sensors

MDPI

Article

WPO-Net: Windowed Pose Optimization Network for Monocular Visual Odometry Estimation

Nivesh Gadipudi [1], Irraivan Elamvazuthi [1,*], Cheng-Kai Lu [1], Sivajothi Paramasivam [2] and Steven Su [3]

[1] Smart Assistive and Rehabilitative Technology (SMART) Research Group, Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Bandar Seri Iskandar 32610, Malaysia; nivesh_18001319@utp.edu.my (N.G.); chengkai.lu@utp.edu.my (C.-K.L.)
[2] School of Engineering, UOWM KDU University College, Shah Alam 40150, Malaysia; siva@kdu.edu.my
[3] School of Biomedical Engineering, University of Technology Sydney, Ultimo 2007, Australia; Steven.Su@uts.edu.au
* Correspondence: irraivan_elamvazuthi@utp.edu.my

Abstract: Visual odometry is the process of estimating incremental localization of the camera in 3-dimensional space for autonomous driving. There have been new learning-based methods which do not require camera calibration and are robust to external noise. In this work, a new method that do not require camera calibration called the “windowed pose optimization network” is proposed to estimate the 6 degrees of freedom pose of a monocular camera. The architecture of the proposed network is based on supervised learning-based methods with feature encoder and pose regressor that takes multiple consecutive two grayscale image stacks at each step for training and enforces the composite pose constraints. The KITTI dataset is used to evaluate the performance of the proposed method. The proposed method yielded rotational error of 3.12 deg/100 m, and the training time is 41.32 ms, while inference time is 7.87 ms. Experiments demonstrate the competitive performance of the proposed method to other state-of-the-art related works which shows the novelty of the proposed technique.

Keywords: visual odometry; pose estimation; pose optimization; deep learning

check for updates

Citation: Gadipudi, N.; Elamvazuthi, I.; Lu, C.-K.; Paramasivam, S.; Su, S. WPO-Net: Windowed Pose Optimization Network for Monocular Visual Odometry Estimation. *Sensors* **2021**, *21*, 8155. https://doi.org/10.3390/s21238155

Academic Editor: Mahmoud Hassaballah, Zhengming Ding and Senthil Yogamani

Received: 8 October 2021
Accepted: 21 November 2021
Published: 6 December 2021

1. Introduction

Autonomous vehicles, including unmanned aerial vehicles (UAV), unmanned ground vehicles (UGV), and unmanned underwater vehicles (UUV), are increasingly used to explore the different difficult and dangerous environments to minimize human interaction. In addition, mobile robots became an integral part of the present industry evolution for logistics and supply chain management. Estimating the ego-motion or continuous localization of the robot in an environment is a fundamental long-standing challenge in autonomous navigation. Traditionally, continuous localization is performed using sensors, such as global positioning systems (GPS), inertial sensors, and wheel encoders for ground robots. Traditional methods suffer from accumulated drift and GPS is constrained to only open environments. Recent studies expressed immense interest to perform the localization task using cameras due to vast information. The method of performing the continuous localization using cameras or visual-only sensors is known as visual odometry (VO). The applications of visual odometry vary widely from scene reconstruction [1], indoor localization [2], biomedical applications [3], and virtual and augmented reality [4] to self-driving vehicles [5].

VO acts as a fundamental block of a similar set of algorithms, such as visual simultaneous localization and mapping (VSLAM) and structure from motion (SfM). State-of-the-art are the earliest methods of VO algorithms and are classified into sparse methods [6,7] and dense methods [8] based on the minimization objectives. Sparse methods use the features extracted from consecutive images to estimate the motion by minimizing reprojection errors. Dense methods concentrate on individual pixels of consecutive images to reconstruct a more comprehensive scene and work on the principle of photometric consistency. Though the state-of-the-art methods are efficient in estimating the motion, these methods require a series of complex pipelines consisting of individual components addressing the multi-view geometric tasks which require hard tuning based on the environment. A slight malfunctioning of a subcomponent can result in the degradation of the entire pipeline. However, estimating visual odometry is a multi-view geometric problem and requires knowledge of the underlying 3-dimensional (3D) structure. In addition, these methods are less generalized, which means they are not intelligent to learn from the different modalities of environments.

Considering the above shortcomings of the state-of-the-art methods, researchers of the computer vision community concentrated on alternative algorithms based on the learning paradigm. Learning-based algorithms gained massive attention due to their capability of implicitly learning the hidden representations with more generalization ability. Recently, methods using deep learning revealed superior performance over traditional methods in object classification, detection, and recognition [9,10]. Earlier learning-based methods used recurrent neural networks to improve the long-term temporal dependencies that mitigate pose drift problems [11]. On the other hand, some methods used optical flow estimates extracted from images to feed the networks [12]. The resultant of either of these are larger network parameters with high computational time. Current work deals only with monocular videos and learning-based methods using left-right consistency for training are not included in the evaluation [13,14].

The main aim of this paper is to improve pose predictions derived from convolutional neural networks given a set of images stacks and ground truths using windowed optimization. This is achieved by multiple forward passes from multiple inputs and a single back-propagation based on cumulative loss. From a point, the proposed network can be viewed as multiple siamese networks that share the same parameters among the same networks. The main contributions of this paper are:

1. A new learning-based optimization method without any additional modifications to the network is proposed.
2. Proposed network is independent of optical flow preprocessing and temporal processing modules, such as recurrent neural networks. Most importantly, WPO-Net is relatively small and consists of only 0.48 million parameters.
3. Experiments are performed to emphasize the importance of data augmentation in learning-based VO methods and the effect of varying window sizes in the proposed optimization framework.
4. Comparative experiments showcase the competitive performance of the proposed method with other geometric or state-of-the-art methods, supervised and unsupervised learning-based methods.

The paper is organized as follows: Section 1.1 presents an overview of the published related works. Section 2 describes the building blocks of the method, including network architecture, windowed pose optimization technique, and loss function. Section 3 presents details of training and testing datasets, hardware, and software environments. In addition, this section also presents the evaluation of the present method on the KITTI dataset, data augmentation, and ablation tests.

1.1. Related Work

VO estimation is a long-standing multi-view geometry problem. Over the years, there have been several approaches that are being used to address the task of VO estimation. These algorithms can be classified into two distinctive types, namely state-of-the-art methods and learning-based methods. State-of-the-art methods are also referred to as geometric or traditional methods, alternatively.

