This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version;





Pixel-Wise Prediction based Visual Odometry via Uncertainty Estimation

Hao-Wei Chen, Ting-Hsuan Liao, Hsuan-Kung Yang, and Chun-Yi Lee Elsa Lab, Department of Computer Science National Tsing Hua University, Hsinchu City, Taiwan

{jaroslaw1007, tingforun, hellochick, cylee}@gapp.nthu.edu.tw

Abstract

This paper introduces pixel-wise prediction based visual odometry (PWVO), which is a dense prediction task that evaluates the values of translation and rotation for every pixel in its input observations. PWVO employs uncertainty estimation to identify the noisy regions in the input observations, and adopts a selection mechanism to integrate pixelwise predictions based on the estimated uncertainty maps to derive the final translation and rotation. In order to train PWVO in a comprehensive fashion, we further develop a data generation workflow for generating synthetic training data. The experimental results show that PWVO is able to deliver favorable results. In addition, our analyses validate the effectiveness of the designs adopted in PWVO, and demonstrate that the uncertainty maps estimated by PWVO is capable of capturing the noises in its input observations.

Keywords— Visual odometry, uncertainty estimation, pixel-wised predictions.

1. Introduction

Visual odometry (VO) is the process of inferring the correspondence of pixels or features via analyzing the associated camera images, and determining the position and orientation of a robot. Conventionally, the process of VO takes raw RGB images as inputs, derives the correspondence from them, and estimates the changes in translation and rotation of the camera viewpoints between consecutive image frames. This process enables a robot to derive its entire trajectory from its observed images over a period of time. In the past decade, there have been a number of VO methods proposed in the literature [1-22, 22-29]. The requirement of estimating such changes from raw images, however, often causes those methods to suffer from the existence of noises coming from moving objects, as the correspondence extracted from those moving objects might not be directly related to the motion of the camera viewpoint. This leads to degradation in accuracy when performing VO,

and limits previous methods to further improve.

In order to deal with the above issue, the objectives of this paper are twofold: (1) validating the fact that the noises from moving objects would degrade the performance of VO, and (2) investigating and proposing an effective approach to mitigate the impacts from them. Instead of embracing VO frameworks that are based on raw RGB inputs [1–22, 26–29], in this work, we focus the scope of our discussion on deriving relative translation and rotation from intermediate representations [23–25, 30, 31], which include extracted optical flow maps, depth maps, as well as known camera intrinsic. Specifically, we aim to derive and validate our method based on clear intermediate representations, decoupling the influences of inaccurate features and correspondence from from the feature extraction processes.

The first step toward the aforementioned objectives is to develop a mechanism that allows the noises of moving objects from those intermediate representations to be either filtered out or suppressed. Past researchers have explored three different directions: (i) semantic masks, (ii) attention mechanisms, and (iii) uncertainty estimation. Among them, semantic masks require another separate segmentation model to discover potential moving objects (e.g., cars, pedestrians, etc.) [32-35]. Attention mechanisms seek the clues of possible candidates from intermediate representations [21, 26–29, 32, 36–39]. Uncertainty estimation, on the other hand, implicitly captures the noises and enables the models to respond to the measured stochasticity inherent in the observations [11, 22, 40-45]. All of these directions have been attempted in the realm of VO. Nevertheless, the previous endeavors have only been concentrating on predicting a single set of translation and rotation values from their input observations, neglecting the rich information concealed in pixels. In light of these reasons, we propose to employ uncertainty estimation as a means to leverage such information, and further extend VO to be based on pixel-wise predictions. This concept can be regarded as an implicit form of the attention mechanism.

Different from conventional VO approaches, pixel-wise

prediction based VO (or simply "*PWVO*" hereafter) is designed as a dense prediction task, which evaluates the values of translation and rotation for every pixel in its input observations. PWVO first performs predictions for all pixels in the input observations, and then integrates these local predictions into a global one. As PWVO is based on pixelwise predictions, such a nature allows it to suppress noisy regions through the usage of uncertainty maps. The regions with high uncertainty are likely to be noises (e.g., moving objects), and should not be considered in the final global prediction. As a result, a weighting and selection strategy is specifically tailored for PWVO to aggregate the local predictions from its input pixels.

In order to validate the advantages of PWVO, we further develop a data generation workflow, which features a high degree of freedom to generate intermediate representations for PWVO. The workflow is fully configurable, and allows various setups of camera intrinsic, viewpoint motion, extrinsic range, as well as a diverse range of the number, size, and speed for moving objects. Such a flexibility enables the training data to be comprehensive, and prevents PWVO from overfitting to the setups of a certain existing dataset. In our experiments, we examine the effectiveness of PWVO in terms of its accuracy, saliency maps, as well as factorized optical maps. We further present a set of ablation analyses to justify the design decisions adopted by PWVO. The main contribution of this paper is the introduction of pixel-wise predictions in VO, and the dataset generation workflow.

The paper is organized as follows. Section 2 reviews the related work. in the literature. Section 3 walks through the PWVO framework and its components. Section 4 describes the dataset generation workflow. Section 5 reports the experimental results. Section 6 discusses the limitations and future directions. Section 7 concludes the paper. Please note that the essential background material, hyperparameter setups, additional results, and our reproducible source codes are offered in the supplementary material.

