

## Article

# Unsupervised Monocular Depth and Camera Pose Estimation with Multiple Masks and Geometric Consistency Constraints

Xudong Zhang <sup>1</sup>, Baigan Zhao <sup>2,\*</sup> , Jiannan Yao <sup>2</sup> and Guoqing Wu <sup>1</sup>

<sup>1</sup> School of Information Science and Technology, Nantong University, Nantong 226019, China; zhang.xd69@ntu.edu.cn (X.Z.); wgq@ntu.edu.cn (G.W.)

<sup>2</sup> School of Mechanical Engineering, Nantong University, Nantong 226019, China; yaojiannan2016@ntu.edu.cn

\* Correspondence: zhaobaigan@ntu.edu.cn; Tel.: +86-513-85012671

**Abstract:** This paper presents a novel unsupervised learning framework for estimating scene depth and camera pose from video sequences, fundamental to many high-level tasks such as 3D reconstruction, visual navigation, and augmented reality. Although existing unsupervised methods have achieved promising results, their performance suffers in challenging scenes such as those with dynamic objects and occluded regions. As a result, multiple mask technologies and geometric consistency constraints are adopted in this research to mitigate their negative impacts. Firstly, multiple mask technologies are used to identify numerous outliers in the scene, which are excluded from the loss computation. In addition, the identified outliers are employed as a supervised signal to train a mask estimation network. The estimated mask is then utilized to preprocess the input to the pose estimation network, mitigating the potential adverse effects of challenging scenes on pose estimation. Furthermore, we propose geometric consistency constraints to reduce the sensitivity of illumination changes, which act as additional supervised signals to train the network. Experimental results on the KITTI dataset demonstrate that our proposed strategies can effectively enhance the model's performance, outperforming other unsupervised methods.

**Keywords:** depth estimation; camera pose; visual odometry; unsupervised learning



**Citation:** Zhang, X.; Zhao, B.; Yao, J.; Wu, G. Unsupervised Monocular Depth and Camera Pose Estimation with Multiple Masks and Geometric Consistency Constraints. *Sensors* **2023**, *23*, 5329. <https://doi.org/10.3390/s23115329>

Academic Editor: Eui Chul Lee

Received: 17 April 2023

Revised: 26 May 2023

Accepted: 2 June 2023

Published: 4 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Estimating scene depth and camera pose from video sequences is a critical topic in visual perception and forms the foundation of many advanced tasks. Such estimations can be used to build 3D scene structures, which can be implemented in various industrial environments, including autonomous driving, visual navigation, and augmented reality [1–3]. Traditional methods rely on geometric cues in the image for inference, making them sensitive to challenging environments with low texture or strong lighting changes [4–8]. Conversely, learning-based depth and pose estimation methods exhibit better adaptability to challenging environments [9–13]. These methods take the image sequence as input and output the depth map and camera pose through a nonlinear mapping formed by the collaboration of neurons. This is similar to the human brain processing high-dimensional information after observing with human eyes. Supervised learning-based methods primarily use exact sensors like differential GPS, LiDAR, and IMU to obtain labeled data and then train the estimation network on these labeled data to learn the mapping function by minimizing the difference between the model's predicted values and the label values.

While supervised methods have shown excellent performance, their reliance on massive quantities of labeled data during model training poses a significant limitation. Acquiring labeled data in real-world scenarios often requires expensive equipment or large amounts of manpower, thus preventing them from further improving the performance of their models [9]. In contrast, unsupervised methods do not require expensive labels during training and can leverage larger datasets and more complex models to achieve better

performance [10]. This is exemplified by recent models such as ChatGPT, which uses large amounts of data and parameters to achieve state-of-the-art performance [14]. Researchers have also proposed numerous unsupervised methods for tasks related to depth recovery and pose estimation [10–13]. A common principle is using the view synthesis process to generate supervised signals to train the model. This technique employs two sub-networks based on convolutional neural networks (CNNs), which respectively estimate the depth map of a view (target view) and the camera pose of the view and the adjacent view (source view). Once the depth and camera pose is estimated, the source view can be projected onto the target view to synthesize a new view, and the entire framework is then trained by minimizing the photometric error between the synthesized new view and the target view.

View synthesis necessitates a static scene without occluded areas; however, real-world scenes are rife with dynamic objects and occlusions, inevitably resulting in unstable training [15]. Consequently, numerous studies have proposed various masks to mitigate outliers in the scene during the view synthesis process [15–21]. However, these methods often overlook the impact of dynamic objects and occlusion areas on pose estimation, jointly trained with depth estimation during the view synthesis process. As a result, any degradation in pose estimation accuracy will decrease depth estimation accuracy. Furthermore, training the entire unsupervised framework primarily relies on the photometric differences between the synthesized view and the target view, implying that when the illumination changes drastically or when the training sequence is lengthy, inconsistent illumination intensities may interfere with the model's learning process.

To overcome these challenges, we propose a new unsupervised learning framework for estimating scene depth and camera pose. Due to the fact that both dynamic objects and occluded regions in the scene can simultaneously affect the computation of the loss function and the estimation of the camera pose, we have designed multiple masks to identify different types of outliers in the scene during forward propagation. These computed masks are combined and applied in two ways: firstly, to prevent outliers from participating in calculating the loss function during view synthesis, and secondly, as a supervised signal for MaskNet, a neural network trained to estimate outliers such as dynamic objects and occluded regions. The mask obtained by MaskNet is then used to preprocess the input of the pose network. This involves multiplying the regions that may be outliers by a small weight coefficient, thereby avoiding the influence of outliers in the scene on pose estimation. Moreover, since training the network solely with view synthesis is prone to be affected by changes in lighting, we propose several geometrically consistent loss functions, such as flow consistency, depth consistency, and pose consistency loss, to exploit the geometric properties of the 3D scene and further strengthen our model's functionality as a whole.

The main contributions of the work are twofold: (1) We introduce multiple mask techniques to mitigate the adverse impact of outliers in the scene during the view synthesis process. Additionally, we employ a MaskNet network to address the detrimental effects of outliers on pose estimation; (2) we propose several geometric consistency constraints to alleviate the limitations of sole training with photometric consistency. Finally, we evaluate our model on the widely used KITTI dataset and demonstrate its superiority over other unsupervised methods.

The remainder of the paper is structured as follows: Section 2 provides an overview of the existing literature concerning depth and pose estimations. Section 3 elaborates on the method and the enhanced strategies proposed in this paper. The effectiveness of our approach is validated through experimental results and ablation studies, which are presented in Section 4. Finally, Section 5 offers a summary of our work.

## 2. Related Works

Visual Simultaneous Localization and Mapping (VSLAM) [4–6] and Structure from Motion (SfM) [7,8] are two examples of traditional geometry-based techniques used to estimate scene depth and camera pose. Both of these methods require extracting hand-designed key points from the image and matching them to estimate the camera pose,

followed by the triangulation technique to estimate the scene's structure. However, the traditional methods may fail in low-texture regions where key points cannot be extracted, presenting a challenge. Learning-based methods have emerged as a solution to overcome this challenge. These methods are classified as supervised or unsupervised based on whether or not they use labels for training.

### 2.1. Supervised Learning of Scene Depth and Camera Pose

Supervised depth recovery and camera motion estimation approaches typically consider these separate tasks. They learn them individually by minimizing the discrepancies between the estimated values and the related ground truth. The pioneering work by Eigen et al. [22] demonstrated the use of deep CNNs to predict depth from a single image using two network stacks: one for global prediction and the other for local refinement. In contrast, Liu et al. [23] employed hierarchical conditional random fields and a super-pixel pooling method to improve the quality of the depth map. Regarding the problem of camera pose estimation, also called visual odometry (VO), optical flow is a widely used technique in learning-based VO methods [24–26], where the optical flow (OF) field contains geometric motion. Costante et al. [25] used a self-encoder to learn the optical flow field's low-dimensional latent feature space. Then they used this feature space to regress the camera's 6-dimensional pose, thus improving the robustness of the estimated model. Zhao et al. [26] also used optical flow as the input to the pose estimation network and continued the work of [25] by adding recurrent neural networks for sequence learning to improve the estimation accuracy further. Despite the promising performance of supervised methods, their utility is limited by the requirement for labeled datasets, which can be arduous, costly, and prone to a lack of generalizability.

### 2.2. Unsupervised Learning of Scene Depth and Camera Pose

Garg et al. [11] pioneered a novel method to reduce the dependence on labeled data by training a network with stereo image pairs as input. The objective function of the training is to minimize the photometric discrepancies between the left image and its corresponding right synthetic image using epipolar geometric inference for view synthesis. Godard et al. [12] extended this methodology by incorporating the left-right consistency constraint for depth estimation. Additionally, Zhan et al. [13] continued to follow this approach of using stereo images as input, solving the absolute scale problem of VO estimation [6]. Meanwhile, they also proposed a feature reconstruction loss to strengthen the training of the framework.

