sensors

MDPI

Article

Unsupervised Learning of Depth and Camera Pose with Feature Map Warping

Ente Guo [1], Zhifeng Chen [1,*], Yanlin Zhou [2] and Dapeng Oliver Wu [2]

[1] College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China; guoente@gmail.com

[2] Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA; zhou.y@ufl.edu (Y.Z.); dpwu@ieee.org (D.O.W.)

* Correspondence: Zhifeng@ieee.org

Abstract: Estimating the depth of image and egomotion of agent are important for autonomous and robot in understanding the surrounding environment and avoiding collision. Most existing unsupervised methods estimate depth and camera egomotion by minimizing photometric error between adjacent frames. However, the photometric consistency sometimes does not meet the real situation, such as brightness change, moving objects and occlusion. To reduce the influence of brightness change, we propose a feature pyramid matching loss (FPML) which captures the trainable feature error between a current and the adjacent frames and therefore it is more robust than photometric error. In addition, we propose the occlusion-aware mask (OAM) network which can indicate occlusion according to change of masks to improve estimation accuracy of depth and camera pose. The experimental results verify that the proposed unsupervised approach is highly competitive against the state-of-the-art methods, both qualitatively and quantitatively. Specifically, our method reduces absolute relative error (Abs Rel) by 0.017–0.088.

Keywords: monocular depth estimation; single camera egomotion; occlusion-aware mask network; feature pyramid matching loss

check for updates

Citation: Guo, E.; Chen, Z.; Zhou, Y.; Wu, D.O. Unsupervised Learning of Depth and Camera Pose with Feature Map Warping. *Sensors* **2021**, *21*, 923. https://doi.org/10.3390/s21030923

Academic Editor: Sukho Lee
Received: 6 January 2021
Accepted: 26 January 2021
Published: 30 January 2021

1. Introduction

Vision-based environment depth and egomotion estimation are essential for autonomous vehicle perception and infrastructure-less robot navigation [1]. At present, LiDAR and RGB-D cameras have been widely used in the depth measurement. LiDAR has become more precise and cheaper, such as Livox mid-40, 100, but it is still not perfect, like the small field of view, irregular scanning pattern, nonrepetitive scanning and motion blur [2]. The application of RGB-D cameras in outdoor environments has also become more extensive, but the measurement range is limited [3]. Therefore, in order to deal with the complex outdoor environment, real outdoor robotic applications focus on multiple sensor fusion. In this context, the better each individual sensor is, the better the final result is [4]. The monocular is attractive because it has the advantages of low price, high resolution, rich information acquisition. More accurate monocular depth estimation is helpful for depth estimation of multiple sensor fusion. Therefore, obtaining depth based on monocular is a valuable study. Recent deep learning-based methods have shown great success on monocular depth and egomotion estimation [5,6]. These methods can be divided into two categories: supervised learning methods [5,7,8] and unsupervised learning methods [9–18]. Our work focuses on monocular unsupervised method of depth and egomotion estimation, since supervised method requires time-consuming handicraft labels.

Most unsupervised learning methods estimate depth and camera egomotion by minimizing a photometric error [10]. The photometric error is the sum of absolute differences (SAD) between the warped frame and target frame, where the warped frame is obtained from adjacent one, predicted depth and relative camera motion of the target frame [9,10].

A common assumption used by current works is photometric consistency, that is, the photometric error of corresponding pixel of the same object in different frames is zero. The photometric consistency assumption is often not satisfied because of brightness change and non-Lambertian surface [19]. To overcome these issues, GeoNet [11] added structural similarity (SSIM) [20] to loss to mitigate the effects of brightness change. SSIM captures more local information than SAD, but it does not capture global information. D3VO [19] predicted the global transformation parameters a, b through a network, and adjusts the image I to $aI + b$. However, D3VO only pays attention to the global brightness change, which is often hard to be satisfied in the real scene. None of these methods consider both local and global information.

In addition, the dynamic objects and occlusion also violate the photometric consistency. To overcome the problem of dynamic objects, the unsupervised method struct2depth [13] segmented all objects in the image and then estimated the 3D motion of each object. This method is suitable for highly dynamic scenes, but the accuracy of the depth is affected by 3D motion estimation. Furthermore, SC-SfmLearner [6] proposed a self-discovery mask for handling moving objects, which improves the accuracy of depth estimation. However, its mask definition adopts relative error, and thus is not sensitive to depth changes in areas with large depth, which causes inaccurate depth estimation. Regarding the occlusion problem, as far as we know, there is no existing unsupervised method in literature.