1.1.1. State-of-the-Art Methods

State-of-the-art or geometric methods are further classified into the sparse of feature-based methods and direct or dense methods. As discussed, feature-based methods work by minimizing the reprojection error between features from consecutive frames. The feature extracted can be edges, lines, or blobs. Most famous feature extraction methods are ORB [15], FAST [16], and SURF [17]. Some of the early feature-based methods, such as in Reference [7], used filtering techniques to simultaneously optimize the map points and position of the robot. The major drawback associated with filtering-based VO/VSLAM is the increase in computational cost as the map grows. This issue was addressed by keyframe-based algorithms, which use independent threads for mapping and tracking threads [4]. These keyframe-based methods use bundle adjustment as the backbone of optimizing the position and map points to reduce drifts. Down the road, these algorithms became more efficient and are highly dependent on the robustness of feature extractors. ORB-SLAM [6] and VISO2 [18] are some of the most efficient real-time feature-based VO/VSLAM algorithms. Nevertheless, feature-based algorithms suffer from textureless and noise-induced regions. On the other hand, direct methods minimize the pixelwise reprojection error from consecutive images. Direct methods can reconstruct more comprehensive 3D scenes but are computationally expensive and limit the real-time usability of these algorithms [8,19]. A combination of direct and feature-based methods are also developed to estimate the pose using the features and the regions surrounding the pixels, and these are known as semi-direct methods [20]. However, the direct method works on the principle of photometric consistency and is not designed to deal with large viewpoint changes.

1.1.2. Learning-Based Methods

Learning-based methods are the most recent VO algorithms. Due to the continuous increase in the availability of graphic processing units (GPUs), benchmark datasets, such as KITTI [21], and synthetic data generation frameworks, such as CARLA [22] and TartanAir [23], there has been a shift in increased research towards learning-based algorithms. Learning-based methods are robust to unmodeled noise and environmental changes and work by learning the hidden feature representations. Learning-based methods are further classified into supervised and unsupervised based on the learning paradigms. One of the main challenges of learning-based methods is adapting to the architectures that were being used for 2D tasks, such as classification, recognition, and localization. These architectures operate by taking a single image as input, but the VO estimation requires a stack of consecutive images.

Supervised learning-based methods rely on the ground truth 6 degrees of freedom (DOF) poses to optimize the parameters. Earliest learning-based method can be dated back to 2008 [24]. Later, the VO estimation was recognized as a regression task. The invention of architectures, such as PoseNet [25], used to regress the absolute 6 DOF pose, and FlowNet [26], used for optical flow extraction between two images, provided great support for learning-based VO estimation algorithms. Supervised learning-based methods learn the hidden mapping by taking optical flow or raw images. LS-VO [27] and Flowdometry [12] learn to predict the pose by used optical flow. However, these methods involve computationally expensive preprocessing to extract the optical flow from images. Methods, such as DeepVO [11] and PCGRU [28], used recurrent neural networks to minimize the prediction errors. Another interesting development includes uncertainty quantification in the pose prediction process [29]. DeepVO estimates the covariance matrix along with pose estimation. This work is highly motivated by the fact that this uncertainty quantification can be used to adaptively weigh the translation and rotational components of the pose estimates. Reference [30] estimates the 2 DOF pose for ground vehicles by neglecting the less significant movement along the other four axis. The proposed WPO-Net inherits some architectural design philosophies, such as rectangular convolutions from Reference [30].

On the other hand, unsupervised methods work on the foundational principle of single view image synthesis. These methods operate in complex end-to-end format involving several networks to address tasks, such as depth estimation, dynamic region masking, and pose estimates. SfMLearner [31] is designed to estimate the depth and pose by neglecting unexplainable pixels. GeoNet [32] further included the dynamic object compensation to avoid the erroneous pose estimates. CM-VO [33] proposed a confidence quantification and refining the trajectory based on the confidence. Though unsupervised methods eliminate the requirement of ground truths, the performance of these methods is not on par with the supervised learning-based methods. To address the above problems in learning-based methods, a windowed optimization approach is presented in this paper. The proposed method optimizes the pose of a short window of images using the trajectory consistency constrain and is analogous to windowed bundle adjustment in traditional methods.

2. Methodology

This section includes the introduction to subcomponents of the proposed method. The entire framework is composed of two subcomponents, namely a feature encoder and pose regressor. The feature encoder transforms the high-level gray images into a compact global feature descriptor. The extracted feature descriptor is transformed into a 6 DOF pose estimate by the pose regressor. Further, CNN-based windowed pose optimization and loss function used for training are explained in Sections 2.4 and 2.5, respectively.

2.1. Preprocessing

The original raw grayscale input images of size 1241 × 376 are resized to 640 × 192 to meet the specifications of the proposed network and to reduce the memory consumption of the GPU. A general procedure of standardizing the images about mean and variance is used to narrow down the distribution and to pace up the convergence. Two consecutive images are stacked along the channels to serve as the input to the feature encoder. A temporal skipping strategy for augmenting the data is used by selecting a consecutive random frame within an interval of 0 to 4 in the forward direction to learn more distinctive and complicated mapping.

2.2. Feature Encoder

VO or continuous ego-motion estimation requires consecutive image pairs. In traditional methods, this is performed by feature matching or photometric consistency across the frames of the sequence. In learning-based methods using deep learning, the hidden representations of the images are automatically extracted to estimate the 6 DOF pose. The proposed feature encoder takes in a stack of two grayscale images of size 640 × 192 at each training step. The details of the architecture of the feature encoder used for this method are presented in Table 1.

Table 1. Architecture of the feature encoder. The filter's size decreases as the depth of the network increases.

Layer	Kernel Size	Channels	Stride	Dilation
Input	-	2	-	-
Layer-1	3 × 9	16	2	2
Layer-2	3 × 9	16	2	1
Layer-3	3 × 7	32	2	2
Layer-4	3 × 7	32	2	1
Layer-5	3 × 5	64	1	2
Layer-6	3 × 5	64	1	1
Layer-7	2 × 2	64	2	1

Feature encoder consists of seven layers using the rectangular kernels, except the last one. A combination of different strides and dilations are used to efficiently reduce the size of the network by extracting the features with greater receptive coverage. The last layer is a special convolutional pooling layer to downsample the dimensions of the descriptor. Batch normalization and ELU (exponential linear unit) are used for every layer to accelerate the convergence.