Nevertheless, camera calibration in stereo systems is complex, leading Zhou et al. [15] to propose an approach relying on monocular sequences. Their core idea is to train both the depth estimation network and the pose estimation network by using the photometric consistency loss generated by the view synthesis process as the objective function. Building upon this pioneering work, subsequent studies have made significant strides. For example, in [27], Mahjourian et al. further considered the 3D geometry of the whole scene and required that the estimated 3D point cloud be consistent across the continuous images. Similarly, [28] utilized the 3D-2D correspondence constraint and deployed it on an autonomous driving platform via 5G telephony and wireless communication.

Despite its potential, the monocular unsupervised method has two significant limitations. Firstly, it cannot provide global scale consistent pose estimation, and secondly, the photometric consistency loss assumes a static scene without occlusion regions and dynamic objects. For scale ambiguity, Bian et al. [18] introduced geometric consistency loss to cope with scale inconsistencies on different samples, resulting in VO results comparable to the stereo image training model. Sun et al. [29] proposed two constraints that operate on predicted depth and relative poses, enforcing consistency across different training samples and jointly promoting pose and depth estimation. For dynamic objects and occlusion areas in the scene, an intuitive method is to design a mask to remove these outliers to avoid their adverse effects on the calculation of reconstruction loss. Therefore, researchers

have proposed a variety of mask-generation methods. Some of them generate masks by adding networks, such as the explainability mask [15] proposed by Zhou et al., the uncertain map [16] proposed by Klodt et al., and the confidence mask [17] proposed by Chen et al. Alternatively, some methods generate masks through computation, such as those in [18–21]. In [18], Bian et al. calculated depth inconsistency to generate a self-discovered mask. Zhao et al. [19] generated an outlier elimination mask by analyzing the consistency of forward and backward optical flows. Wang et al. [20] derive overlap and blank masks during forward calculation, while Jiang et al. [21] generate a mask based on the assumption that the outlier reconstruction error is significantly greater than the average photometric error. These methods have a positive impact on reconstruction loss computation.

Additionally, several studies have explored the potential benefits of jointly training the subtasks of scene depth, camera pose, and optical flow by exploiting their inherent geometric correlation, thus allowing for better nonlinear solutions when constraints are added to the loss function [30]. For instance, Yin et al. [31] proposed a collaborative learning framework to estimate all three subtasks and reconstruct a view containing both static and dynamic scenes using the geometric relationships between them. Zhang et al. [32] introduced an optical flow estimation network and added additional supervised signals to the training of the framework through multi-view synthesis to improve the overall estimation accuracy. Based on this joint estimation framework, Zou et al. [33] also proposed optical flow consistency loss for rigid regions in the scene to improve their results. Finally, Ranjan et al. [34] added a motion segmentation task to the above three tasks, and the individual performance of each task was enhanced by joint training.

Although these methods have shown considerable advancements in the basic models, their treatment of changing scenes is limited to calculating the loss function in view synthesis, ignoring the impact on VO estimation. Moreover, the detrimental effects on training when the photometry is inconsistent are not considered.

### 3. Methods

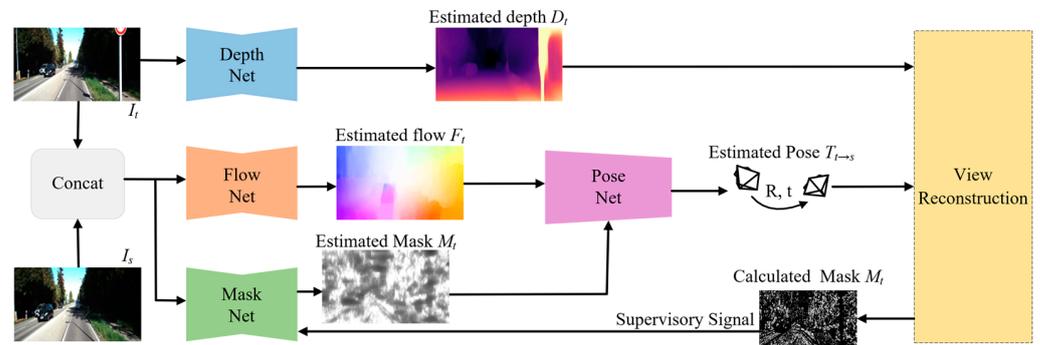
This paper aims to learn the depth and camera pose from unlabeled monocular video sequences while exhibiting good robustness to outliers in the scene, such as dynamic objects and occlusion areas. Our framework utilizes training samples composed of three consecutive frames, with the middle frame as the target view and the other two as the source view, to obtain two target-source image pairs. Since the operations performed on the two image pairs are the same, we only show the process for one image pair, as shown in Figure 1. Our method comprises four sub-networks: (1) DepthNet, which estimates the depth map of a single image; (2) FlowNet, which estimates the optical flow between adjacent frames, is an off-the-shelf model that does not require training; (3) MaskNet, which estimates a mask containing outliers based on two adjacent frames, and its supervised signal mainly comes from the calculation of multiple masks in the view synthesis process; and (4) PoseNet, which uses the mask-preprocessed optical flow as input and outputs the camera motion.

We train the DepthNet, MaskNet and PoseNet sub-networks jointly with a final loss function comprising four parts: a photometric consistency loss, which is the primary supervised signal used during network training; a depth smoothness loss, which ensures the smoothness of the estimated depth values; a mask loss, which enhances the model's performance by training a mask to preprocess PoseNet's input; and a geometric consistency loss, which provides an additional weak supervised signal by adding several constraints to the model. Therefore, the final loss function can be expressed as

$$L = \sum_l (L_{ph}^l + \lambda_s L_{sm}^l + \lambda_m L_m^l + L_g^l), \quad (1)$$

with  $L_{ph}^l$ ,  $L_{sm}^l$ ,  $L_m^l$  and  $L_g^l$  denoting the photometric consistency loss, smoothness loss, mask loss, and geometric consistency loss, respectively,  $\lambda_s$  and  $\lambda_m$  representing the corresponding weight values. The parameter  $l$  is the scale factor of different image sizes, and similar to

previous work [15], the DepthNet outputs four depth estimation maps at different scales, with the loss function being calculated for each scale separately.



**Figure 1.** Overview of our method. The input contains two adjacent images. The DepthNet estimates the depth map of the target view ( $I_t$ ). The FlowNet estimates the optical flow with the target view ( $I_t$ ) and the source view ( $I_s$ ). The MaskNet also uses the target-source image pairs as input to estimate the outliers in the scene. The estimated mask is used to preprocess the optical flow, and then the preprocessed optical flow is used as the input of the PoseNet. The estimated depth ( $D_t$ ), camera pose ( $T_{t \rightarrow s}$ ), and the source view ( $I_s$ ) are used to synthesize a new target view ( $\hat{I}_t$ ). Then the DepthNet and the PoseNet are trained by minimizing the difference between the synthesized view ( $\hat{I}_t$ ) and the target view ( $I_t$ ) as the main supervised signal. Moreover, the mask obtained by calculation in the process of forward propagation is also used as the supervised signal for the MaskNet.

### 3.1. Photometric Consistency Loss and Smoothness Loss

The primary supervised signal for training the entire framework is derived from the photometric consistency loss generated during view synthesis. The process is illustrated in Figure 2. Given two consecutive input frames ( $I_t, I_{t+1}$ ), represented as the target view and the source view, The DepthNet estimates the depth map of the target view. Using this depth information ( $D_t$ ), a pixel in the image can be projected onto a 3D point cloud as follows:

$$Q_t^{i,j} = D_t^{i,j} K^{-1} [i, j, 1]^T, \quad (2)$$

where  $K$  is the intrinsic camera matrix and  $Q_t^{i,j}$  is the 3D point. The camera motion  $T_t$  estimated by the PoseNet is used to transform  $Q_t^{i,j}$  to the coordinates at frame  $t + 1$  via  $Q_{t+1}^{i,j} = T_t Q_t^{i,j}$ . This 3D point can then be transformed to the camera coordinates at frame  $t + 1$  using the intrinsic camera matrix  $K$ . Therefore, by using these projection transformation relations, we can get the projection relationship between the coordinates of the target view and the source view, which is expressed as:

$$[\hat{i}, \hat{j}, 1]^T = K T_t \left( D_t^{i,j} K^{-1} [i, j, 1]^T \right). \quad (3)$$

To reconstruct the target view  $I_t$ , we need to obtain the corresponding pixel values of the reconstructed frame  $\hat{I}_t$  based on the projected position on frame  $I_{t+1}$ . However, the pixel values after projection are not integers. Hence we use bilinear interpolation to obtain the pixel value at  $p_{t+1}$ . Specifically, we linearly interpolate the 4-pixel values (top-left, top-right, bottom-left, and bottom-right) around  $p_{t+1}$  using the formula:

$$\hat{I}_t(p_t) = \sum_{i \in \{l,b\}, j \in \{l,r\}} w^{ij} I_{t+1} \left( p_{t+1}^{ij} \right), \quad (4)$$

where  $w^{ij}$  is the proportional term for bilinear interpolation, measuring the spatial proximity of  $p_{t+1}$  and  $p_{t+1}^{ij}$  with  $\sum_{i,j} w^{ij} = 1$ . Then, the synthesized view  $\hat{I}_t$  is obtained.