Our contributions are as follows.

1. We propose feature pyramid matching loss (FPML) capturing local and global information, which is more robust than SAD and SSIM and can solve the problem of photometric inconsistency caused by brightness change.

2. The proposed occlusion-aware mask (OAM) addresses, for the first time, the problem of photometric inconsistency causing by occluded pixels in the image with the consideration of novel relationship between two adjacent masks.

3. Furthermore, OAM solves the problem of dynamic objects by balancing the photometric error and the regularization term of the mask and improve the accuracy of depth and camera egomotion.

2. Related Work

The development of deep learning has facilitated the application of supervised and unsupervised methods. We briefly overview some supervised depth estimation methods and introduce current SOTA unsupervised methods for single view depth and egomotion estimation.

2.1. Supervised Depth Estimation Via Convolutional Neural Network (CNN)

The supervised learning methods establish the relationship between image and corresponding depth through CNN. Eigen et al. [7] first proposed using CNN to predict monocular image depth in 2014. They proposed a multiscale method that uses two deep network stacks: one makes a rough global prediction based on the whole image, and the other optimizes the prediction locally. Eigen et al. [8] improved the previous method by increasing the number of multiscale layers to obtain more image details. They used a single multiscale CNN architecture to accomplish three different computer vision tasks: depth prediction, surface normal estimation and semantic labeling. Li et al. [21] improved depth estimation on the basis of Eigen et al. [7] and proposed a fast-to-train multiscale CNN with skip connections between multiscale layers to speed up convergence during training. Laina et al. [22] proposed a fully convolutional network, encompassing residual learning to map monocular images to depth. They presented a novel upsampling method to improve the output resolution and introduced the reverse Huber loss to improve the accuracy of depth estimation. Xu et al. [23] proposed a deep model that fuses complementary information derived from multiple CNN side outputs. They presented two fusion methods: one is based on a cascade of multiple conditional random fields and the other is based on a unified graphical model.

The above-mentioned supervised methods need a large number of ground truths during training, but acquiring ground truths is difficult in practice. Using synthetic data is a good alternative, but these data cannot simulate the physical world accurately [24].

2.2. Unsupervised Depth and Egomotion Estimation

Compared with the supervised methods, the unsupervised learning methods do not need labels; thus, the latter methods overcome the disadvantage of the supervised learning relying on labels. Unsupervised depth and camera egomotion estimation only needs raw video sequences. These methods refines the model from the video gathered from a new scene [13]; thus, it can be rapidly deployed in practical applications.

Garg et al. [25] proposed an unsupervised depth estimation method using stereo pairs for the first time. The autoencoder network predicts the depth of the left image, and a reconstructed the left image is synthesized by epipolar geometry constraint [26] and the right image. The photometric error between the left image and the synthesized left image is used as a loss term to train the autoencoder network. Godard et al. [9] extended Garg's work and proposed the left-right depth consistent loss function to improve the accuracy of depth estimation. Stereo unsupervised learning requires stereo image pairs and the known pose between stereo cameras during training.

SfmLearner [10] only used the monocular video sequence while learning the monocular depth and egomotion in a coupled way. They used depth network to predict monocular depth and pose network to predict the relative camera pose between consecutive frames. The color inconsistency between target image and synthesized target images, which warped from the reference image, was used as the supervision signal. SfmLearner proposed an explainability mask to alleviate the influence of moving objects and non-Lambertian surfaces for making the system more robust. SFM-Net [12] outputted k motion objects' mask and their rigid motion through the motion network to overcome the influence of moving objects. However, it is limited by the maximum number of moving objects. In contrast to SFM-Net, Yin et al. [11] decomposed motion into rigid and nonrigid components and introduced a residual flow learning module to deal with nonrigid scenes. Casser et al. [13] segmented all possible moving objects by Mask R-CNN [27] before training and then estimated the 3D motion of each object to overcome the weakness of SFM-Net. However, masking all possible moving objects prevents the network from learning the depth object and Mask RCNN increases the amount of calculation. SC-SfMLearner [6] proposed a self-discovery mask for dynamic scene in consideration of geometric consistency constraints, which improves the accuracy of depth estimation. However, it has room for improvement in the area of large depth, because the relative error decreases with the increase of depth in the case of the same absolute error of depth. We propose OAM, which can not only address the problem of occluded pixels but also reduce the depth blur caused by moving objects.