2. Related Work

Traditional VO methods are based on multiview geometry. According to the algorithms adopted by them, these methods can be roughly categorized into either featurebased [46–48] or direct [49–51] methods. The former involves correspondences between image frames by using sparse key points, while the latter attempts to recover camera poses by minimizing the image warping photometric errors. Both categories suffer from the scale drift issue if the absolute depth scale is unknown. The authors in [2, 3] pioneered the usage of convolutional neural networks (CNNs) to learn camera pose estimation with various types of loss functions and model architectures. The authors in [5, 10, 15, 16] proposed to directly regress six degrees of freedom (6-DoF) camera poses through weighted combinations of position and orientation errors. Methods employing geometric re-projection errors [13] and visual odometry constraints [10, 18, 19] were also introduced in the literature. Moreover, the authors in [17, 20] attempted to train their models with extra synthetic data, while the authors in [4] trained their VO models from videos using recurrent neural networks. Furthermore, the technique proposed in [5] adopted an end-to-end approach that merges camera inputs with inertial measurement unit (IMU) readings using an LSTM network. Most of these image-based localization methods perform VO via retrieving information from input images. As a result, feature extraction is critical to their performance, which in turn impacts their generalizability to challenging scenarios. To reduce the impacts from feature extraction stages, some researchers [23-25] proposed to use optical flow maps as inputs instead of RGB images. The authors in [24] further proposed to utilize autoencoder networks to learn a better representations for their optical flow maps. To eliminate the noises from input observations, a number of researchers have investigated attention based VO methods [21, 26-28, 32, 36-38] and uncertainty based VO methods [22, 41, 44, 45] as well to mitigate the impacts of moving objects from their input observations.

3. Methodology

In this section, we first formally define our problem formation, and provide an overview of the PWVO framework. Next, we describe the two constituent stages in PWVO. Finally, we introduce the refinement strategy as well as the total loss term, and elaborate on the rationale behind them.

3.1. Problem Formation

Given an optical flow field $\mathbf{F}_{i}^{total} \in \mathbb{R}^{H \times W \times 2}$, depth maps $(\mathbf{D}_{i}, \mathbf{D}_{i+1}) \in \mathbb{R}^{H \times W \times 1}$, two dimensional pixel coordinates $\mathbf{x} \in \mathbb{R}^{H \times W \times 2}$, and the camera intrinsic $\mathbf{K}_{i} \in \mathbb{R}^{3 \times 3}$, the proposed PWVO framework aims at predicting a tuple of camera rotation $\tilde{\gamma}_{i} \in \mathbb{R}^{3}$ and translation $\tilde{\varphi}_{i} \in \mathbb{R}^{3}$, where *i* denotes the frame index, and *H*, *W* represent the height and width of the input frames, respectively.

In order to achieve the above objective, PWVO first performs pixel-wise predictions of the camera motion $(\tilde{\mathcal{R}}_i, \tilde{\mathcal{T}}_i)$, where $\tilde{\mathcal{R}}_i \in \mathbb{R}^{H \times W \times 3}$ and $\tilde{\mathcal{T}}_i \in \mathbb{R}^{H \times W \times 3}$ correspond to the pixel-wise rotation and translation maps, respectively. In addition, PWVO further generates an uncertainty map tuple $(\tilde{\mathcal{U}}_i^{\mathcal{R}}, \tilde{\mathcal{U}}_i^{\mathcal{T}})$, where $\tilde{\mathcal{U}}_i^{\mathcal{R}} \in \mathbb{R}^{H \times W \times 3}$ and $\tilde{\mathcal{U}}_i^{\mathcal{T}} \in \mathbb{R}^{H \times W \times 3}$, to reflect the uncertainty (i.e., noises) in the input observations. The predicted $(\tilde{\mathcal{R}}_i, \tilde{\mathcal{T}}_i)$ are then used along with $(\tilde{\mathcal{U}}_i^{\mathcal{R}}, \tilde{\mathcal{U}}_i^{\mathcal{T}})$ to derive the final $(\tilde{\gamma}_i, \tilde{\varphi}_i)$.

3.2. Overview of the PWVO Framework

Fig. 1 illustrates the proposed PWVO framework, which consists of two stages: (i) an encoding stage and (ii) a pixel-wise prediction stage. The function of the encoding stage is



Figure 1: An overview of the proposed PWVO framework.

to encode the inputs (i.e., \mathbf{F}_{i}^{total} , \mathbf{D}_{i} , \mathbf{D}_{i+1} , and \mathbf{x}) by a series of convolutional operations into a feature embedding ψ , which bears the motion information concealed in the inputs. This embedding is then forwarded to the pixel-wise prediction stage to generate $(\tilde{\mathcal{R}}_i, \tilde{\mathcal{T}}_i)$ and $\tilde{\mathcal{U}}_i$ by two separated branches. The uncertainty map \mathcal{U}_i is utilized to reflect the noises contained in the inputs. On the other hand, the predicted $(\mathcal{R}_i, \mathcal{T}_i)$ and \mathcal{U}_i are later fed into a selection module, which employs a patch selection procedure to mitigate the impacts of the regions that may potentially contain moving objects by referring to \mathcal{U}_i , and aggregates the weighted predictions to derive the final $(\tilde{\gamma}_i, \tilde{\varphi}_i)$. In order to further refine the predicted $(\tilde{\gamma}_i, \tilde{\varphi}_i)$, PWVO additionally re-constructs an ego flow prediction $\tilde{\mathbf{F}}_i^{ego} \in \mathbb{R}^{H \times W \times 2}$ as well as a depth map $\tilde{\mathbf{D}}_{i+1} \in \mathbb{R}^{H \times W \times 1}$ based on $(\tilde{\gamma}_i, \tilde{\varphi}_i)$. The reconstructed $\tilde{\mathbf{F}}_i^{ego}$, $\tilde{\mathbf{D}}_{i+1}$, and $(\tilde{\gamma}_i, \tilde{\varphi}_i)$ are then compared against their corresponding ground truth labels \mathbf{F}_{i}^{ego} , \mathbf{D}_{i+1} , and (γ_i, φ_i) respectively to optimize the model parameters θ in PWVO through backward propagation.