2.3. Pose Regressor

The extracted global feature descriptor from the feature encoder is transformed into a 6 DOF pose estimate by feeding into a two-layered MLP (multilayer perceptron). The first layer consists of 256 nodes with ELU activation. The output or the second layer of the pose regressor consists of 6 nodes with linear activation. The output vector represents the translations and rotations in Euler angles about each axis. The predicted values are quantitatively used to estimate the loss with the labeled ground truth.

2.4. Windowed Pose Optimization

Proposed approach adopts a unique strategy motivated by the benefits of windowed bundle adjustment in reducing drifts. The proposed networks use four images of the video sequence and stack them into 3 overlapping samples to feed the network. Let $\{I_t, I_{t+1}, I_{t+2}, I_{t+3}\}$ be the four consecutive images stacked into $\{I_{t,t+1}, I_{t+1,t+2}, I_{t+2,t+3}\}$, as shown in Figure 1. First, each training iteration consists of forward propagating a triplet network using three consecutive image stacks. Second, the gradients are propagated backward by estimating the cumulative loss of predictions from triplets. A detailed explanation of the formulated loss function used for training is presented in Section 2.5. A $SE(3)$ composition layer is used to estimate the implicit transformations $\{T_{t \to t+2}, T_{t \to t+3}, T_{t+1 \to t+3}\}$ of unrelated stacks during training. $SE(3)$ composition layer is used to transform the predicted 6 DOF pose estimate $\mathfrak{se}(3)$ into $SE(3)$ transformation matrix, and vice versa, ensuring the differentiable properties. The elements of $\mathfrak{se}(3)$ can be mapped to $SE(3)$ by using an exponential map and $SE(3)$ to $\mathfrak{se}(3)$ using the logarithmic map.

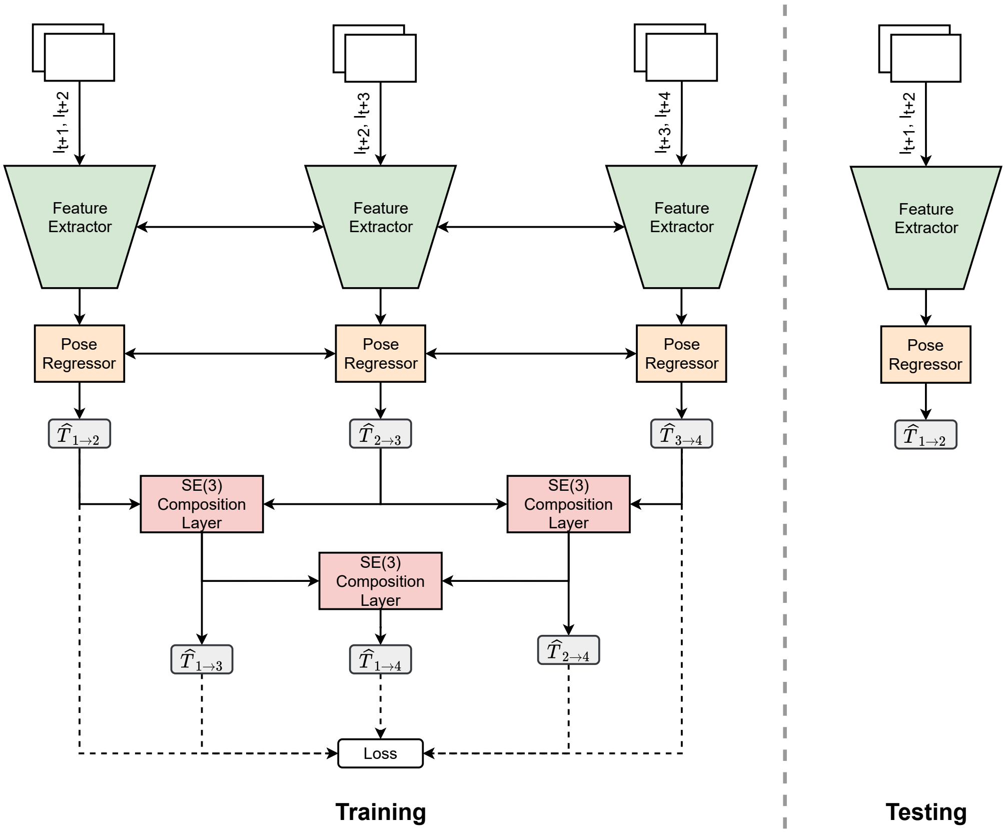


Figure 1. Overview of the WPO-NET. $SE(3)$ composition layer is used to derive the composite poses from predicted poses.

Consider $u = [\mathrm{x}, \mathrm{y}, \mathrm{z}, \omega_1, \omega_2, \omega_3] \in \mathfrak{se}(3)$, where $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and $(\omega_1, \omega_2, \omega_3)$ representing the translations and Euler angles. The corresponding generators of $\mathfrak{se}(3)$ representing the derivatives of translations and rotations about each axis can be formulated as Equation (1):

$$\begin{gathered} G_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, G_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, G_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ G_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, G_5 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, G_6 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{gathered} \tag{1}$$

For mathematical convenience, we denote translations u and rotations ω separately. The linear combinations of generators can written as Equation (2):

$$\delta = (p \quad \omega) = xG_1 + yG_2 + zG_3 + \omega_1 G_4 + \omega_2 G_5 + \omega_3 G_6 \in \mathfrak{se}(3), \tag{2}$$

where G_1, G_2, G_3 are partial derivatives of translations about X, Y, Z axis with linear combinations $p = xG_1 + yG_2 + zG_3$, respectively. G_4, G_5, G_6 are partial derivatives of Euler angles $(\omega_1, \omega_2, \omega_3)$ on the X, Y, Z axis with linear combinations $\omega = \omega_1 G_4 + \omega_2 G_5 + \omega_3 G_6$, respectively. The linear combinations of generators representing $\delta = (p \quad \omega) \in \mathfrak{se}(3)$ are transformed to $SE(3)$ by applying the exponential mapping

$$\exp(\delta) = \left(\begin{array}{cc} e^{\omega_x} & \mathrm{V}p \\ 0 & 1 \end{array} \right). \tag{3}$$