Assuming that the photometric values of 3D spatial points projected onto the target view and the source view are equal, it can be theoretically deduced that the photometric

values of the target view and the synthesized target view obtained through interpolation using the source view should be consistent. This property is used to construct the loss function for training the entire framework. Similar to other works [20,21], we use the combination of the L1 norm and the structural similarity index measure (SSIM) [35] to construct the photometric consistency loss, which is expressed as:

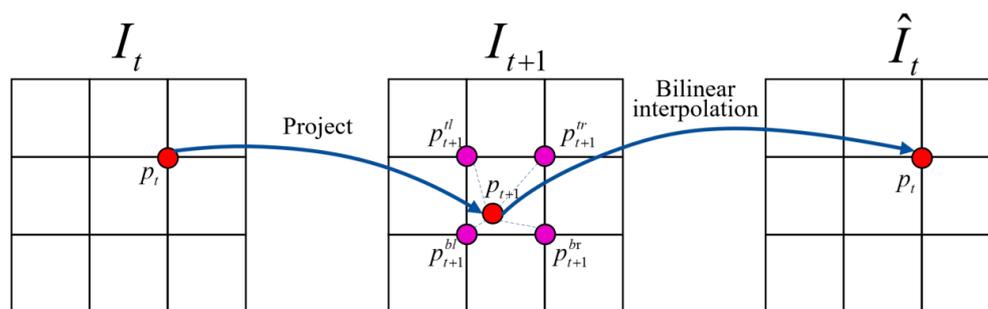
$$L_{ph}(I_t, \hat{I}_t) = \alpha \frac{1 - SSIM(I_t, \hat{I}_t)}{2} + (1 - \alpha) \|I_t - \hat{I}_t\|_1, \tag{5}$$

where  $\alpha$  is set to 0.85 empirically.

During the training of the depth estimation model, the edge smoothness loss is often used to filter out incorrect predictions and preserve clear details. We use the same loss function as [31], expressed as

$$L_{sm} = \sum_{p_t} |\nabla D(p_t)| \left( e^{-|\nabla I(p_t)|} \right)^T \tag{6}$$

where  $|\cdot|$  denotes the elementwise absolute value,  $\nabla$  represents the vector differential operator, and  $T$  denotes the transpose of image gradient weighting.



**Figure 2.** Illustration of the view synthesis process. The projection process involves using the estimated depth and camera pose to calculate the position of each point  $p_t$  in the target view  $I_t$  projected onto the source view  $I_{t+1}$ . The bilinear interpolation means that the pixel value of each point on the synthesized view  $\hat{I}_t$  is calculated by using bilinear interpolation on the source view  $I_{t+1}$ .

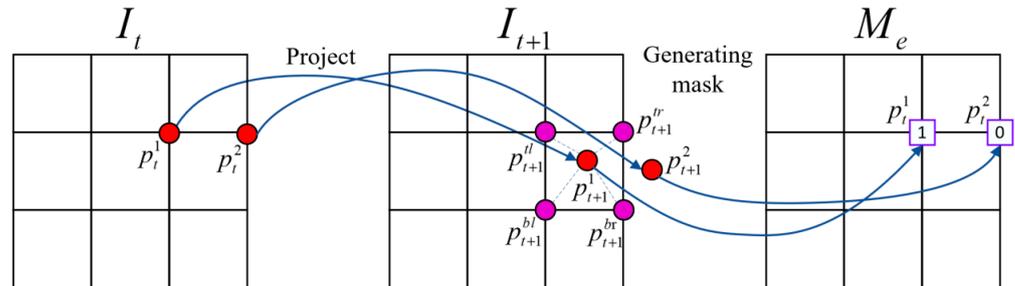
### 3.2. Calculated Mask and Mask Loss

Due to the many assumptions involved in view synthesis, such as the scene is static and devoid of dynamic objects and occlusions, training images that violate these assumptions will inevitably hinder model training. Therefore, it is necessary to consider the influence of these factors when using the view synthesis process to calculate the loss function. A common principle is to design masks to shield these outlier regions and prevent them from participating in the calculation of the loss function, thereby improving the performance of the model.

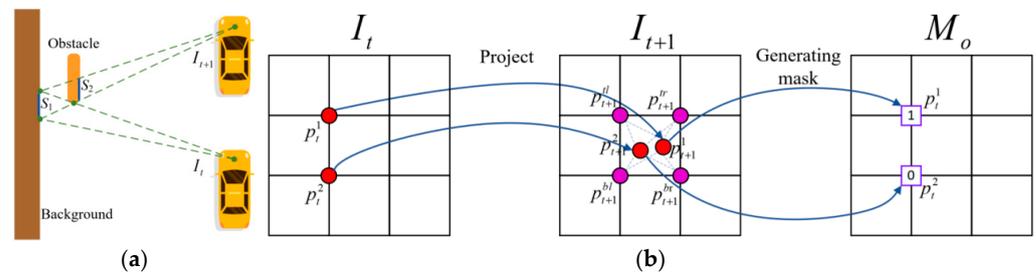
During the view synthesis process, when the pixels in the target view are projected onto the source view, some pixels will be projected beyond the imaging plane of the source view so that the pixel value of the point cannot be reconstructed. As shown in Figure 3, suppose there are two points ( $p_t^1, p_t^2$ ) in the target view  $I_t$ . When the camera moves to the frame  $I_{t+1}$ , the projection of  $p_t^1$  onto source view  $I_{t+1}$  is  $p_{t+1}^1$ , and the projection of  $p_t^2$  is  $p_{t+1}^2$ , which is located outside the boundary of  $I_{t+1}$ , making it impossible to determine the pixel value of that point accurately. Therefore, we mark all these points projected outside the boundary as 0, thus generating a boundary mask denoted as  $M_e$ . After being multiplied by the target view, the mask can avoid calculating the loss function for some boundary points.

In addition to the boundary points that will affect the training of the model, the occluded areas in the scene will also affect the training of the model. For example, as shown in Figure 4a, when a car equipped with a camera travels from frame  $I_t$  to frame  $I_{t+1}$ , there is an obstacle on the left side of the yellow car. In frame  $I_t$ , the camera can capture the areas

of  $S_1$  and  $S_2$ , but in frame  $I_{t+1}$ , the area of  $S_1$  will be occluded by the area of  $S_2$ , and only the area of  $S_2$  can be seen. Thus, the problem arises that the areas of  $S_1$  and  $S_2$  captured by frame  $I_t$  is projected onto frame  $I_{t+1}$  and overlap in coordinates, making it impossible to use the image captured at frame  $I_{t+1}$  to restore the area of  $S_1$  captured at frame  $I_t$ .



**Figure 3.** The process of generating the boundary mask.



**Figure 4.** (a) Effect of occluded areas on view synthesis process. (b) The process of generating the occlusion mask.

When occluded areas appear in the image, they must be marked to avoid their participation in calculating the loss function. As shown in Figure 4b, when two pixels ( $p_t^1$ ,  $p_t^2$ ) in frame  $I_t$  are projected onto frame  $I_{t+1}$ , if they fall in the same grid, that is, they have four identical interpolation points, we consider occlusion to have occurred. According to the distance between these two pixels and the camera, we mark the farther point as 0 and the closer point as 1. In Figure 4b, we assume that  $p_t^1$  is closer, so it is marked as 1, while  $p_t^2$  is marked as 0. This generates a mask denoted as  $M_o$ , which is used to identify the occlusion area.

Due to the pixel points that meet the hypothesis, the photometric errors will gradually converge to a lower value during training. In contrast, for some pixel points caused by various adverse factors, the photometric error value will always remain at a higher value [21]. Using this characteristic, those points with photometric errors much higher than the average are identified as outliers. Precisely, for each pixel on the target view, we can determine whether it is an outlier point according to the following expression:

$$M_a(I_t, \hat{I}_t) = \begin{cases} 1, & L_{ph}(I_t, \hat{I}_t) \leq \beta \bar{L}_{ph}(I_t, \hat{I}_t) \\ 0, & L_{ph}(I_t, \hat{I}_t) > \beta \bar{L}_{ph}(I_t, \hat{I}_t) \end{cases} \quad (7)$$

where  $\bar{L}_{ph}$  is the average value of all photometric errors,  $M_a$  indicates mask,  $\beta$  is the corresponding weight that reflects the tolerance for outliers. The larger the value, the more outliers will be retained. 1.5 is used here empirically.