3.3. Encoding Stage

The encoding stage first forwards \mathbf{F}_i^{total} , \mathbf{D}_i , \mathbf{D}_{i+1} , and \mathbf{x} through three distinct branches, which are then followed by an encoder to transform the concatenated feature embeddings from the outputs of the three branches into $\tilde{\psi}$. In addition to the flow and depth information, the last branch of the encoding stage is designed to encapsulate the positional clues of \mathbf{x} . This encapsulation process allows PWVO to learn translation dependence [52], and is expressed as:

$$\mathbf{X}_{i}^{p} = \mathbf{K}_{i}^{-1} \mathbf{x}_{i}^{p}, \mathbf{K}_{i} \in \mathbb{R}^{3 \times 3}, \mathbf{x}_{i}^{p} \in \mathbb{R}^{3}, \mathbf{N} \in \mathbb{R}^{(H \times W)},$$
(1)

where p denotes the pixel index, \mathbf{X}_i^p represents the three dimensional coordinates in the film space, and \mathbf{N} is the pixel number $H \times W$. Eq. (1) reveals that the third branch of the encoding stage allows the information of the camera intrinsic and \mathbf{x} to be carried over to $\tilde{\psi}$. This can potentially benefit PWVO to possess better understanding about positioning and translation dependence, as the motion features provided by \mathbf{F}_i^{total} and the coordinate information offered by x can complement each other.

3.4. Pixel-Wise Prediction Stage

The pixel-wise prediction stage first utilizes two decoders Decoder-rt and Decoder-un, as depicted in Fig. 1, to upsample $\tilde{\psi}$ to $(\tilde{\mathcal{R}}_i, \tilde{\mathcal{T}}_i)$ and $(\tilde{\mathcal{U}}_i^{\mathcal{R}}, \tilde{\mathcal{U}}_i^{\mathcal{T}})$, respectively. In the following subsections, we elaborate on the distribution learning procedure and the selection module.

3.4.1 Distribution Learning

The distribution learning procedure in PWVO aims at learning the posterior probability distributions of rotation and translation at each pixel. Assume that the noises are modeled as Laplacian, this procedure can be carried out by leveraging the concept of heteroscedastic aleatoric uncertainty of deep neural networks (DNNs) discussed in [40]. More specifically, $(\tilde{\mathcal{R}}_i, \tilde{\mathcal{T}}_i)$ and $(\tilde{\mathcal{U}}_i^{\mathcal{R}}, \tilde{\mathcal{U}}_i^{\mathcal{T}})$ are learned together by minimizing the loss terms \mathcal{L}_i^R and \mathcal{L}_i^T for all pixels, which can be expressed as the following:

$$\hat{\mathcal{L}}_{i}^{\mathcal{R}} = \frac{1}{\mathbf{N}} \sum_{p=1}^{\mathbf{N}} \frac{\mathcal{E}^{\mathcal{R}}(\gamma_{i}, \tilde{\mathcal{R}}_{i}^{p})}{\tilde{\mathcal{U}}_{i}^{\mathcal{R}}(p)} + \log(\tilde{\mathcal{U}}_{i}^{\mathcal{R}}(p)), \qquad (2)$$

$$\hat{\mathcal{L}}_{i}^{\mathcal{T}} = \frac{1}{\mathbf{N}} \sum_{p=1}^{\mathbf{N}} \frac{\mathcal{E}^{\mathcal{T}}(\varphi_{i}, \tilde{\mathcal{T}}_{i}^{p})}{\tilde{\mathcal{U}}_{i}^{\mathcal{T}}(p)} + \log(\tilde{\mathcal{U}}_{i}^{\mathcal{T}}(p)), \qquad (3)$$

where $\mathcal{E}^{\mathcal{R}}(x,y) = ||x-y||$ and $\mathcal{E}^{\mathcal{T}}(x,y) = ||\langle x \rangle - \langle y \rangle || + (||x||_2 - ||y||_2)^2$ serve as loss function for rotation and translation, respectively, $\langle \cdot \rangle$ denotes the Euclidean normalization vector, and $(\tilde{\mathcal{R}}_i^p, \tilde{\mathcal{T}}_i^p)$ and $(\tilde{\mathcal{U}}_i^{\mathcal{R}}(p), \tilde{\mathcal{U}}_i^{\mathcal{T}}(p))$ represent the means and variances of the probability distributions of rotation and translation at pixel p, respectively. Please note that $(\tilde{\mathcal{U}}_i^{\mathcal{R}}(p), \tilde{\mathcal{U}}_i^{\mathcal{T}}(p))$ are learned implicitly, and the second terms in Eqs. (2) and (3) regulate the scales of them. The loss functions allow PWVO to adapt its uncertainty estimation for different pixels, which in turn enhance its robustness to noisy data or erroneous annotations.

In practice, Decoder-un is modified to predict log variance, and Eqs. (2)-(3) are reformulated as follows:

$$\hat{\mathcal{L}}_{i}^{\mathcal{R}} = \frac{1}{\mathbf{N}} \sum_{p=1}^{\mathbf{N}} \exp\left(-\tilde{s}_{i}^{\mathcal{R}}(p)\right) \cdot \mathcal{E}^{\mathcal{R}}(\gamma_{i}, \tilde{\mathcal{R}}_{i}^{p}) + \tilde{s}_{i}^{\mathcal{R}}(p), \quad (4)$$

$$\hat{\mathcal{L}}_{i}^{\mathcal{T}} = \frac{1}{\mathbf{N}} \sum_{p=1}^{\mathbf{N}} \exp\left(-\tilde{s}_{i}^{\mathcal{T}}(p)\right) \cdot \mathcal{E}^{\mathcal{T}}(\varphi_{i}, \tilde{\mathcal{T}}_{i}^{p}) + \tilde{s}_{i}^{\mathcal{T}}(p).$$
(5)

where $\tilde{s}_i^{\mathcal{R}}(p) = \log(\tilde{\mathcal{U}}_i^{\mathcal{R}}(p)), \tilde{s}_i^{\mathcal{T}}(p) = \log(\tilde{\mathcal{U}}_i^{\mathcal{T}}(p))$. This modification allows the training progress of PWVO to be stabler than the original formulation, as it avoids errors resulted from division by zero. Moreover, the exponential mapping enables PWVO to regress unconstrained $\tilde{s}_i^{\mathcal{R}}(p)$ and $\tilde{s}_i^{\mathcal{T}}(p)$, as $\exp(\cdot)$ guarantees the outputs to be positive.