Using Taylor expansion, exponential map of ω and V can be formulated as:

$$\begin{gathered} e^{\omega_{\mathrm{x}}} = \mathrm{I}_3 + \frac{\sin\theta}{\theta}\omega_x + \frac{1-\cos\theta}{\theta^2}\omega_x^2, \\ \mathrm{V} = \mathrm{I}_3 + \frac{1-\cos\theta}{\theta^2}\omega_x + \frac{\theta - \sin\theta}{\theta^3}\omega_x^2, \end{gathered} \tag{4}$$

where $\theta = |\omega|$, ω_x is the skew-symmetric matrix from the linear combination of rotational generators. Similarly, $T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$, where $T \in SE(3)$. $R \in SO(3)$ and $t \in \mathbb{R}^3$ are translational and rotational elements and can be inverted to the logarithmic map using:

$$\begin{gathered} \delta = \begin{pmatrix} p \\ \omega \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \\ \theta = arccos\left(\frac{tr(R) - 1}{2}\right), \\ ln(R) = \frac{\theta}{\sin\theta} \cdot \left(R - R^T\right), \end{gathered} \tag{5}$$

where θ is the axis angle calculated from Equation (5). ω can be recovered from the off-diagonal elements of $ln(R)$ and $p = V^{-1}t$. These pose estimates from $SE(3)$ composition layers are referred to as unrelated stacks due to the reason that these are estimated based on the predicted poses of $\{T_{t\to t+1}, T_{t+1\to t+2}, T_{t+2\to t+3}\}$ corresponding to image stacks $\{I_{t,t+1}, I_{t+1,t+2}, I_{t+2,t+3}\}$ in the forward pass from:

$$\begin{gathered} T_{t\to t+2} = T_{t\to t+1} \odot T_{t+1\to t+2}, \\ T_{t\to t+3} = T_{t\to t+1} \odot T_{t+1\to t+2} \odot T_{t+2\to t+3}, \\ T_{t+1\to t+3} = T_{t+1\to t+2} \odot T_{t+2\to t+3}, \end{gathered} \tag{6}$$

where $\odot$ represents the dot product.

2.5. Loss Function

The training process consists of adjusting the network parameters $\boldsymbol{\theta}$ by minimizing the deviation between predicted $\widehat{u_t}$ and ground truth u_t poses. The conditional probability of the VO problem can be formulated, and optimal parameters $\boldsymbol{\theta}^*$ can be estimated by maximizing the following objective:

$$\begin{gathered} P(U_t|I_t) \;=\; P(u_1,\, u_2,\, u_3,\, \ldots\ldots,\, u_t|i_1,\, i_2,\, i_3, \ldots\ldots,\, i_t), \\ \boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P(u_t \mid I_t, I_{t+1}; \boldsymbol{\theta}). \end{gathered} \tag{7}$$

This method uses a homoscedastic uncertainty-based loss function to automatically choose the weighting coefficient between translational and rotational counterparts. The selected homoscedastic loss function consists of two uncertainty quantification regularization terms $(\hat{s}_p,\ \hat{s}_\omega)$ as given in Equation (8):

$$\mathcal{L}oss = \frac{1}{t}\sum_{k=1}^{t} \mathcal{L}_p \exp(-\hat{s}_p) + \hat{s}_p + \mathcal{L}_\omega \exp(-\hat{s}_\omega) + \hat{s}_\omega, \tag{8}$$

where $\mathcal{L}_p = \|\hat{p}_t - p_t\|_2^2$ and $\mathcal{L}_\omega = \|\hat{\omega}_t - \omega_t\|_2^2$ are the euclidean distance between ground truth $(p_t,\ \omega_t)$ and predicted $(\widehat{p}_t,\ \widehat{\omega_t})$ translational and rotational elements, respectively. Standard networks solely minimize the relative transformational errors. Optimizing the nearest frames by enforcing the geometric constraints using composite poses jointly is the key to maintain lesser drifts. The total loss term consists of directly estimated relative poses with estimated composite poses are written as Equation (9):

$$\begin{gathered} \mathcal{L}oss_{relative} = \mathcal{L}oss_{t\to t+1} \;+\; \mathcal{L}oss_{t+1\to t+2} \;+\; \mathcal{L}oss_{t+2\to t+3}, \\ \mathcal{L}oss_{composite} = \mathcal{L}oss_{t\to t+2} \;+\; \mathcal{L}oss_{t\to t+3} \;+\; \mathcal{L}oss_{t+1\to t+3}, \\ \mathcal{L}oss_{total} = \mathcal{L}oss_{t\to t+1} \;+\; \mathcal{L}oss_{t+1\to t+2} \;+\; \mathcal{L}oss_{t+2\to t+3} \;+\; \mathcal{L}oss_{t\to t+2} \;+\; \mathcal{L}oss_{t\to t+3} \;+\; \mathcal{L}oss_{t+1\to t+3}, \\ \mathcal{L}oss_{total\ (DA)} = \mathcal{L}oss_{t\to t+j} \;+\; \mathcal{L}oss_{t+j\to t+k} \;+\; \mathcal{L}oss_{t+k\to t+l} \;+\; \mathcal{L}oss_{t\to t+k} \;+\; \mathcal{L}oss_{t\to t+l} \;+\; \mathcal{L}oss_{t+k\to t+j}, \end{gathered} \tag{9}$$

where $\mathcal{L}oss_{total\ (DA)}$ is the loss function for samples with data augmentation (DA), and j, k, l are the random values ranging from 0 to 4.

3. Experiments

This section presents the details of the performance evaluation of the proposed method. First, the software and hardware environment used to train and test the proposed method with a set of selected hyperparameters are presented. Second, details of the benchmark and evaluation metrics associated are described. Next, the importance of DA in the VO task is presented by choosing the varying amount of augmented data. Performance of the related works is compared relatively to current method to evaluate the efficiency and accuracy of the current windowed deep optimization technique. Finally, a detailed ablation study is performed on the network to visualize the importance of windowed optimization with a detailed run-time analysis.

3.1. Implementation Details

The network was trained and tested using PyTorch framework in Python on Nvidia 2080S GPU with a memory of 8 GB and Intel i9-10900F at 2.80 GHz. An Adam optimizer with default setting of $\beta_1 = 0.9, \beta_2 = 0.999$ was used, as presented in Reference [34]. The initial learning rate of 0.001 with a half decay rate for every 30 epochs until 150 epochs was selected to train the network. Even though our model only consumes one-fourth of the total GPU available, batch size remained at 32 for training and testing.