Most of these methods are based on photometric errors and assume constant brightness and Lambertian surface of objects. However, meeting these conditions is difficult in real scenes. To handle the problem, [9,11,13,28,29] added SSIM [20] as a loss term to produce more robust matching and improve the performance of depth prediction. Unsupervised optical flow [30] also used the photometric error as loss function. They adopted robust kernel functions to deal with cases in which photometric consistency assumptions are not met. In contrast to hand-craft feature, we propose a FPML that is inspired by PWC-Net cost volume [31]. Instead of matching hand-craft features, a trainable feature pyramid is constructed by CNN.

3. Method

3.1. Preliminaries

Our method uses single-view depth and multiview pose networks, with a loss based on warping the adjacent frames to the current frame using the computed depth and pose. In this work, we propose a framework containing three networks: a depth prediction network (DepthNet), a camera egomotion network (MotionNet) and an occlusion-aware mask network (MaskNet). The networks will be trained together due to the loss function but can be applied independently at test time. The framework of the networks and loss functions are shown in Figure 1, in which the blue arrows represent the input and output of the networks. DepthNet input is a frame, which can predict the corresponding depth. The information of multiple frames is enough to estimate the camera egomotion [26], so the input of MotionNet is the current frame I_t and the adjacent frames I_f. The output of MotionNet is the camera egomotion $T_{t \to f}$, including rotation Euler angle and 3D position, where the adjacent frames include the past and next frames, $I_f \in \{I_{t-1}, I_{t+1}\}$. In order to predict occluded pixels and moving objects, the input of MaskNet is the current frame and the adjacent frames, and the output is consistent mask M_f and occlusion mask V_f. The masks outputted by the MaskNet are only used in the training stage. It can exclude pixels that do not conform to the static scene and are occluded, ensuring that DepthNet and MotionNet can learn the correct depth and camera egomotion respectively. In the training phase, DepthNet, MaskNet and MotionNet are trained at the same time. However, in the testing phase, MaskNet is not needed, so it can be called an auxiliary network for auxiliary training. The details of the networks are described in Section 3.5.