3.4.2 Selection Module

Fig. 2 illustrates the mechanism behind the selection module, which aims to derive $(\tilde{\gamma}_i, \tilde{\varphi}_i)$ from $(\tilde{\mathcal{R}}_i, \tilde{\mathcal{T}}_i)$ and $(\tilde{\mathcal{U}}_i^{\mathcal{R}}(p), \tilde{\mathcal{U}}_i^{\mathcal{T}}(p))$. It adopts a hierarchical derivation procedure, in which $H \times W$ pixels of a frame are first grouped into $h \times w$ patches of size $k \times k$ pixels, where h = H/k, w =W/k. The rotation, translation, and uncertainty maps for each patch can then be directly extracted from the original ones, and are represented as $(\tilde{\mathfrak{r}}_{l,m}, \tilde{\mathfrak{t}}_{l,m})$ and $(\tilde{\mathfrak{u}}_{l,m}^{\mathcal{R}}, \tilde{\mathfrak{u}}_{l,m}^{\mathcal{T}})$, where l and m denote the row and the column indices of a certain patch. The selection module next selects the pixel with the lowest uncertainty value within each patch, represented as: $p_{l,m}^{\mathcal{R}} = argmin(\tilde{\mathfrak{u}}_{l,m}^{\mathcal{R}}), p_{l,m}^{\mathcal{T}} = argmin(\tilde{\mathfrak{u}}_{l,m}^{\mathcal{T}}),$ where $p_{l,m}^{\mathcal{R}}$ and $p_{l,m}^{\mathcal{T}}$ are the pixel indices corresponding to patch (l, m). They are used to derive the final $(\tilde{\gamma}_i, \tilde{\varphi}_i)$ as:

$$\tilde{\gamma}_{i} = \sum_{l=1}^{h} \sum_{m=1}^{w} \mathcal{W}_{l,m}^{\mathcal{R}} \cdot \tilde{\mathfrak{r}}_{l,m}(p_{l,m}^{\mathcal{R}}),$$

$$\mathcal{W}_{l,m}^{\mathcal{R}} = \frac{\exp(\tilde{\mathfrak{u}}_{l,m}^{\mathcal{R}}(p_{l,m}^{\mathcal{R}}))}{\sum_{l=1}^{h} \sum_{m=1}^{w} \exp(\tilde{\mathfrak{u}}_{l,m}^{\mathcal{R}}(p_{l,m}^{\mathcal{R}}))}.$$

$$\tilde{\varphi}_{i} = \sum_{l=1}^{h} \sum_{m=1}^{w} \mathcal{W}_{l,m}^{\mathcal{T}} \cdot \tilde{\mathfrak{t}}_{l,m}(p_{l,m}^{\mathcal{T}}),$$

$$\mathcal{W}_{l,m}^{\mathcal{T}} = \frac{\exp(\tilde{\mathfrak{u}}_{l,m}^{\mathcal{T}}(p_{l,m}^{\mathcal{T}}))}{\sum_{l=1}^{h} \sum_{m=1}^{w} \exp(\tilde{\mathfrak{u}}_{l,m}^{\mathcal{T}}(p_{l,m}^{\mathcal{T}}))}.$$

$$(6)$$

$$(7)$$

The advantage of the hierarchical procedure is that it enforces $(\tilde{\gamma}_i, \tilde{\varphi}_i)$ to be derived from all the patches from the entire image instead of focusing on a certain local region.

3.5. Refinement and Total Loss Adopted by PWVO

In order to further refine the predicted $(\tilde{\gamma}_i, \tilde{\varphi}_i)$, PWVO additionally reconstructs $\tilde{\mathbf{F}}_i^{ego}$ and $\tilde{\mathbf{D}}_{i+1}$ based on $(\tilde{\gamma}_i, \tilde{\varphi}_i)$, and compare them against their ground truth labels \mathbf{F}_i^{ego}



Figure 2: An illustration of the mechanism of the selection module.

and D_{i+1} to optimize the model parameters in both stages of PWVO. The total loss of PWVO can be formulated as:

$$\mathcal{L}_i^{Total} = \hat{\mathcal{L}}_i^{\mathcal{R}} + \hat{\mathcal{L}}_i^{\mathcal{T}} + \hat{\mathcal{L}}_i^{\mathcal{D}} + \hat{\mathcal{L}}_i^{\mathcal{F}}, \tag{8}$$

where $\hat{\mathcal{L}}_i^{\mathcal{D}}$ and $\hat{\mathcal{L}}_i^{\mathcal{F}}$ represent the loss functions for ego flow and depth reconstruction, respectively. Please note that the detailed formulation of the two loss functions $\hat{\mathcal{L}}_i^{\mathcal{D}}$ and $\hat{\mathcal{L}}_i^{\mathcal{F}}$ are provided in the supplementary material.

The rationale behind the additional two loss terms (i.e., $\hat{\mathcal{L}}_i^{\mathcal{D}}$ and $\hat{\mathcal{L}}_i^{\mathcal{F}}$) in Eq. (8) can be explained from two different perspectives. First, re-projection from 2D coordinates to 3D coordinates might potentially cause ambiguity issues if depth information is not taken into consideration. Second, due to the pixel-wise design of PWVO, the optimization target should be different for each pixel coordinate, as the position and depth is different for each pixel. As a result, only optimizing $(\tilde{\mathcal{R}}_i, \tilde{\mathcal{T}}_i)$ without considering the depth and the positional information could be insufficient. These are the reasons that $\hat{\mathcal{L}}_i^{\mathcal{D}}$ and $\hat{\mathcal{L}}_i^{\mathcal{F}}$ are included in the final optimization target of PWVO. Section 5.2.3 offers an ablation analysis to validate the effectiveness of these loss terms.