3.2. Dataset

We used the KITTI VO benchmark [21] to train and test WPO-Net. The dataset consists of 21 sequences composed of 23,201 images; 11 of the 21 sequences are available with ground truth pose estimates. For this work, we adopted a split used in Reference [31–33,35–37], which reserves 00-08 sequences for training and 09, 10 sequences for testing. A station wagon is used to collect the dataset in outdoor environments with a frequency of 10 frames per second and compromises of challenging scenarios with dynamic objects. The default image size of the images in the dataset is 1241 × 376, and the images are resized to half for training and testing the proposed network to constrain the computational cost. Training data is augmented using a temporal skipping technique, and no DA is involved while testing the network.

Three evaluation metrics, namely absolute trajectory error $(ATE(m))$, translational error $(t_{rel}(\%))$, and rotational error $(r_{rel}(\text{deg}/100\text{ m}))$, are used to efficiently evaluate within various sizes of samples of the present method and related works. Translational and rotational errors are obtained by averaging the subsequence errors from 100 to 800 m with an interval of 100 m.

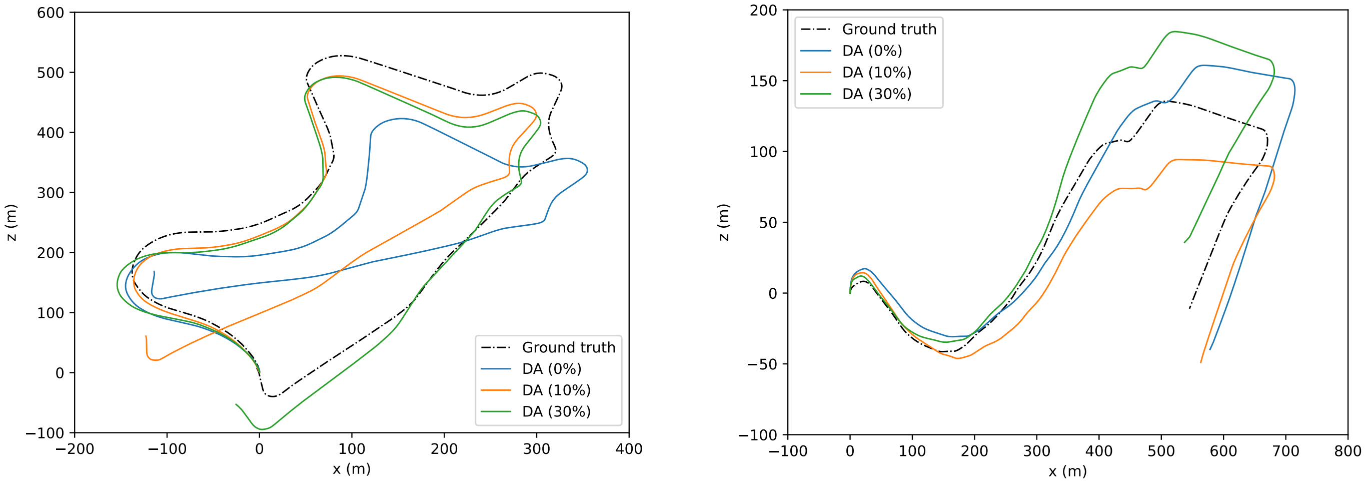
3.3. Effects of Data Augmentation

Data is one of the crucial components for any learning-based paradigm, such as deep learning. This section emphasis on a long-standing yet challenging problem in training deep networks. The majority of supervised learning works adapted a manual weighting approach to tune the balance between the rotational and translational elements, which is time-consuming and needs an extensive parameter search space. However, it is very difficult to derive a quantitative measure between rotational and translational samples in the VO task, and, to avoid these data-related uncertainties and to adaptively weight the elements, a homoscedastic based loss is used. Another interesting direction is to increase the size of the available dataset with techniques, such as random sampling, cropping, and noise addition. A temporal skipping technique is used for this study to augment the data, and the effects of different percentages of augmentation with respect to evaluation metrics are shown in Table 2.

Table 2. Effects of varying quantities of DA on MPO-Net (the least error results are high-lighted in bold text).

DA	ATE	Trans	Rot
(%)	(m)	t_{rel} (%)	r_{rel} (deg/100 m)
0	91.82	12.82	5.07
10	57.52	**7.95**	3.27
20	96.81	9.31	3.91
30	**48.76**	8.57	**3.06**
40	94.03	9.70	3.49
50	79.06	9.38	3.28

The predicted trajectories of the best model DA (30%), second-best DA (10%) are plotted against the ground truth in Figure 2. The overall estimated trajectory trained with DA 30 percent performed well on ATE and translational error (t_{rel}). This study considers ATE as one of the significant evaluation metrics in the aspects of VO tasks to reduce the drift and is often underemphasized.