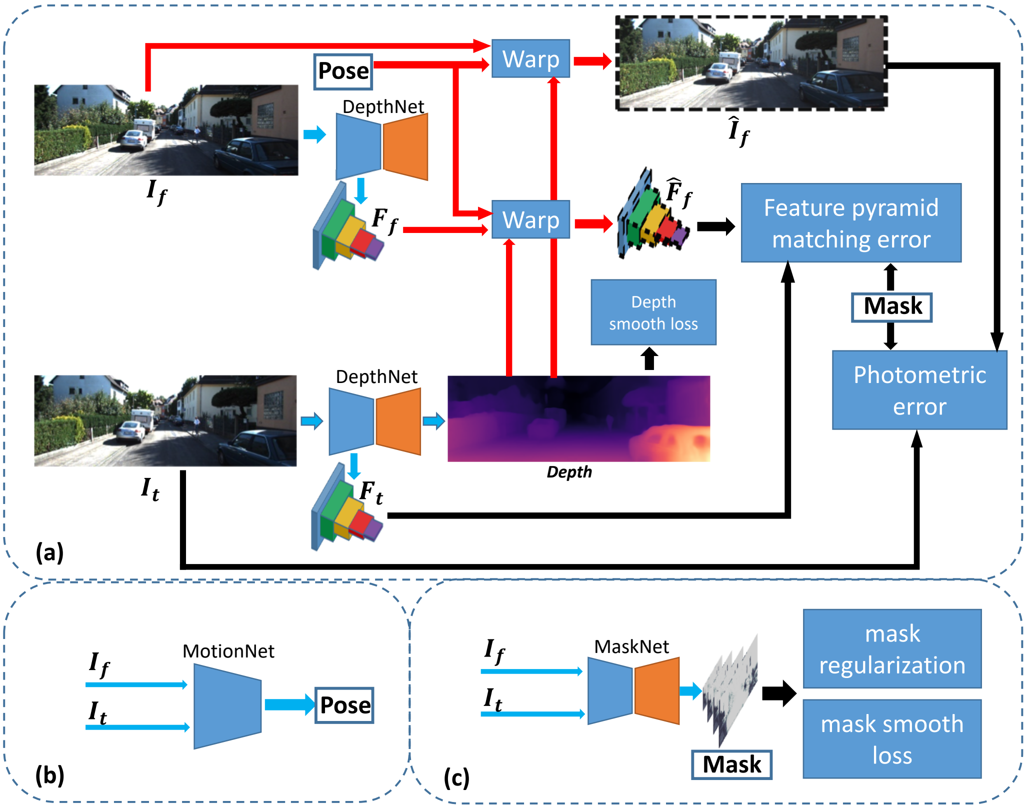

Figure 1. System architecture. (**a**) DepthNet, loss function and warping; (**b**) MotionNet (**c**) MaskNet. It consists of the DepthNet for predicting depth map of the current frame I_t, the MotionNet for estimating egomotion from current frame I_t to adjacent frame I_f, and the MaskNet for generating occlusion-aware mask (OAM). The reconstructed current frame $\hat{I}_f$ and reconstructed current feature pyramid $\hat{F}_f$ are synthesized by warping. The total loss function consists of photometric error, depth smooth loss, mask regularization term, mask smooth loss and feature pyramid matching loss (FPML).

The warp process is to find the corresponding point in the adjacent frames through the depth map of the current frame and the camera egomotion, and then synthesize the current frame. The warping process is divided into two steps: coordinate transformation and interpolation reconstruction. According to the pinhole camera model, $P = D_t(p_t)K^{-1}p_t$ is a back projection process [26], where P represents a point in 3D space, p_t denotes the homogeneous coordinate of the point on the current frame, K is the given camera intrinsic parameters, and $D_t(p_t)$ is the depth of p_t. The projection p_f of P in the adjacent frames is inferred as follows,

$$\begin{aligned} D_f(p_f)p_f &= KT_{t\to f}\begin{bmatrix} P \\ 1 \end{bmatrix} \\ &= K(D_t(p_t)T_{t\to f}\begin{bmatrix} K^{-1}p_t \\ 1 \end{bmatrix}). \end{aligned} \tag{1}$$

The process of interpolation reconstruction is to synthesize the pixel value of p_t according to the adjacent frames, $\hat{I}_f(p_t) = I_f(p_f)$, where $\hat{I}_f$ represents the current frame synthesized by I_f. We use the differentiable bilinear interpolation proposed by the spatial transformer network [32] to obtain $I_f(p_f) = \sum_{i,j}\omega_{i,j}I_f(p_f^{i,j})$, where $p_f^{i,j}$ is the integer pixel located at the neighborhood (top left, top right, bottom left, and bottom right) of p_f, and $\sum_{i,j}\omega_{i,j} = 1$. As shown in Figure 1, the red arrows in the framework are the input and output of the warp module. The warp process of the feature map is similar to the warp of the RGB image, except that the multichannel feature map replaces the three-channel color.

The loss we propose includes a photometric error L_p weighted by the OAM, a depth smoothness loss L_s, a mask regularization loss L_m, a mask smoothness loss L_{ms} and the FPML L_f. we define overall loss function as follows,

$$L_{all} = \sum_{n=0}^{3}(L_p^n + \lambda_s L_s^n + \lambda_m L_m^n + \lambda_{ms} L_{ms}^n + \lambda_f L_f^n), \tag{2}$$

where λ_s, λ_m, λ_{ms}, λ_f are the weight of depth smoothness loss, weight of mask regularization term, weight of mask smoothness loss and weight of feature pyramid matching loss respectively. The settings for them are described in Section 4.1. The total loss is applied on four scales to combat the problem of holes caused by gradient locality [10], and n indexes are considered over different depth map scales. The photometric error, the OAM and the FPML elaborated in Sections 3.2–3.4 respectively.