The three masks mentioned above can effectively remove the majority of outliers. However, there are still some special cases that need to be considered, such as objects moving at speed similar to the camera or scenes where the camera is stationary, both of which violate the assumptions required for the view synthesis process, i.e., a moving camera and a stationary scene. To address this issue, we adopt the auto-making technique

used in [12], i.e., the photometric error of the synthesized target view should be less than the photometric error calculated directly using the source view. The mask denoted as  $M_s$  is expressed as follows:

$$M_s(I_t, \hat{I}_t) = \begin{cases} 1, & L_{ph}(I_t, \hat{I}_t) < L_{ph}(I_t, I_s) \\ 0, & L_{ph}(I_t, \hat{I}_t) \geq L_{ph}(I_t, I_s) \end{cases} \quad (8)$$

The minimum projection technique proposed in [12] addresses the problem of occluded areas in the scene, which in essence, is also a mask, and we denote this mask by  $M_m$  and calculate it by the following expression:

$$M_m(I_t, \hat{I}_t) = \begin{cases} 1, & L_{ph}(I_t, \hat{I}_t) \leq \min_s L_{ph}(I_t, I_s) \\ 0, & L_{ph}(I_t, \hat{I}_t) > \min_s L_{ph}(I_t, I_s) \end{cases} \quad (9)$$

Then, all the masks are combined using element-wise logical conjunction, as shown below, to generate the final mask, marked  $M_f$ .

$$M_f(I_t, \hat{I}_t) = M_e(I_t, \hat{I}_t) \cdot M_o(I_t, \hat{I}_t) \cdot M_a(I_t, \hat{I}_t) \cdot M_s(I_t, \hat{I}_t) \cdot M_m(I_t, \hat{I}_t), \quad (10)$$

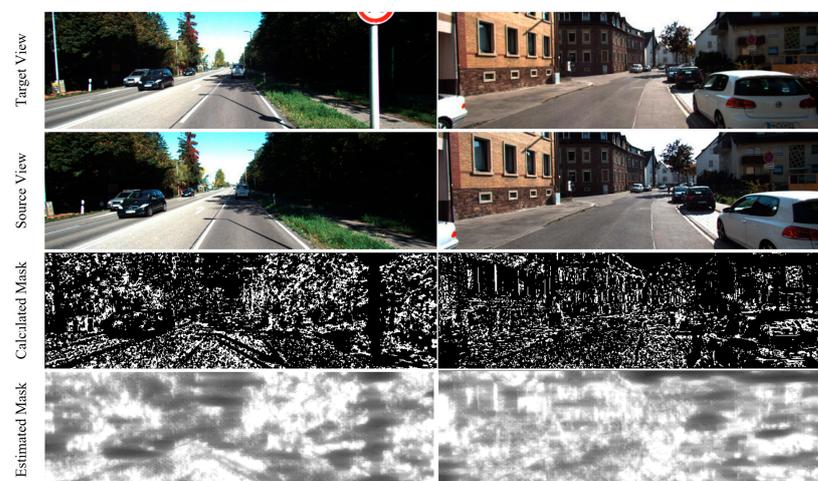
Finally, the photometric consistency loss is updated as follows:

$$L_{ph}(I_t, \hat{I}_t) = M_f(I_t, \hat{I}_t) \cdot L_{ph}(I_t, \hat{I}_t). \quad (11)$$

The final mask obtained through forward propagation also serves as a supervised signal for the MaskNet. The MaskNet takes the target-source image pairs as input and generates a mask of estimated outliers. Unlike the final mask, the values in the generated mask are not binary but continuous, ranging from 0 to 1, making it more convenient for model training. The MaskNet is trained by minimizing the difference between the estimated mask ( $M_e$ ) and the calculated mask ( $M_f$ ). The loss function is as follows:

$$L_m(M_f, M_e) = -\frac{1}{n} \sum_i \left[ M_{f_i} \log M_{e_i} + (1 - M_{e_i}) \log (1 - M_{f_i}) \right] \quad (12)$$

where  $n$  represents the number of elements in  $M_f$  and  $M_e$ , and  $\log$  is the logarithm function. Figure 5 shows two examples of the visualization of the calculated mask ( $M_f$ ) and the estimated mask ( $M_e$ ).



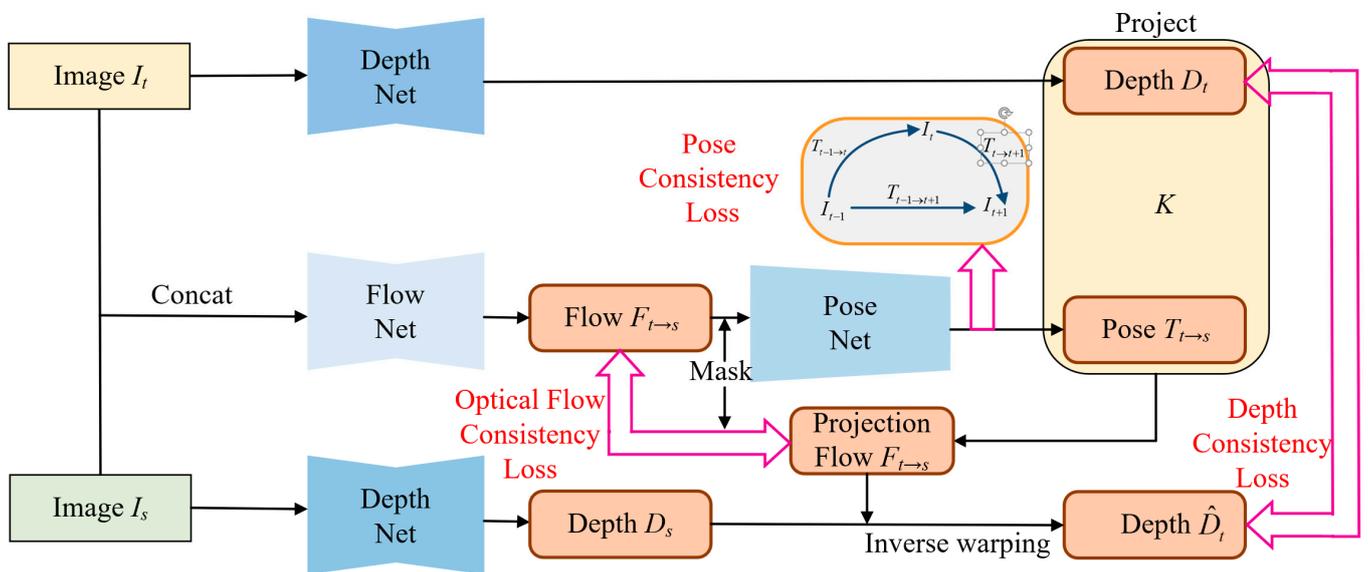
**Figure 5.** Two examples of our proposed mask visualization. The goal of the calculated and estimated masks is to mitigate the negative impact of challenging scenes on the view synthesis process and VO estimation, respectively.

### 3.3. Geometric Consistency Loss

To address the issue that the photometric consistency loss function fails to hold in scenes with significant illumination variations, we propose the geometric consistency loss with its generation mechanism illustrated in Figure 6. This loss leverages the geometric constraints inherent to the 3D scene, rendering it impervious to illumination variations and serving as a complementary measure to the photometric consistency loss. The geometric consistency loss comprises optical flow consistency loss, depth consistency loss, and pose consistency loss, which is expressed as follows:

$$L_g = \lambda_f L_{flo} + \lambda_d L_{dep} + \lambda_p L_{pos} \quad (13)$$

with  $L_{flo}$ ,  $L_{dep}$  and  $L_{pos}$  denoting the optical flow consistency loss, the depth consistency loss, and the pose consistency loss, respectively, and  $\lambda_f$ ,  $\lambda_d$  and  $\lambda_p$  are the corresponding weighting coefficients.



**Figure 6.** The generation mechanism of the geometric consistency loss function. The geometric consistency loss consists of three components: optical flow consistency loss, depth consistency loss, and pose consistency loss. The optical flow consistency loss is calculated from the difference between the estimated optical flow and the calculated projected optical flow; the depth consistency loss is calculated from the difference between the estimated depth and the inverse warping depth [18]; and the pose consistency loss is obtained in three frames of snippets by ensuring a tight coupling of the transformation matrices with each other.