(**a**) Sequence 09

(**b**) Sequence 10

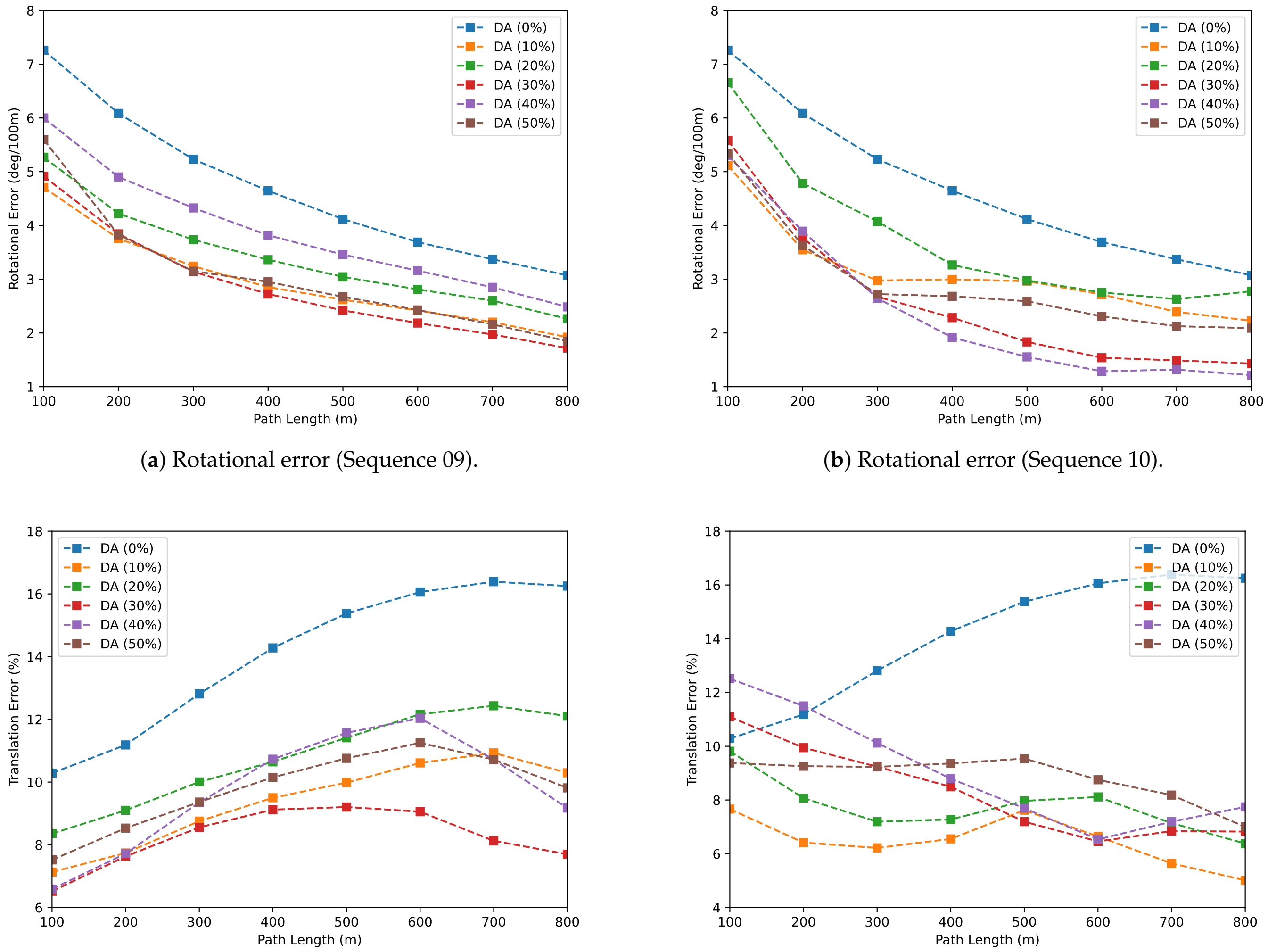
Figure 2. Trajectories of sequences 09 (**a**) and 10 (**b**) under different data augmentation (DA) quantities. *X* and *Y*-axis represent motion along the *Z* (forward) and *X* (left/right) axis of the vehicle in the vehicular frame.

From the experiments, it is evident that increasing the dataset by augmenting does not always result in higher accuracies, especially in a complex multi-view geometry problem, such as VO. The best model for comparison with other related works is chosen to be the dataset with DA (30%). Though the dataset with DA (10%) performed superior to other splits in terms of translational error, the dataset with DA (30%) outperformed it over the other two evaluation metrics. Rotational and translation errors of models trained on different augmentation split and tested on sequences 09, 10 for subsamples are shown in Figure 3. From Figure 3c,d, it can be observed that the model trained on DA with 30 percent is stable and accurate compared to other splits. Similarly, from Figure 3a,b, DA (30%) performed superior to other splits. Though DA (30%) is lagging behind DA (10%) in a singular case (translational error (t_{rel})), overall performance of DA (30%) is better compared to others, and this model is used to compare with the related works in the next section.

3.4. Comparison with Related Works

This section evaluates the proposed method with other significant published works. The proposed WPO-Net is evaluated across three different algorithms. First, Monocular VISO2 [18] and ORB-SLAM [6] are used to evaluate against the state of art algorithms. Second, a supervised version of Reference [35], DeepVO [11], and Flowdometry [12] are employed to compare with the supervised learning-based methods. Though DeepVO and Flowdometry are some of the most prominent supervised learning-based methods, different splits were used for training and testing. To effectively deal with such train-test split discrepancies in comparison with other methods, the average translation, and rotational errors across all sequences are used. Finally, unsupervised learning-based methods, such as in References [31–33,36,37], are included in the comparison with WPO-Net in Table 3.

Although the performance of WPO-Net is slightly unsatisfactory on sequence 09 against VISO2M, the overall performance advantage is higher and accurate. In addition, the current method avoids the complex pipeline involving numerous subsystems, such as VISO2M and ORB-SLAM. On the other hand, WPO-Net performed significantly better on sequence 09 than any other learning-based methods used for comparison. Supervised learning-based methods take the advantage of implicitly learning the scale during the training process. The overall rotational error is minimal in comparison with other methods. This experiment verifies the ability of the learning-based windowed pose optimization technique in improving the accuracy of the system.



(**a**) Rotational error (Sequence 09).

(**b**) Rotational error (Sequence 10).

(**c**) Translational error (Sequence 09).

(**d**) Translational error (Sequence 10).

Figure 3. Comparison of rotational and translational errors of different DA quantities at subsamples of varying length (100 m, 200 m, 300 m, . . . , 800 m) sequences 09 and 10.

Table 3. Comparative results on the KITTI benchmark (data is extracted from the corresponding works/citations, the least error results are high-lighted in bold text).