3.2. Photometric Error and Smooth Loss

Under the assumption of surface Lambertian and static rigid scenes, the brightness of the same object under different views should be consistent. Therefore, the current frame $\hat{I}_f$ synthesized by the depth, camera egomotion and adjacent frame images should be similar to the current frame I_t. We construct a robust photometric error loss function as follows,

$$L_p = \sum_{f\in\{t-1,t+1\}} V_f M_f \delta(I_t, \hat{I}_f), \tag{3}$$

where $\delta(I_t, I_f)$ represents the difference between the current frame and the reconstructed frame, $\delta(I_t, I_f) = \alpha\frac{1-SSIM(I_t,I_f)}{2} + (1-\alpha) \parallel I_t - I_f \parallel_1$; SSIM is structural similarity index [20]; M_f and V_f are the consistent mask and occlusion mask respectively, which are defined in Section 3.3.

In order to make the depth smooth and the edge of it sharp, we also use the following image gradient [9] based depth smoothness loss function,

$$L_s = |\bigtriangledown_x D_t|e^{-|\bigtriangledown_x I_t|} + |\bigtriangledown_y D_t|e^{-|\bigtriangledown_y I_t|}, \tag{4}$$

where $\bigtriangledown_x$ and $\bigtriangledown_y$ represent the gradients in X and Y directions, respectively.

3.3. Occlusion-Aware Mask

Photometric consistency assumes that the scene is static and the objects are nonoccluded. However, dynamic objects and occlusion usually occur in real scenes. As shown in Figure 2, the pixels in the yellow dash area are visible in the past frame I_{t-1} and current frame I_t but blocked by the vehicle in the next frame I_{t+1}. If the network predicts the correct depth of the pixels in the yellow dashed area in current frame, then the corresponding occluded area in the next frame does not match the the current frame. This condition results in the large photometric error. The average photometric error is affected by occlusion. Occlusion often occurs at the edge of the object and the inferred incorrect depth. Thus, we propose a multiframe formulation to train a network for predicting occlusions.

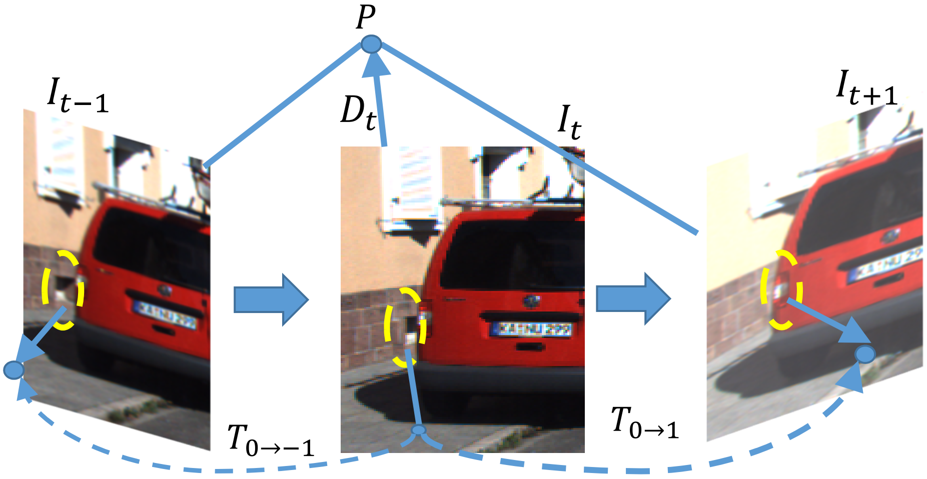

Figure 2. Example of occlusion.

We assume a object is visible in the current frame. Depending on whether the corresponding pixel on adjacent frames is visible, there are four cases of the corresponding pixel as follows: visible in all adjacent frames, occluded in all adjacent frames, occluded in the past or occluded in the future. The case that a pixel occluded in all adjacent frames rarely occurs in practice is discarded.