Interestingly, part of the design of the optimization target $\hat{\mathcal{L}}_i^{\mathcal{F}}$ is similar to that of the well-known perspective-n-point (PnP) [53] approach, and can be viewed as a variant of it. This is because the re-projection error \mathbf{E}_i in $\hat{\mathcal{L}}_i^{\mathcal{F}}$ can be formulated as the L2 loss between $\tilde{\mathbf{F}}_i^{ego}$ and \mathbf{F}_i^{ego} , given by:

$$\mathbf{E}_{i} = \sum_{p=1}^{\mathbf{N}} \|\tilde{\mathbf{F}}_{i}^{ego}(p) - \mathbf{F}_{i}^{ego}(p)\|_{2}$$
$$= \sum_{p=1}^{\mathbf{N}} \|(\frac{1}{\mathbf{D}_{i+1}}\mathbf{K}_{i}\mathbf{M}_{i}\mathbf{X}_{i}^{p} - \mathbf{x}^{p}) - (\mathbf{F}_{i}^{ego}(p))\|_{2}, \quad (9)$$

As a result, $\hat{\mathcal{L}}_{i}^{\mathcal{F}}$ implicitly introduces geometric constraints for improving the estimation of $(\tilde{\gamma}_{i}, \tilde{\varphi}_{i})$.

4. Data Generation Workflow

In this section, we introduce our data generation workflow for generating synthetic training data. The workflow is developed to be fully configurable, with an aim to provide various setups of camera intrinsic K, background depth D_t , where t represents the current timestep, as well as diverse combinations of the motions of the camera and the moving objects. Fig. 3 illustrates the data generation workflow,



Figure 3: An illustration of the data generation workflow.

which consists of five distinct steps. **Step 1** initializes a **K** by sampling from a distribution, which is detailed in the supplementary material. **Step 2** randomly generates a **D** based on the focal lengths in **K**. **Step 3** randomly initializes the rotation γ and translation φ for the camera and a set of moving objects, and use them to derive their corresponding transformation matrices **M**. Subsequently, in **Step 4**, each transformation matrix is forwarded along with **K** and **D**_i to a rigid flow module to derive a rigid flow map \mathbf{F}^{rigid} . The derivation procedure can be formulated as:

$$\mathbf{F}^{rigid} = \frac{1}{\mathbf{D}_{i+1}} \mathbf{K} \mathbf{M}(\mathbf{D}_i) \mathbf{K}^{-1} \mathbf{x} - \mathbf{x}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{r}(\gamma) & \varphi \\ \mathbf{0}_3^{\mathsf{T}} & 1 \end{bmatrix}$$
(10)

where $\mathbf{r}(\cdot)$ represents the function that transforms a Euler angle to a rotation matrix. Please note that the rigid flow map derived from the camera motion is referred to as the ego flow map \mathbf{F}^{ego} , while the rigid flow maps derived from the motions of the moving objects correspond to the object flow maps \mathbf{F}^{obj} . Finally, **Step 5** combines all the flow maps together to obtain the total flow map \mathbf{F}^{total} . The generated \mathbf{F} , \mathbf{K} , \mathbf{D} , and \mathbf{M} are all used in Eq. (8) for training the model parameters in PWVO.

5. Experimental Results

In this section, we present the setups, the quantitative and qualitative results, and a set of analyses.

5.1. Experimental Setup

In order to evaluate the performance of PWVO, the effectness of each component of PWVO, as well as the proposed data generation workflow, we design a number of experiments based on the following experimental setups. We train PWVO on a dataset of 100k samples generated by the proposed data generation workflow, and evaluate the trained PWVO on the validation sets of Sintel [54] and the TartanAir [55]. The detailed configuration for generating

	EPE	R_{err}	T_{err}
VONet [56]	0.909	0.110	0.061
VONet + self-att. [27]	0.894	0.117	0.076
PWVO (naive)	0.829	0.091	0.061
PWVO	0.626	0.081	0.043

Table 1: Comparison of PWVO and the baselines in terms of R_{err} , T_{err} , and EPE. It can be observed that PWVO outperforms the two baselines as well as PWVO (*naive*) by noticeable margins.

the training dataset is provided in the supplementary material. The trained PWVO is then evaluated using the following metrics: (1) the average rotation error R_{err} and translation error T_{err} , which are defined as the L1 error between (γ_i, φ_i) and $(\tilde{\gamma}_i, \tilde{\varphi}_i)$; (2) the end-point-error (EPE) for measuring the quality of the reconstructed $\tilde{\mathbf{F}}_i^{ego}$, which can serve as another metric for evaluating the performance of VO. In our experiments, EPE is defined as the average L1 error between \mathbf{F}_i^{ego} and $\tilde{\mathbf{F}}_i^{ego}$, which is commonly adopted by flow estimation methods.

5.2. Quantitative Results

In this section, we first compare PWVO against two baselines that adopt different mechanisms for suppressing noisy regions in the input observations. Next, we ablatively examine the effectiveness of the components in PWVO. Finally, we validate the importance of $\hat{\mathcal{L}}_i^{\mathcal{F}}$ in optimizing PWVO. The primary objective of our paper is to investigate an effective VO method for suppressing the noises from moving objects contained in inputs. Although the TartanAir dataset [55] consists of lots of training data, its optical flow samples do not contain moving objects, making it inappropriate for our experiments. Therefore, all of our quantitative results are evaluated on the validation set of Sintel [54], as it includes moving objects in the optical flow data samples.