To better extract the geometric features in the image when estimating the camera pose, we use the optical flow estimated by FlowNet as an input. In addition, the estimated optical flow ( $F^f$ ) can provide an additional supervised signal to the framework. Specifically, using the estimated depth and the estimated camera pose, we can calculate the projected optical flow ( $F^{cal}$ ) using the following expression:

$$F^{cal}(p_t) = K T_{t \rightarrow s} D_t(p_t) K^{-1} p_t - p_t. \quad (14)$$

After removing outliers, the calculated optical flow ( $F^{cal}$ ) should be theoretically consistent with the estimated optical flow ( $F^f$ ). Then the optical flow consistency loss can be calculated using the following formula:

$$L_{flo} = \sum_{p_t \in V} \|F^{cal}(p_t) - F^f(p_t)\|_1 \quad (15)$$

where  $V$  denotes the valid region after excluding the outliers.

Since the correspondence between the target view and source view coordinates can be determined computationally during the projection process, there is also a correspondence between the target view depth map and the source view depth map estimated by DepthNet. That is, the source view depth map and the projected optical flow can inverse warp the target view depth map, which is consistent with the depth map estimated by DepthNet [18]. Therefore, we also use the L1 norm to calculate their differences, and the depth consistency loss is represented as follows:

$$L_{dep} = \sum_{p_t \in V} \|D_t(p_t) - \bar{D}_t(p_t)\|_1. \quad (16)$$

The pose consistency loss is obtained in the three-frame snippet by ensuring that the transformation matrices are closely coupled. Specifically, with the PoseNet network, the pose information between each pair of the three image frames can be estimated, denoted as  $T_{t-1 \rightarrow t}$ ,  $T_{t \rightarrow t+1}$  and  $T_{t-1 \rightarrow t+1}$ . Using the transformation relationship between them, i.e.,  $T_{t-1 \rightarrow t} \cdot T_{t \rightarrow t+1} = T_{t-1 \rightarrow t+1}$ , the pose consistency loss is proposed to constrain the entire model further, expressed as follows:

$$L_{pos} = T_{t-1 \rightarrow t} \cdot T_{t \rightarrow t+1} - T_{t-1 \rightarrow t+1}. \quad (17)$$

#### 4. Experiments

In this section, we conduct several experiments to evaluate the estimation results of our pose and depth models and visualize the estimated results. We also conduct ablation experiments to validate the effectiveness of our used strategies as well as tests on unfamiliar datasets to verify the generalization ability of the model.

##### 4.1. Implementation Details

Our framework comprises four sub-networks. For the depth estimation network (DepthNet), we employ a U-shaped architecture [36] with skip connections, which takes a single image as input and outputs the corresponding depth map. The encoder mainly consists of cascaded residual convolutional neural networks, with ResNet18 [37] as the underlying network that contains 11 million trainable parameters. The decoder primarily consists of cascaded deconvolutional layers. For the pose estimation network (PoseNet), we also use ResNet18 as the encoder, followed by a global average layer before the final prediction to obtain the 6 DOF camera pose (3 for translation and 3 for rotation in Euler angles), with the pre-processed optical flow as input. The mask estimation network (MaskNet) has the same network structure as DepthNet but with different inputs, taking a stack of two frames in RGB channels and outputting the probability of each pixel being an outlier in the scene. Finally, for optical flow estimation, we adopt a pre-trained network, MaskFlowNet [38], to accelerate the training process by computing the optical flow information for all adjacent and intermediate frames in advance, which can be directly read during training.

The framework was implemented on the PyTorch platform [39], and all experiments were performed using a single NVIDIA graphics card (RTX 1080 Ti) and an Intel Core i7 3.6 GHz CPU. We use Adam [40] for optimization. The hyperparameters of the loss function, including  $\lambda_s = 0.001$ ,  $\lambda_m = 0.2$ ,  $\lambda_f = 0.2$ ,  $\lambda_d = 0.2$  and  $\lambda_p = 0.5$ . To ensure fair comparisons with other works, we cropped the image resolution to  $640 \times 192$ , accelerating the network training. Similar to other works [20,21], we applied data augmentation techniques such as cropping, random scaling, and horizontal flips. For all ResNet18-based models, including DepthNet, PoseNet, and MaskNet, we initialized their encoder parts using the weights pre-trained in ImageNet [41], following the practice of MonoDepth2 [12]. The batch size was set to 8, the initial learning rate was set to 0.0001, and it was multiplied by 0.6 every 5 epochs. The total training time was approximately 30 h, and the network was trained for 20 epochs.

#### 4.2. Datasets and Metrics

We conducted our training and testing on the widely used KITTI benchmark [42], the most popular dataset in the field of autonomous driving. It provides 56 driving scenes at a rate of 10 frames per second with an image resolution of approximately  $1226 \times 370$ , covering various urban, residential, and highway driving scenarios. In addition, the dataset includes ground truth for camera poses and depth maps, which are derived from multiple modalities such as high-precision LiDAR, GPS, and IMU sensors. Since our method is unsupervised and only requires a sequence of consecutive frames as input, we also pre-trained our model on the Cityscapes dataset [43], which consists of video sequences of cars driving in over 50 cities and stereo data without annotations. Additionally, we validated the generalization performance of our trained model on the Make3D dataset [44], which includes single-view images and corresponding low-resolution depth maps but no monocular sequences or stereo images.

As in previous work [15], we adopted the absolute trajectory error (ATE) to evaluate pose estimation and the synthetic policy [12] to evaluate depth estimation. The following is the definition of ATE:

$$F_i = Q_i^{-1}SP_i$$

$$ATE = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\text{trans}(F_i)\|^2} \quad (18)$$

$P_i$  and  $Q_i$  represent the estimated pose and its corresponding ground truth,  $S$  denotes the similarity transformation matrix, and  $\text{trans}$  represents the translation component [45].

Depth estimation was evaluated using various metrics, including absolute relative error (*Abs. Rel*), which measures the relative error between predicted and ground truth values; square relative error (*Sq. Rel*), which squares *Abs. Rel* and accentuates the differences; root mean squared error (*RMSE*), which reflects the absolute error between predicted and ground truth values; log root mean squared error (*RMSE log*), which uses logarithmic operations to reduce the impact of outliers on *RMSE*; and prediction accuracy ( $\delta$ ), which intuitively reflects the accuracy of the predictions. The following are the definitions of these evaluation standards:

$$\text{Abs.Rel} = \frac{1}{N} \sum_{i=1}^N \frac{|D_i - D_i^*|}{D_i^*}, \quad (19)$$

$$\text{Sq.Rel} = \frac{1}{N} \sum_{i=1}^N \frac{|D_i - D_i^*|^2}{D_i^{*2}}, \quad (20)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N |D_i - D_i^*|^2}, \quad (21)$$

$$\text{RMSE}(\log) = \sqrt{\frac{1}{N} \sum_{i=1}^N |\lg D_i - \lg D_i^*|^2}, \quad (22)$$

$$\delta = \max\left(\frac{D_i}{D_i^*}, \frac{D_i^*}{D_i}\right) < T, \quad (23)$$

where  $D_i$  and  $D_i^*$  represent the estimated depth and its related ground truth. We used  $T$  values of 1.25,  $1.25^2$ , and  $1.25^3$ . Lower values for error metrics (*Abs. Rel*, *Sq.Rel*, *RMSE*, *RMSE log*) and higher values for the accuracy metric ( $\delta$ ) indicate better performance.