The input of MaskNet is the current and adjacent frames $I = [I_k, I_f]$, and the output is the consistent masks M_f corresponding to the reconstructed frames $\hat{I}_f$. Each element on the consistent mask indicates probability that the pixel satisfies photometric consistency assumption. If pixel p_t satisfies photometric consistency assumption in the adjacent frames, we have $I_t(p_t) = \hat{I}_f(p_t), f \in \{t-1, t+1\}$, and $M_{t-1}(p_t) = M_{t+1}(p_t)$. When occlusion only occurs in the past frame, we have $\parallel I_t(p_t) - \hat{I}_{t-1}(p_t) \parallel_1 > \parallel I_t(p_t) - \hat{I}_{t+1}(p_t) \parallel_1$ and $M_{t-1}(p_t) < M_{t+1}(p_t)$. Otherwise, we have $M_{t-1}(p_t) > M_{t+1}(p_t)$. We extract occlusion masks V_{t-1} and V_{t+1} from consistent masks M_{t-1} and M_{t+1} to indicate whether pixels are visible on the adjacent frames. When $M_{t-1}(p_t) > M_{t+1}(p_t)$, p_t is more likely to be visible in the past frame than in the future; as a result, $V_{t-1}(p_t) = 1, V_{t+1}(p_t) = 0$. If $M_{t-1}(p_t) = M_{t+1}(p_t)$, there are two situations; if $M_{t-1}(p_t)$ and $M_{t+1}(p_t)$ tend to zero, there may be dynamic objects in the adjacent frames, and if they tend to one, there are no dynamic objects. For occlusion, we let $V_{t-1}(p_t) = V_{t+1}(p_t) = 0.5$, it means p_t is visible in all adjacent frames.

Similar to SfmLearner [10], we add a regularization term of mask, that is,

$$L_m = -e^{\beta \| M_{t-1} - M_{t+1} \|_1} (\log M_{t-1} + \log M_{t+1}). \tag{5}$$

In other words, the loss prevents the mask to always be zero, since most points in the scene meet the photometric consistent. We also introduce the smoothing loss of the mask to ensure that the pixels in the neighborhood have the similar state, that is,

$$L_{ms} = \sum_{f \in \{t-1, t+1\}} | \bigtriangledown_x M_f | e^{-|\bigtriangledown_x I_t|} + | \bigtriangledown_y M_f | e^{-|\bigtriangledown_y I_t|}. \tag{6}$$

3.4. Feature Pyramid Matching Loss

To consider both global high-level and local detailed information, we extract feature pyramid from images and construct FPML for reducing the effect of brightness change and non-Lambertian surface. Figure 3 summarizes the key processes of FPML, which consists of feature pyramid and matching error. Given current image I_t and adjacent frames I_f, we generate L levels pyramid feature, lth current feature map c_t^l and lth adjacent feature map c_f^l. Specifically, current image and adjacent frames are input to DepthNet, and the layers of conventional filters output the different scale feature maps to construct the feature pyramid. The encoder module of DepthNet generates a feature pyramid with $L = 5$ layers, and the numbers of feature channels are 64, 64, 128, 256 and 512. FPML makes use of the features generated in the encoder and therefore causes a minimal overhead. We synthesize the current frame feature map by warping $\hat{c}_f^l = g(D_t^l, T_{t\to f}, c_f^l)$ according to the feature map c_f^l generated by adjacent frames, downsampled depth mapD_t^l of current frame and camera egomotion $T_{t\to f}$. The resolution of D_t^l is same as that of lth feature map c_f^l. The corresponding feature of the same object in different frames is similar regardless of brightness changes, occlusion and dynamic objects. Thus, we define cosine similarity loss between lth feature maps as follows,

$$L_f^l = 1 - \frac{{c_t^l}^{\mathrm{T}} \hat{c}_f^l}{\parallel c_t^l \parallel \parallel \hat{c}_f^l \parallel}. \tag{7}$$

