2018.

[21] C.-H. Lin and S. Lucey. Inverse compositional spatial transformer networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2252–2260, 2017.

[22] D. Liu, X. Liu, and Y. Wu. Depth reconstruction from single images using a convolutional neural network and a condition random field model. *Sensors*, 2018.

[23] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38:2024–2039, 2016.

[24] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI)*, 1981.

[25] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018.

[26] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5667–5675, 2018.

[27] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 879–886, 2017.

[28] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31:1147–1163, 2015.

[29] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images.

In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012.

[30] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

[31] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

[32] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Proceedings of Advances in neural information processing systems (NeurIPS)*, 2005.

[33] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision (IJCV)*, 76:53–69, 2008.

[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580. IEEE, 2012.

[35] C. Tang and P. Tan. Ba-net: Dense bundle adjustment network. *International Conference on Learning Representations (ICLR)*, 2019.

[36] A. Torralba and A. Oliva. Depth estimation from image structure. *TPAMI*, 24:1226–1238, 2002.

[37] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5622–5631, 2017.

[38] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[39] 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*, pages 2022–2030, 2018.

[40] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050, 2017.

[41] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *Proceedings of Advances in neural information processing systems (NeurIPS)*, 2017.

[42] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1983–1992, 2018.

[43] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*, 2018.

[44] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017.