Configurations	EPE	R_{err}	T_{err}
VONet [56]	0.909	0.110	0.061
+ pixel-wise	1.07	0.106	0.055
+ $\mathcal{U}^{\mathcal{R}}$ + $\mathcal{U}^{\mathcal{T}}$ (i.e., PWVO (<i>naive</i>))	0.829	0.091	0.061
$+ \mathcal{U}^{\mathcal{D}} + \mathcal{U}^{\mathcal{F}}$	0.766	0.087	0.062
+ Selection Module (i.e., PWVO)	0.626	0.081	0.043

Table 2: Ablation study for the effectiveness of the components in PWVO. The pixel-wise predictions are averaged to generate final outputs by default, if the selection module is not adopted.

5.2.1 Comparison of PWVO and the Baselines

In this experiment, we compare PWVO against VONet [56] and its variant with self-attention mechanism [27], which are employed as the baselines and are denoted as VONet and VONet+self-att., respectively. VONet is implemented using a similar architecture as the encoding stage of PWVO, and directly predicts rotation and translation from ψ . For PWVO, we consider two different configurations: PWVO (naive) and PWVO, where the former directly derives $(\tilde{\gamma}_i, \tilde{\varphi}_i)$ from $(\mathcal{R}_i, \mathcal{T}_i)$ by performing an average operation instead of using the selection module. Moreover, PWVO (naive) does not take into account the uncertainty maps in its $\mathcal{L}_{i}^{\mathcal{D}}$ and $\hat{\mathcal{L}}_{i}^{\mathcal{F}}$, and simply resorts to L1 and L2 losses when optimizing its $\tilde{\mathbf{D}}_{i+1}$ and $\tilde{\mathbf{F}}_{i}^{ego}$, respectively. The comparison results are shown in Table 1. It can be observed that both versions of PWVO are able to outperform the baselines in terms of R_{err} , T_{err} , and EPE, validating the effectiveness of the pixel-wise prediction mechanism.

5.2.2 The Effectiveness of the Components in PWVO

In this section, we ablatively examine the effectiveness of each component of PWVO by gradually incorporating them into the framework. The results are reported in Table 2. Please note that $\mathcal{U}^{\mathcal{D}}$ and $\mathcal{U}^{\mathcal{F}}$ are the uncertainty maps used in $\hat{\mathcal{L}}^{\mathcal{D}}$ and $\hat{\mathcal{L}}^{\mathcal{F}}$, which are detailed in the supplementary material. It can be observed that, when simply incorporating the pixel-wise design into VONet without uncertainty estimation, the performance degrades slightly. However, when incorporating both the pixel-wise design and the uncertainty estimation for $\mathcal{U}^{\mathcal{R}}$ and $\mathcal{U}^{\mathcal{T}}$, PWVO (*naive*) becomes capable of outperforming VONet, indicating that the proposed pixel-wise design is complementary to the uncertainty estimation strategy. The results also reveal that the performance of the model keeps increasing when each new component is added, validating that all them are crucial for PWVO.

5.2.3 Importance of the Reconstruction Loss $\hat{\mathcal{L}}^{\mathcal{F}}$

In this section, we validate the importance of the reconstruction loss discussed in Section 3.5. Our hypothesis is that incorporating an additional reconstruction loss term $\hat{\mathcal{L}}^{\mathcal{F}}$ could introduce geometric constraints for improving the performance of PWVO. To validate the assumption, we train the

	EPE		R_{err}		T_{err}	
	w/ $\hat{\mathcal{L}}^{\mathcal{F}}$	w/o $\hat{\mathcal{L}}^{\mathcal{F}}$	w/ $\hat{\mathcal{L}}^{\mathcal{F}}$	w/o $\hat{\mathcal{L}}^{\mathcal{F}}$	w/ $\hat{\mathcal{L}}^{\mathcal{F}}$	w/o $\hat{\mathcal{L}}^{\mathcal{F}}$
VONet [56]	0.909	1.276	0.110	0.113	0.061	0.069
VONet + self-att. [27]	0.894	1.312	0.117	0.118	0.076	0.065
PWVO	0.829	1.241	0.091	0.144	0.061	0.095

Table 3: Validation of the effectiveness of the additional loss term $\hat{\mathcal{L}}^{\mathcal{F}}$.



Figure 4: A comparison of VONet and PWVO via the highlighted pixels in their saliency maps.

baselines and PWVO with and without the reconstruction loss $\hat{\mathcal{L}}^{\mathcal{F}}$ and analyze their results, which are summarized in Table 3. It can be observed that, with the help of the reconstruction loss, nearly all the approaches are able to further enhance their performance in terms of EPE, R_{err} , and T_{err} . This evidence thus supports our hypothesis that optimizing PWVO with $\hat{\mathcal{L}}^{\mathcal{F}}$ is indeed beneficial.

5.3. Qualitative Results

In this section, we examine the qualitative results for validating the designs adopted by PWVO.

5.3.1 Examination of the Ability for Dealing with Noises through Saliency Map

Fig. 4 compares PWVO and the baseline VONet from the perspective of their saliency maps, which highlight the pixels that contribute to the predictions of $(\tilde{\gamma}_i, \tilde{\varphi}_i)$. The first column shows an input \mathbf{F}^{total} and its corresponding \mathbf{F}^{obj} from Sintel, the second column depicts the saliency maps of VONet and PWVO using their integrated gradients [57], and the third column overlaps the saliency maps with \mathbf{F}^{obj} . It can be observed that the highlighted pixels of VONet's saliency map are widely scattered, and cover both the object regions and the background. In contrast, the highlighted pixels of PWVO's saliency map only fall on the background. This observation thus confirms our hypothesis that PWVO is capable of effectively suppressing the influence of noises when performing VO tasks.