#### 4.3. Evaluation of Depth Estimation

In the monocular depth estimation experiments, we evaluated the performance of our depth estimation network using the widely used Eigen split of the KITTI dataset [42]. Specifically, our training set consisted of 39,810 monocular triplets, while the validation set consisted of 4424 triplets, and the testing set included 697 representative frames. As with

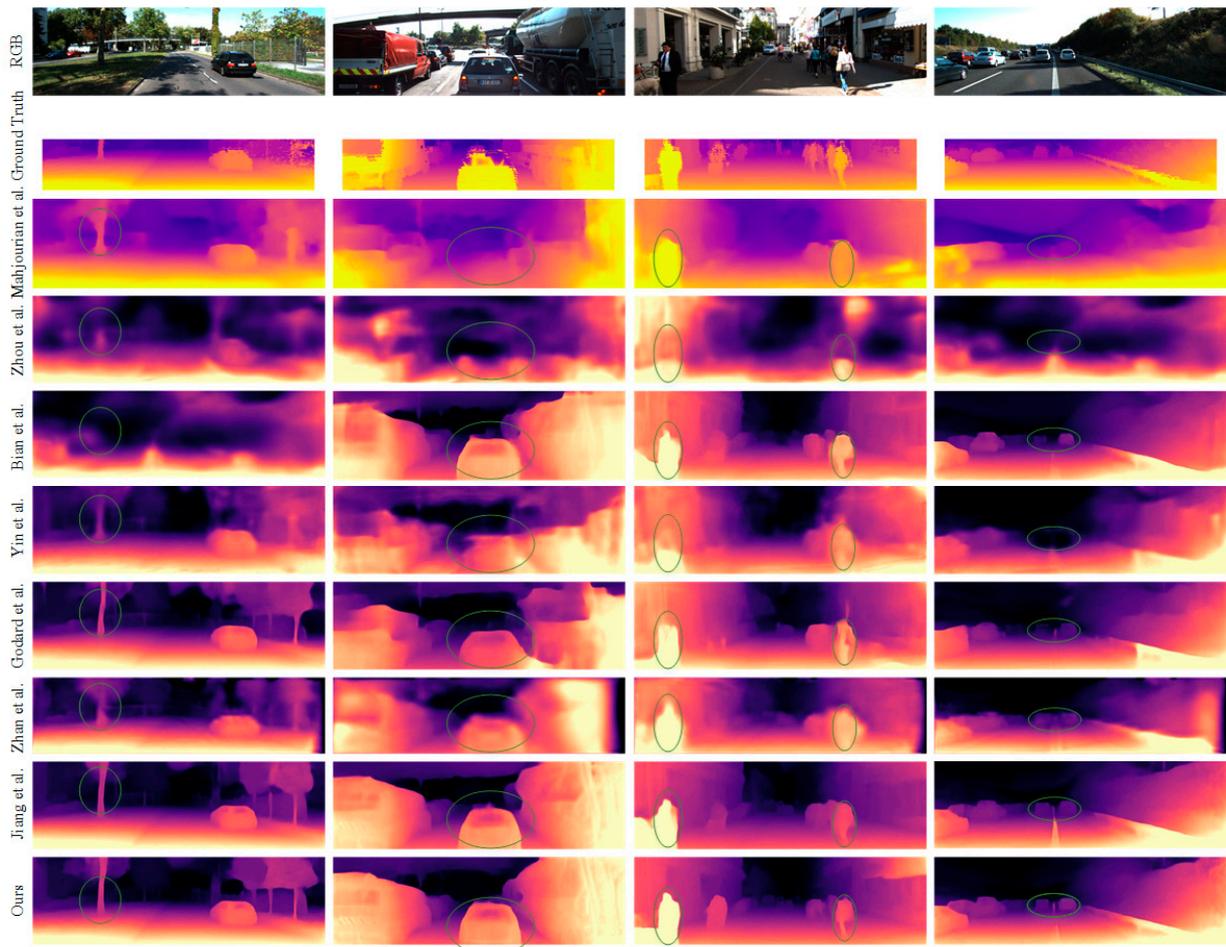
other unsupervised methods, the median ratio [15] aligns each predicted depth map with the corresponding ground truth depth map. Again, conventional metrics and the cropping region in [12] were used, and the upper limit of the standard depth was set to 80 m. Finally, we compared our method with other classic approaches using learning-based methods, and the quantitative and qualitative comparisons are shown in Table 1 and Figure 7.

**Table 1.** The quantitative comparison of our method with other methods.

Method	Supervised Signal	Training Dataset	Error Metric				Accuracy Metric		
			Abs.Rel	Sq.Rel	RMSE	RMSE (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [22] Coarse	Depth	K	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. [22] Fine	Depth	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [23]	Depth	K	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhan et al. [13]	Stereo	K	0.144	1.391	5.869	0.241	0.803	0.928	0.969
Godard et al. [12]	Stereo	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Zhou et al. [15]	Mono	K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou et al. [15] updated	Mono	K	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al. [27]	Mono	K	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Jin et al. [28]	Mono	K	0.169	1.387	6.670	0.267	0.748	0.904	0.960
Yin et al. [31]	Mono	K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Ranjan et al. [34]	Mono	K	0.140	1.070	5.326	0.217	0.826	0.941	0.975
Wang et al. [20]	Mono	K	0.147	0.889	4.290	0.214	0.808	0.942	0.979
Bian et al. [18]	Mono	K	0.137	1.089	5.439	0.217	0.830	0.942	0.975
Jiang et al. [21]	Mono	K	0.112	0.875	4.795	0.190	0.880	0.960	0.981
Godard et al. [12]	Mono	K	0.154	1.218	5.699	0.231	0.798	0.932	0.973
Ours	Mono	K	0.110	0.793	4.553	0.184	0.886	0.964	0.983
Zhou et al. [15]	Mono	CS + K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian et al. [27]	Mono	CS + K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Jin et al. [28]	Mono	CS + K	0.162	1.039	4.851	0.244	0.767	0.920	0.969
Yin et al. [31]	Mono	CS + K	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Ranjan et al. [34]	Mono	CS + K	0.139	1.032	5.199	0.213	0.827	0.943	0.977
Wang et al. [20]	Mono	CS + K	0.155	1.184	5.765	0.229	0.790	0.933	0.975
Ours	Mono	CS + K	0.103	0.634	3.367	0.178	0.899	0.969	0.984

In the supervised signal column of Table 1, “Depth” indicates that the method is supervised, “Stereo” suggests that the method is trained on stereo images with baseline information, and “Mono” suggests that the method is trained solely on monocular image sequences. The third column indicates the dataset used for training, where “K” denotes training only on the KITTI dataset and “CS + K” denotes fine-tuning on the KITTI dataset following pre-training on the Cityscapes dataset. Among all the unsupervised methods, our method outperforms all the others, especially Godard’s Monodepth2 [12], a classical depth estimation network. Our model has the same number of parameters compared to Monodepth2’s, but the model’s performance is significantly improved by using a variety of optimization strategies we have proposed. We also pre-trained our model on the Cityscapes dataset [43] and fine-tuned it on the KITTI dataset. The results (in the bottom part of Table 1) show that increasing the training data can improve the model’s performance. This demonstrates the advantage of unsupervised methods over supervised methods, where the estimation accuracy increases as the training data grows and the model size increases. However, for supervised methods, increasing the training data requires more work to generate labels, and increasing the model capacity blindly without increasing the data can lead to overfitting. This is why unsupervised methods are increasingly preferred by researchers.

A qualitative comparison of our method with some classical methods is shown in Figure 7. Compared with other methods, our method estimates the boundaries of various objects more clearly, including dynamic objects (moving cars and pedestrians) and distant cars.



**Figure 7.** Our approach is compared qualitatively with that of Mahjourian et al. [27], Zhou et al. [15], Bian et al. [18], Yin et al. [31], Godard et al. [12], Zhan et al. [13], and Jiang et al. [21].

#### 4.4. Evaluation of VO Estimation

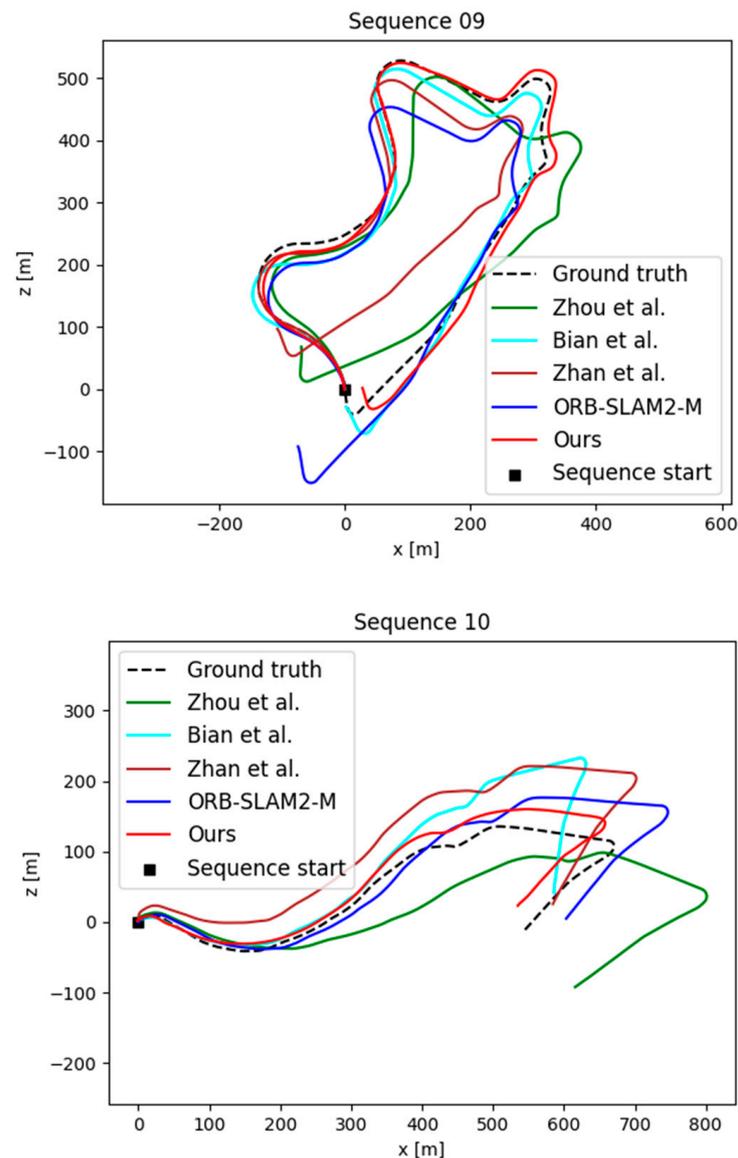
For pose estimation, we conducted experiments on the KITTI odometry dataset, using sequences 00–08 for training and sequences 09 and 10 for testing, following the method of previous work [15]. To ensure a fair comparison, the input sequences were modified from 3 to 5 frames to be consistent with other methods [11–13]. Our method’s qualitative and quantitative results compared to similar methods are shown in Figure 8 and Table 2, respectively.