Method	**Sequence 09**		**Sequence 10**		**Avg**	
	Trans t_{rel} **(%)**	**Rot** r_{rel} **(deg/100 m)**	**Trans** t_{rel} **(%)**	**Rot** r_{rel} **(deg/100 m)**	**Trans** t_{rel} **(%)**	**Rot** r_{rel} **(deg/100 m)**
VISO2M [18]	**7.08**	**1.15**	41.60	32.99	24.34	17.07
ORB-SLAM [6]	-	-	86.51	98.90	30.01	35.53
Flowdometry [12]	12.64	8.04	11.65	7.28	11.42	6.92
DeepVO [11]	-	-	8.11	8.83	**5.96**	6.12
SfM Learner [31]	17.84	6.78	37.91	17.78	27.88	12.28
GeoNet [32]	43.76	16.00	35.6	13.80	39.68	14.90
Zhan et al. [36]	11.92	3.60	12.62	3.43	12.27	3.52
Wang et al. [35]	9.30	3.50	**7.21**	3.90	8.26	3.70
SC-SfM [37]	11.20	3.35	10.10	4.96	10.65	4.16
CM-VO [33]	9.69	3.37	10.01	4.87	9.85	4.12
WPO-Net (proposed)	8.19	3.02	8.95	**3.12**	8.57	**3.06**

3.5. Ablation Study

This section includes the experimentation on the proposed WPO-Net to examine the efficiency of learning-based windowed pose optimization. The conclusion is drawn by training and testing the network with three different window sizes (WS). The WS defines the number of consecutive images used for every single backpropagation. Let WS be equal to n images, and the number of times the network is forward propagated is given by $(n-1)$ with a single backpropagation. When $WS = 2$, the network by default acts as a standard supervised network with one sample input and one sample output. The three different window sizes are selected to observe the efficiency of windowed pose optimization by examining the evaluation metrics. Figure 4 illustrates the number of images used for a single iteration as the windows slide towards the right.

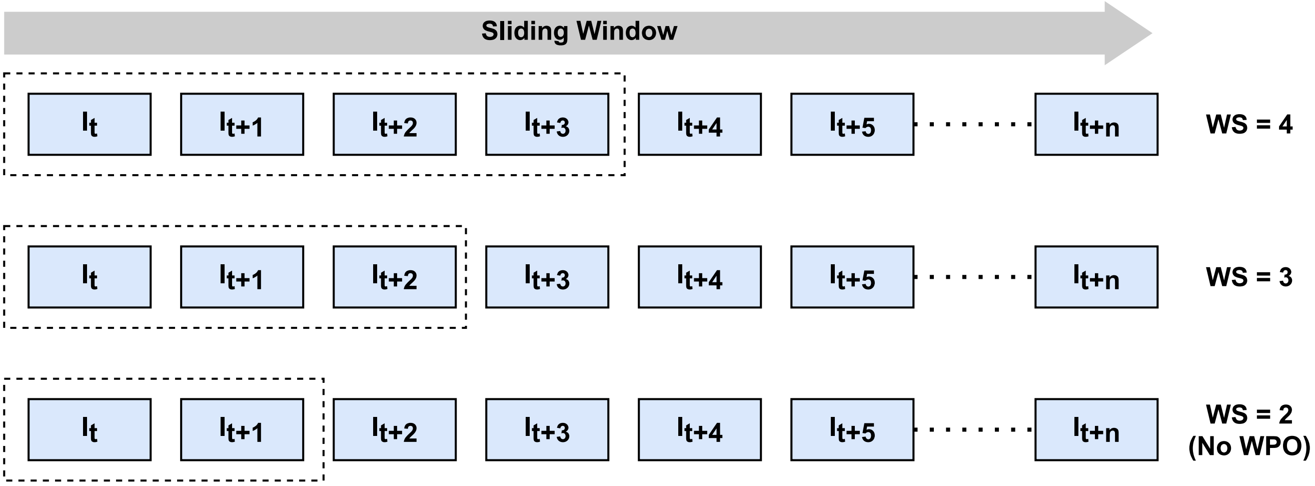


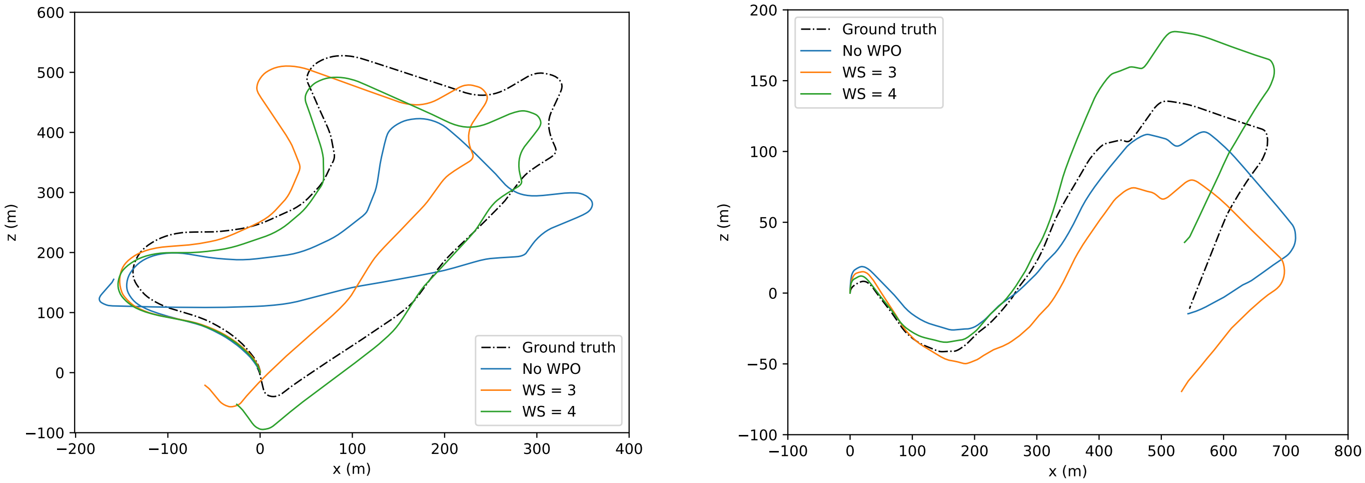
Figure 4. An illustration of the number of images taken as input to the network for $WS = 2, 3, 4$.

All the networks used for comparison in this section are trained and tested with the same split, as mentioned in Section 3.3, with 30 percent of DA. The network with $WS = 4$ was the one used to compare with related work, and the data is derived from Section 3.4. The results of the evaluation metrics of different WS's are presented in Table 4.

Table 4. Effects of different window sizes (ws) on MPO-Net (the least error results are high-lighted in bold text).

WS (n)	**Forward Passes** $(n-1)$	**ATE** **(m)**	**Trans** t_{rel} **(%)**	**Rot** r_{rel} **(deg/100 m)**
2 (no WPO)	1	98.30	12.95	4.79
3	2	84.25	9.41	3.48
4	**3**	**48.76**	**8.57**	**3.06**

This experiment provides clear evidence of increased performance while using windowed optimization. This technique also can be viewed as a resemblance to windowed bundle optimization used in state-of-the-art VO methods. It is also important to consider the computational overheads during training with a larger WS. Thus, to limit the total training time of WPO-Net, WS is limited to 4. Furthermore, the predicted trajectories of $WS = 2, 3, 4$ are illustrated in Figure 5.