5.3.2 Examination of the Uncertainty Map

In order to examine if the uncertainty maps predicted by PWVO can correctly capture moving objects from their background, we further showcase an example selected from Sintel and visualize its input \mathbf{F}^{total} , \mathbf{F}^{ego} , and \mathbf{F}^{obj} in the



Figure 5: A comparison of the uncertainty map predicted by PWVO v.s. the error maps and \mathbf{F}^{obj} .

first row of Fig. 5, as well as the uncertainty map $\hat{\mathcal{U}}$ and the error maps of $(\tilde{\gamma}_i, \tilde{\varphi}_i)$ predicted by PWVO in the second row of Fig. 5. It can be observed that $\tilde{\mathcal{U}}$ is highly correlated with the error maps and \mathbf{F}^{obj} , implying that $\tilde{\mathcal{U}}$ is indeed able to capture the noises in the input observations. Since $\tilde{\mathcal{U}}^{\mathcal{R}}$ and $\tilde{\mathcal{U}}^{\mathcal{T}}$ are similar in this case, we only depict one of them in Fig. 5.

5.3.3 Evaluation on the Sintel Validation Set

Fig. 6 presents several examples for demonstrating the qualitative evaluation results of PWVO on the Sintel validation set. It can be observed that the predicted $\hat{\mathbf{F}}^{ego}$ and $\hat{\mathbf{F}}^{obj}$ align closely with the ground truth annotations, and the estimated uncertainty maps are highly correlated with the error maps of $\hat{\mathbf{F}}^{ego}$. This experimental evidence therefore validates the fact that PWVO trained on the dataset generated by the proposed data generation workflow can deliver favorable results on the validation set of the Sintel dataset as well. Please note that visualizations of additional examples are provided in the supplementary material.

5.3.4 Evaluation on the Tartan Validation Set

Sintel and TartanAir datasets differ in that the data samples from TantanAir only involve the motion of the camera, while the data samples from Sintel may involve both the motions of the camera and the objects. Although there is no moving object contained in the optical flow data samples of the TartanAir dataset, we still evaluate our PWVO on it as the qualitative results reveal some interesting insights. It can be observed from Fig. 6 that the uncertain regions in \mathcal{U} highly correlate with the regions of moving objects. On the other hand, Fig. 7 shows that when evaluated on the TartanAir dataset, the uncertain regions become sparse, with no specific target to focus on. This evidence reveals that most of the regions can be utilized by PWVO to generate the final $(\tilde{\gamma}_i, \tilde{\varphi}_i)$, as there exists no noisy region caused by moving objects in the data samples. These two examples validate our hypothesis that the uncertainty maps adopted by PWVO can indeed reflect the noisy regions in the input observations, allowing the selection module to suppress

those regions to derive $(\tilde{\gamma}_i, \tilde{\varphi}_i)$. Please note that in these two figures and Fig. 5 of the main manuscript, the error maps $\mathbf{F}^{ego} - \tilde{\mathbf{F}}^{ego}$ are normalized to allow the regions of differences to be visualized, since the original error values are relatively small.

6. Limitations and Future Directions

Albeit effective, PWVO still has limitations in certain hard scenarios. For example, in the case that moving objects cover most of the regions in the input observations, PWVO might be misled and treats them as $\tilde{\mathbf{F}}^{ego}$. This is due to the lack of sufficient reference information to derive the motion of the camera, which might lead to severe negative impacts on other contemporary VO techniques as well. The future direction of this research is to further extend PWVO to incorporate random occlusion masks or noises into the input observations, and introduce more challenging scenarios to the data generation workflow to reflect more difficult scenarios. From the experimental evidence presented in this paper, we believe that pixel-wise prediction is a promising avenue to explore and investigate in the realm of VO, and may lead to further breakthrough in many benchmarks.

7. Conclusion

In this paper, we proposed the concept of utilizing pixelwise predictions in VO. To achieve this objective, we developed a PWVO framework, which integrates pixel-wise predictions based on the estimated uncertainty maps to derive the final ($\tilde{\gamma}_i, \tilde{\varphi}_i$). In order to provide comprehensive data for training PWVO, we designed a fully configurable data generation workflow for generating synthetic training data. In our experiments, we presented results evaluated on the validation set of Sintel. The results demonstrated that PWVO can outperform the baselines in terms of R_{err} , T_{err} , and EPE. In addition, our analyses validated the effectiveness of the components adopted by PWVO, and demonstrated that the designs adopted in PWVO can indeed capture the noises, and suppress the influence from them.

Acknowledgements

The authors acknowledge the supports from the National Science and Technology Council (NSTC) in Taiwan under grant numbers MOST 111-2223-E-007-004-MY3 and MOST 111-2628-E-007-0, as well as the financial support from MediaTek Inc., Taiwan. The authors would also like to acknowledge the donation of the GPUs from NVIDIA Corporation and NVIDIA AI Technology Center (NVAITC) used in this research work. The authors thank National Center for High-Performance Computing (NCHC) for providing computational and storage resources.





References

- Kishore Konda and Roland Memisevic. Learning visual odometry with a convolutional network. In VISAPP International Conference on Computer Vision Theory and Applications, 2015.
- [2] Pulkit Agrawal, João Carreira, and Jitendra Malik. Learning to see by moving. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2015.
- [3] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In Proc. IEEE Int. Conf. on Computer Vision (ICCV), 2015.
- [4] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.
- [5] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Imagebased localization using lstms for structured feature correlation. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2017.
- [6] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Proc. Robotics: Science and Systems (RSS)*, 2018.
- [7] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *Proc. European Conf. on Computer Vision (ECCV)*, 2018.
- [8] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proc. IEEE Int. Conf. on Computer Vision Workshop (ICCVW)*, 2017.
- [9] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2017.
- [10] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proc. IEEE Int. Conf. on Computer Vision* (*ICCV*), pages 2938–2946, 10 2015.
- [11] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.
- [12] Ming Cai, Chunhua Shen, and Ian Reid. A hybrid probabilistic model for camera relocalization. In *Proc. British Machine Vision Conf. (BMVC)*, 2018.
- [13] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564, 2017.