**Table 2.** The performance of various methods on the KITTI VO dataset was compared in terms of Absolute Trajectory Error (ATE).

Method	ATE of Seq.09	ATE of Seq.10
ORB-SLAM [6]	$0.014 \pm 0.008$	$0.012 \pm 0.011$
Zhou et al. [15]	$0.016 \pm 0.009$	$0.013 \pm 0.009$
Bian et al. [18]	$0.016 \pm 0.007$	$0.015 \pm 0.015$
Mahjourian et al. [27]	$0.013 \pm 0.010$	$0.012 \pm 0.011$
Yin et al. [31]	$0.012 \pm 0.007$	$0.012 \pm 0.009$
Ranjan et al. [34]	$0.012 \pm 0.007$	$0.012 \pm 0.008$
Ours	$0.008 \pm 0.005$	$0.007 \pm 0.005$

Our method achieves the best results, especially over Zhou et al. [15]. On the one hand, they use raw RGB images in estimating the pose, which contains redundant information that does not help the network learn the motion information of the camera. On the other

hand, many outliers in the scene, such as occlusions and dynamic objects, impact their pose estimation model. In contrast, our method uses optical flow as input, which contains camera motion information that can be more easily learned by the network. In addition, the interference of outliers, such as dynamic objects, is effectively avoided by the multiple masking techniques, which leads to a significant improvement in the accuracy of pose estimation.



**Figure 8.** Comparison of trajectories generated by our method and the methods of Zhou et al. [15], Bian et al. [18], Zhan et al. [13], and ORB-SLAM2-M [6] on the KITTI sequences 09 and 10.

#### 4.5. Ablation Study

To verify the effectiveness of our proposed strategies, we conducted an ablation study on the entire framework. Since the DepthNet and the PoseNet adopt a joint training method and their accuracy also affects each other, it is sufficient to test only one sub-network. Here we test the PoseNet, using the sequences 00–08 on the KITTI dataset as the training set and the sequences 09 and 10 as the test set. The results of the ablation experiments are shown in Table 3.

The baseline model represents that no mask is used, and the loss function used for training only includes the photometric consistency loss and the smoothness loss.  $M_f$  denotes that the model calculated the mask used to eliminate outliers during training and

used the mask when calculating the photometric consistency loss. The model is represented by  $M_e$  not only calculates the mask  $M_f$  during training but also adds  $L_m$  to the final loss function to train the MaskNet and preprocesses the input of PoseNet with the estimated mask  $M_e$ .  $L_{flo}$  represents that only the optical flow consistency constraint is used,  $L_{dep}$  represents that only the depth consistency constraint is used,  $L_{pos}$  represents that only the pose consistency constraint is used, and  $L_{full}$  represents a complete model. That is, two masks and three consistency constraints are used.

**Table 3.** Comparison between various variants of our model on KITTI odometry.

Method	ATE of Seq.09	ATE of Seq.10
Baseline	0.018 ± 0.011	0.017 ± 0.011
Baseline + $M_f$	0.014 ± 0.010	0.013 ± 0.010
Baseline + $M_e$	0.012 ± 0.008	0.011 ± 0.008
Baseline + $M_e + L_{dep}$	0.012 ± 0.007	0.010 ± 0.007
Baseline + $M_e + L_{pos}$	0.011 ± 0.009	0.009 ± 0.006
Baseline + $M_e + L_{flo}$	0.009 ± 0.005	0.007 ± 0.005
Baseline + $M_e + L_{full}$	0.008 ± 0.005	0.007 ± 0.005

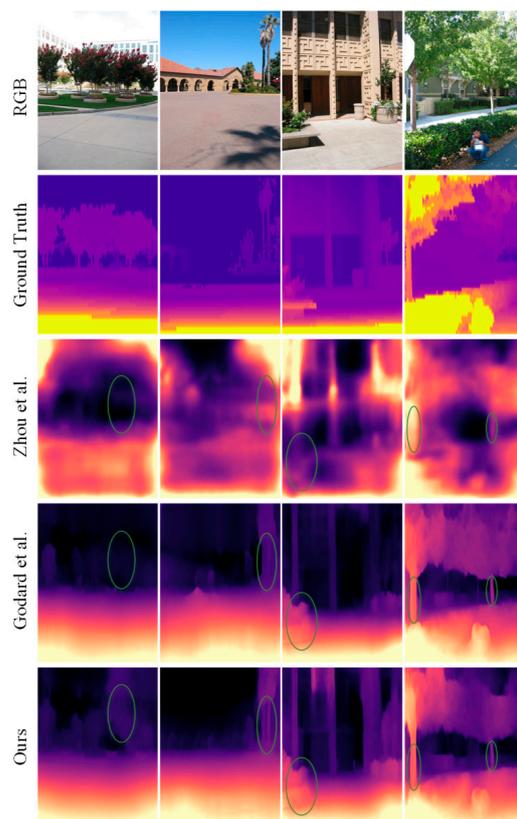
We evaluate each improved component in the proposed monocular system and remove them from the whole system to prove their effectiveness indirectly. It can be seen from Table 3 that the accuracy of the baseline is the worst of all models, and all the strategies we propose help to improve the accuracy of the framework. In addition, we found that the increase of  $L_{dep}$  and  $L_{pos}$  was relatively small, while the increase of  $L_{flo}$  was relatively large. We believe this is because the optical flow consistency constraint, the supervised signal used, comes from the trained network. This additional supervised signal has further improved the accuracy of the model. Depth consistency and pose consistency are implicit constraints within the model. Their design is mainly used to ensure global consistency in pose estimation. Since the input is a continuous multi-frame picture, the scale consistency constraint of every two frames will also ensure that the constant multi-frame pictures maintain the same scale, thus ensuring the global consistency of pose estimation.

#### 4.6. Generalization on Make3D Dataset

To verify the generalization of our trained model, we conducted tests on the Make3D [44] dataset using the model trained on the KITTI and Cityscapes datasets. This means that our model has never encountered any images from the Make3D dataset before. The qualitative and quantitative results of the depth estimation network are shown in Figure 9 and Table 4. It can be observed that our method has certain advantages over other methods of the same type. However, due to different domain biases in different datasets, there is still room for improvement in the performance of our model compared to the results obtained on the KITTI dataset.

**Table 4.** Quantitative comparison of depth estimation of different methods on Make3D dataset.

Method	Error Metric			
	Abs. Rel	Sq. Rel	RMSE	RMSE (log)
Liu et al. [23]	0.481	6.761	10.55	0.169
Zhou et al. [15]	0.396	5.731	10.869	0.513
Godard et al. [12]	0.579	11.235	11.892	0.201
Ours	0.304	3.452	7.186	0.203



**Figure 9.** A qualitative comparison of our method with that of Zhou et al. [15] and Godard et al. [12] on the Make3D dataset.

## 5. Conclusions

This paper introduces a novel unsupervised learning framework for estimating scene depth and camera pose from video sequences, focusing on challenging scenes. Our proposed method incorporates multiple mask techniques to identify and eliminate the influence of outliers in challenging scenes during the view synthesis process and pose estimation. Furthermore, we have proposed several geometrically consistent loss functions as additional supervised signals to enhance the performance of our model. We conducted evaluation and ablation experiments on the KITTI dataset, and the results validate the effectiveness of our contributions. Our framework has promising potential for addressing the challenges of estimating scene depth and camera pose in real-world scenarios. Future work can extend our method to handle more complex scenes and design the framework using high-capacity models such as Transformer and training with more data.