(**a**) Sequence 09

(**b**) Sequence 10

Figure 5. Trajectories of sequences 09 (**a**) and 10 (**b**) under different window sizes (WS). X and Y-axis represent motion along the Z (forward) and X (left/right) axis of the vehicle in the vehicular frame.

Time taken for inference and training are measured by using a batch size of 2 averaged over hundred iterations. The inference, training time on GPU is 3.98, 19.54 and CPU is 7.87, 41.32 ms, respectively. The total parameter count of WPO-Net is 0.48 million, which makes it a light and affordable network to run on embedded controllers. Comparison of run-time analysis of WPO-Net with other methods is not included because the hardware used is different from method-to-method.

4. Conclusions

In this paper, an optimization method for learning-based VO is proposed. The proposed method can reduce overall trajectory drift and improves the accuracy of the system. From experiments, it was clear that increasing the data augmentation over a specific point degrades the performance. The proposed method outperformed most of the unsupervised methods included in comparison on the KITTI dataset. This method achieved the least rotational error than any other methods included in the comparison. The mean rotational error was improved by 13.06% compared to Reference [36], which is the best among the related works used to compare. It is certainly helpful to also note that learning-based methods included in the evaluation consist of a larger number of parameters than WPO-Net. The inference time of the proposed method on the CPU is 7.87 ms. In future work, we will validate the real-time performance of the proposed WPO-Net, along with some generalization tests.

Author Contributions: Conceptualization, methodology, software and validation, N.G. and I.E.; formal analysis and investigation, N.G. and I.E.; supervision, project administration and funding acquisition, I.E.; technical assistance and review, I.E., C.-K.L., S.P., S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by YUTP Grant (015LC0-243).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The KITTI dataset [21] used for this study is openly available at http://www.cvlibs.net/datasets/kitti/ (accessed on 15 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mazurek, P.; Hachaj, T. SLAM-OR: Simultaneous Localization, Mapping and Object Recognition Using Video Sensors Data in Open Environments from the Sparse Points Cloud. *Sensors* **2021**, *21*, 4734. [CrossRef] [PubMed]
2. Patruno, C.; Colella, R.; Nitti, M.; Reno, V.; Mosca, N.; Stella, E. A Vision-Based Odometer for Localization of Omnidirectional Indoor Robots. *Sensors* **2020**, *20*, 875. [CrossRef] [PubMed]
3. Hwang, S.J.; Park, S.J.; Kim, G.M.; Baek, J.H. Unsupervised Monocular Depth Estimation for Colonoscope System Using Feedback Network. *Sensors* **2021**, *21*, 2691. [CrossRef] [PubMed]
4. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
5. Nistér, D.; Naroditsky, O.; Bergen, J. Visual odometry for ground vehicle applications. *J. Field Robot.* **2006**, *23*, 3–20. [CrossRef]
6. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
7. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef] [PubMed]
8. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 834–849.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
11. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *Int. J. Robot. Res.* **2018**, *37*, 513–542. [CrossRef]
12. Muller, P.; Savakis, A. Flowdometry: An optical flow and deep learning based approach to visual odometry. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 624–631.
13. Mun, J.H.; Jeon, M.; Lee, B.G. Unsupervised learning for depth, ego-motion, and optical flow estimation using coupled consistency conditions. *Sensors* **2019**, *19*, 2459. [CrossRef] [PubMed]
14. Zhang, J.; Su, Q.; Liu, P.; Xu, C.; Chen, Y. Unsupervised learning of monocular depth and ego-motion with space–temporal-centroid loss. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 615–627. [CrossRef]
15. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
16. Muja, M.; Lowe, D.G. Fast matching of binary features. In Proceedings of the 2012 Ninth Conference on Computer and Robot Vision, Toronto, ON, Canada, 28–30 May 2012; pp. 404–410.
17. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
18. Geiger, A.; Ziegler, J.; Stiller, C. Stereoscan: Dense 3d reconstruction in real-time. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 963–968.
19. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625. [CrossRef] [PubMed]
20. Perdices, E.; Cañas, J.M. SDVL: Efficient and accurate semi-direct visual localization. *Sensors* **2019**, *19*, 302. [CrossRef] [PubMed]
21. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
22. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on Robot Learning. PMLR, Seoul, Korea, 13–15 November 2017; pp. 1–16.
23. Wang, W.; Zhu, D.; Wang, X.; Hu, Y.; Qiu, Y.; Wang, C.; Hu, Y.; Kapoor, A.; Scherer, S. Tartanair: A dataset to push the limits of visual slam. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 4909–4916.
24. Roberts, R.; Nguyen, H.; Krishnamurthi, N.; Balch, T. Memory-based learning for visual odometry. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 47–52.
25. Kendall, A.; Grimes, M.; Cipolla, R. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
26. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; Brox, T. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2758–2766.
27. Costante, G.; Ciarfuglia, T.A. LS-VO: Learning dense optical subspace for robust visual odometry estimation. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1735–1742. [CrossRef]
28. Zhai, G.; Liu, L.; Zhang, L.; Liu, Y.; Jiang, Y. Poseconvgru: A monocular approach for visual ego-motion estimation by learning. *Pattern Recognit.* **2020**, *102*, 107187. [CrossRef]

29. Kendall, A.; Cipolla, R. Geometric loss functions for camera pose regression with deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5974–5983.
30. Wang, X.; Zhang, H. Deep Monocular Visual Odometry for Ground Vehicle. *IEEE Access* **2020**, *8*, 175220–175229. [CrossRef]
31. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised learning of depth and ego-motion from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1851–1858.
32. Yin, Z.; Shi, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1983–1992.
33. Liu, Y.; Wang, H.; Wang, J.; Wang, X. Unsupervised monocular visual odometry based on confidence evaluation. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
34. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
35. Wang, R.; Pizer, S.M.; Frahm, J.M. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5555–5564.
36. Zhan, H.; Garg, R.; Weerasekera, C.S.; Li, K.; Agarwal, H.; Reid, I. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 340–349.
37. Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.M.; Reid, I. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 35–45.