- [14] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixé. Understanding the limitations of cnn-based absolute camera pose regression. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [15] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [16] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In Proc. IEEE Int. Conf. on Computer Vision Workshop (IC-CVW), 2017.
- [17] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.
- [18] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. In *IEEE Robotics Autom. Lett.*, 2018.
- [19] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.
- [20] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving deeper into convolutional neural networks for camera relocalization. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.
- [21] Fei Xue, Qiuyuan Wang, Xin Wang, Wei Dong, Junqiu Wang, and Hongbin Zha. Guided feature selection for deep visual odometry. In *Proc. Asian Conf. on Computer Vision* (ACCV), 2018.
- [22] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [23] Gabriele Costante, Michele Mancini, Paolo Valigi, and Thomas A. Ciarfuglia. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. In *IEEE Robotics Autom. Lett.*, 2016.
- [24] Gabriele Costante and Thomas A. Ciarfuglia. Ls-vo: Learning dense optical subspace for robust visual odometry estimation. In *IEEE Robotics Autom. Lett.*, 2018.
- [25] Peter Muller and Andreas Savakis. Flowdometry: An optical flow and deep learning based approach to visual odometry. In *Proc. IEEE Winter Conf. on Applications of Computer Vision* (WACV), 2017.
- [26] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. arXiv preprint arXiv:1909.03557, 2019.

- [27] Hamed Damirchi, Rooholla Khorrambakht, and Hamid D. Taghirad. Exploring self-attention for visual odometry. *arXiv*, abs/2011.08634, 2020.
- [28] Emilio Parisotto, Devendra Singh Chaplot, Jian Zhang, and Ruslan Salakhutdinov. Global pose estimation with an attention-based recurrent network. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW), pages 237–246, 2018.
- [29] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10542–10551, 2019.
- [30] Thomas A. Ciarfuglia, Gabriele Costante, Paolo Valigi, and Elisa Ricci. Evaluation of non-geometric methods for visual odometry. *Robotics Auton. Syst.*, 62(12):1717–1730, 2014.
- [31] Tianguang Zhang, Xiaodong Liu, Kolja Kühnlenz, and Martin Buss. Visual odometry for the autonomous city explorer. In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS), pages 3513–3518, 2009.
- [32] Xin-Yu Kuo, Chien Liu, Kai-Chen Lin, Evan Luo, Yu-Wen Chen, and Chun-Yi Lee. Dynamic attention-based visual odometry. In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS), pages 5753–5760, 2020.
- [33] Masaya Kaneko, Kazuya Iwami, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Mask-slam: Robust featurebased monocular SLAM by masking using semantic segmentation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW), pages 258–266, 2018.
- [34] Berta Bescós, José M. Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics Autom. Lett.*, 3(4):4076–4083, 2018.
- [35] Ting Sun, Yuxiang Sun, Ming Liu, and Dit-Yan Yeung. Movable-object-aware visual SLAM via weakly supervised semantic segmentation. *CoRR*, abs/1906.03629, 2019. URL http://arxiv.org/abs/1906.03629.
- [36] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [37] Feng Gao, Jincheng Yu, Hao Shen, Yu Wang, and Huazhong Yang. Attentional separation-and-aggregation network for self-supervised depth-pose learning in dynamic scenes. *CoRR*, abs/2011.09369, 2020.
- [38] Bin Li, Shuling Wang, Haifeng Ye, Xiaojin Gong, and Zhiyu Xiang. Cross-modal knowledge distillation for depth privileged monocular visual odometry. *IEEE Robotics and Automation Letters*, 7(3):6171–6178, 2022. doi: 10.1109/LRA. 2022.3166457.

- [39] Seokju Lee, François Rameau, Fei Pan, and In So Kweon. Attentive and contrastive learning for joint depth and motion field estimation. *CoRR*, abs/2110.06853, 2021.
- [40] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In Proc. Conf. on Neural Information Processing Systems (NeurIPS), pages 5574–5584, 2017.
- [41] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: Learning sfm from sfm. In *Proc. European Conf. on Computer Vision (ECCV)*, 2018.
- [42] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Real-time monocular slam: Why filter? In 2010 IEEE International Conference on Robotics and Automation, pages 2657–2664, 2010. doi: 10.1109/ROBOT.2010.5509636.
- [43] Jakob Engel, Jürgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In 2013 IEEE International Conference on Computer Vision, pages 1449– 1456, 2013. doi: 10.1109/ICCV.2013.183.
- [44] Xu-Yang Dai, Qing-Hao Meng, and Sheng Jin. Uncertaintydriven active view planning in feature-based monocular vslam. *Applied Soft Computing*, 108:107459, 2021. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2021. 107459. URL https://www.sciencedirect.com/ science/article/pii/S1568494621003823.
- [45] Gabriele Costante and Michele Mancini. Uncertainty estimation for data-driven visual odometry. *IEEE Trans. Robotics*, 36(6):1738–1757, 2020.
- [46] Raul Mur-Artal and Juan D. Tardós. Orb-slam2: an opensource slam system for monocular, stereo and rgb-d cameras. In *IEEE Trans. Robotics*, 2017.
- [47] Georg Klein and David William Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (IS-MAR)*, 2007.
- [48] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium (IV)*, 2011.
- [49] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.
- [50] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsdslam: Large-scale direct monocular. In Proc. European Conf. on Computer Vision (ECCV), 2014.
- [51] Richard A. Newcombe, Steven Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In Proc. IEEE Int. Conf. on Computer Vision (ICCV), 2011.
- [52] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the

coordconv solution. In Proc. Conf. on Neural Information Processing Systems (NeurIPS), pages 9628–9639, 2018.

- [53] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL https://doi.org/10.1145/358669.358692.
- [54] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 611–625, 2012.
- [55] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS), 2020.
- [56] Wenshan Wang, Yaoyu Hu, and Sebastian A. Scherer. Tartanvo: A generalizable learning-based vo. In *Proc. Conf. on Robot Learning (CoRL)*, 2020.
- [57] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. Int. Conf. on Learning Representations Workshop (ICLRW)*, 2014.