**Author Contributions:** Conceptualization, X.Z. and B.Z.; methodology, X.Z.; software, X.Z.; validation, J.Y., G.W. and X.Z.; formal analysis, X.Z.; investigation, X.Z.; resources, X.Z.; data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, B.Z. and J.Y.; visualization, X.Z.; supervision, J.Y.; project administration, G.W.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (Grant No. 51805273), the Qing Lan Project of Jiangsu Province, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and Industry-university-research Cooperation Project in Jiangsu Province (Grant No. BY2020542).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data are available in a publicly accessible repository.

**Acknowledgments:** The authors gratefully acknowledge the support of the National Natural Science Foundation of China (Grant nos. 51805273 and 51975569), the support of the Qing Lan Project of Jiangsu Province and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and the support of Industry-university-research Cooperation Project in Jiangsu Province, (Grant No. BY2020542).

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

## References

1. Wei, H.; Huang, Y.; Hu, F.; Zhao, B.; Guo, Z.; Zhang, R. Motion Estimation Using Region-Level Segmentation and Extended Kalman Filter for Autonomous Driving. *Remote Sens.* **2021**, *13*, 1828. [[CrossRef](#)]
2. Rosique, F.; Navarro, P.J.; Miller, L.; Salas, E. Autonomous Vehicle Dataset with Real Multi-Driver Scenes and Biometric Data. *Sensors* **2023**, *23*, 2009. [[CrossRef](#)] [[PubMed](#)]
3. Luo, G.; Xiong, G.; Huang, X.; Zhao, X.; Tong, Y.; Chen, Q.; Zhu, Z.; Lei, H.; Lin, J. Geometry Sampling-Based Adaption to DCGAN for 3D Face Generation. *Sensors* **2023**, *23*, 1937. [[CrossRef](#)] [[PubMed](#)]
4. Zou, Y.; Eldemiry, A.; Li, Y.; Chen, W. Robust RGB-D SLAM Using Point and Line Features for Low Textured Scene. *Sensors* **2020**, *20*, 4984. [[CrossRef](#)] [[PubMed](#)]
5. Romero-Ramirez, F.J.; Muñoz-Salinas, R.; Marín-Jiménez, M.J.; Cazorla, M.; Medina-Carnicer, R. sSLAM: Speeded-Up Visual SLAM Mixing Artificial Markers and Temporary Keypoints. *Sensors* **2023**, *23*, 2210. [[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. Liang, Y.; Yang, Y.; Fan, X.; Cui, T. Efficient and Accurate Hierarchical SfM Based on Adaptive Track Selection for Large-Scale Oblique Images. *Remote Sens.* **2023**, *15*, 1374. [[CrossRef](#)]
8. Liu, Z.; Qv, W.; Cai, H.; Guan, H.; Zhang, S. An Efficient and Robust Hybrid SfM Method for Large-Scale Scenes. *Remote Sens.* **2023**, *15*, 0769. [[CrossRef](#)]
9. Fang, X.; Li, Q.; Li, Q.; Ding, K.; Zhu, J. Exploiting Graph and Geodesic Distance Constraint for Deep Learning-Based Visual Odometry. *Remote Sens.* **2022**, *14*, 1854. [[CrossRef](#)]
10. Wan, Y.; Zhao, Q.; Guo, C.; Xu, C.; Fang, L. Multi-Sensor Fusion Self-Supervised Deep Odometry and Depth Estimation. *Remote Sens.* **2022**, *14*, 1228. [[CrossRef](#)]
11. Garg, R.; Bg, V.K.; Carneiro, G.; Reid, I. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 740–756.
12. Godard, C.; Aodha, O.M.; Firman, M.; Brostow, G. Digging into Self-Supervised Monocular Depth Estimation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Seoul, Republic of Korea, 2019; pp. 3827–3837.
13. Zhan, H.; Garg, R.; Weerasekera, C.S.; Li, K.; Agarwal, H.; Reid, I.M. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Salt Lake City, UT, USA, 2018; pp. 340–349.
14. Van Dis, E.A.; Bollen, J.; Zuidema, W.; van Rooij, R.; Bockting, C.L. ChatGPT: Five Priorities for Research. *Nature* **2023**, *614*, 224–226. [[CrossRef](#)]
15. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6612–6619.
16. Klodt, M.; Vedaldi, A. Supervising the New with the Old: Learning SFM from SFM. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018; pp. 713–728.
17. Chen, S.; Han, J.; Tang, M.; Dong, R.; Kan, J. Encoder-Decoder Structure with Multiscale Receptive Field Block for Unsupervised Depth Estimation from Monocular Video. *Remote Sens.* **2022**, *14*, 2906. [[CrossRef](#)]
18. Bian, J.-W.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.-M.; Reid, I. Unsupervised Scale-Consistent Depth and Ego-Motion Learning from Monocular Video. *arXiv* **2019**, arXiv:1908.10553.
19. Zhao, B.; Huang, Y.; Ci, W.; Hu, X. Unsupervised Learning of Monocular Depth and Ego-Motion with Optical Flow Features and Multiple Constraints. *Sensors* **2022**, *22*, 1383. [[CrossRef](#)] [[PubMed](#)]
20. Wang, G.; Wang, H.; Liu, Y.; Chen, W. Unsupervised Learning of Monocular Depth and Ego-Motion Using Multiple Masks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Montreal, QC, Canada, 2019; pp. 4724–4730.
21. Jiang, H.; Ding, L.; Sun, Z.; Huang, R. Unsupervised Monocular Depth Perception: Focusing on Moving Objects. *IEEE Sensors J.* **2021**, *21*, 27225–27237. [[CrossRef](#)]
22. Eigen, D.; Puhrsch, C.; Fergus, R. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 13 December 2014; pp. 2366–2374.
23. Liu, F.; Shen, C.; Lin, G.; Reid, I. Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2024–2039. [[CrossRef](#)]

24. 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; IEEE: Santa Rosa, CA, USA, 2017; pp. 624–631.
25. 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](#)]
26. Zhao, B.; Huang, Y.; Wei, H.; Hu, X. Ego-Motion Estimation Using Recurrent Convolutional Neural Networks through Optical Flow Learning. *Electronics* **2021**, *10*, 222. [[CrossRef](#)]
27. Mahjourian, R.; Wicke, M.; Angelova, A. Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; IEEE: Salt Lake City, UT, USA, 2018; pp. 5667–5675.
28. Jin, F.; Zhao, Y.; Wan, C.; Yuan, Y.; Wang, S. Unsupervised Learning of Depth from Monocular Videos Using 3D–2D Corresponding Constraints. *Remote Sens.* **2021**, *13*, 1764. [[CrossRef](#)]
29. Sun, Q.; Tang, Y.; Zhang, C.; Zhao, C.; Qian, F.; Kurths, J. Unsupervised Estimation of Monocular Depth and VO in Dynamic Environments via Hybrid Masks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 2023–2033. [[CrossRef](#)]
30. Mahdy, A.M.S. A Numerical Method for Solving the Nonlinear Equations of Emden-Fowler Models. *J. Ocean. Eng. Sci.* **2022**, 1–8. [[CrossRef](#)]
31. Yin, Z.; Shi, J. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; IEEE: Salt Lake City, UT, USA, 2018; pp. 1983–1992.
32. Zhang, J.-N.; Su, Q.-X.; Liu, P.-Y.; Ge, H.-Y.; Zhang, Z.-F. MuDeepNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose Using Multi-View Consistency Loss. *Int. J. Control. Autom. Syst.* **2019**, *17*, 2586–2596. [[CrossRef](#)]
33. Zou, Y.; Luo, Z.; Huang, J.-B. DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018; pp. 38–55.
34. Ranjan, A.; Jampani, V.; Balles, L.; Kim, K.; Sun, D.; Wulff, J.; Black, M.J. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; IEEE: Long Beach, CA, USA, 2019; pp. 12232–12241.
35. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
36. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention, MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Springer: Munich, Germany, 2015; pp. 234–241.
37. 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 (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; IEEE: Las Vegas, NV, USA, 2016; pp. 770–778.
38. Zhao, S.; Sheng, Y.; Dong, Y.; Chang, E.I.; Xu, Y. MaskFlowNet: Asymmetric Feature Matching with Learnable Occlusion Mask. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6277–6286.
39. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in Pytorch. In Proceedings of the Advances in Neural Information Processing Systems Workshop, Long Beach, CA, USA, 9 December 2017.
40. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
41. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
42. 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.
43. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 3213–3223.
44. Saxena, A.; Sun, M.; Ng, A.Y. Learning 3-d scene structure from a single still image. In Proceedings of the 2007 IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
45. Zhang, Z.; Scaramuzza, D. A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; IEEE: Madrid, Spain, 2018; pp. 7244–7251.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.