The total FPML function is

$$L_f = \sum_{f \in \{t-1, t+1\}} \sum_{l \in \{0,1,2,3,4\}} L_f^l. \tag{8}$$

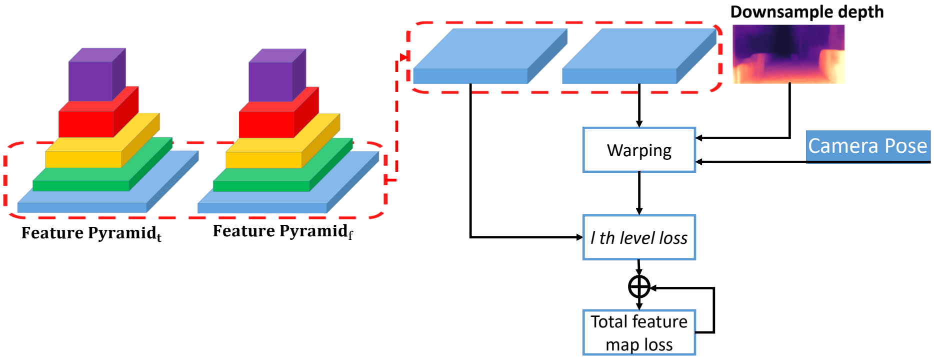

Figure 3. Feature pyramid matching error of the current frame and adjacent frame. Feature pyramid is constructed by different scale feature maps. Adjacent feature maps warped using the downsampled depth and camera pose computes a matching error.

3.5. Network Architecture

DepthNet and MaskNet

The DepthNet and MaskNet we proposed based on encoder-decoder architecture, in which the decoder part can share the shallow information of the encoder part through skip connections.

The encoder part adopts the standard ResNet18 [33], which contains 11M parameters and uses the weights pretrained on ImageNet as the initial parameters. The difference of the encoder parts between the DepthNet and MaskNet is the number of input images. The first convolution layer parameter of the DepthNet is 3 × 64 × 3 × 3. The first convolution layer parameter of the MaskNet is set as 9 × 64 × 3 × 3 for adapting to the input images.

In the decoder modules, ELU [34] is adopted as all nonlinear activation functions; five times of upsampling can obtain the feature map with the same resolution of input image, and the upsampling parts use bilinear interpolation. Like SfmLearner [10], the decoder output layer of the DepthNet is activated by sigmoid and converted into a non-negative reasonable depth map. The process is formulated as $D = \frac{1}{a*sigmoid(x)+b}$, where $a = 10$ and $b = 0.1$. The MaskNet uses sigmoid activation to output two channels mask images corresponding to the adjacent frames. Similar to Godard et al. [9] in border filling, we use reflection padding instead of zero padding, which can reduce the border artifacts of the depth map.

MotionNet

The input of MotionNet contains RGB images of the current frame and adjacent frames, and the outputs are camera poses of the current frame and adjacent frames. MotionNet consists of a ResNet18 and four convolution layers. The parameter of ResNet18 input layer is $9 \times 64 \times 3 \times 3$, and the weights pretrained in ImageNet are also used as initial parameters. All activation functions use RELU, except for the last output layer. The output of the last layer is two channels 6D vector $\phi \in \mathbb{R}^{2*6}$, including a 3D rotating Euler angle and a 3D position.

4. Experiments

In this section, we compare results of our method with existing state-of-the-art approaches on depth and camera egomotion estimation.

4.1. Experimental Settings

Implementation details

Our models are implemented with PyTorch [35] and trained for 20 epochs. We set the initial value of loss weights based on experience and other similar papers [9–11], and then tune them with a sampled validation set from training images. In our entire training process, we set weight of depth smoothness loss $\lambda_s = 10^{-3}$, weight of mask regularization term $\lambda_m = 0.12$, weight of mask smooth loss $\lambda_{ms} = 10^{-3}$ and weight of FPML $\lambda_f = 0.01$. During training, we use the Adam optimizer [36] with $\beta_1 = 0.9$, $\beta_2 = 0.999$. We also set the learning rate of the first 15 epochs to 10^{-4}, and then to 10^{-5} and mini-batch size of 12. All the images in experiments are from KITTI monocular image sequences.

KITTI dataset

We use the KITTI [37] dataset as the main dataset for training and testing. In previous works [7–15,28,29], KITTI is often used to evaluate performance on depth and egomotion. The KITTI dataset contains images collected by four cameras (two grayscale and two RGB), as well as point cloud collected by a Velodyne HDL-64E laser scanner and pose collected by GPS/IMU. The KITTI dataset provides videos from 200 different scenes, including city streets, roads and campus, etc. During the training, 156 image sequences without test scenes are used, and the left and right images are treated independently. Furthermore, we follow SfmLearner's preprocessing to remove static frames [10]. A total of 40,109 are obtained for training and 4431 for validation. We choose the Eigen split [7] for depth testing. The Eigen split consists of 697 images, where the depth ground truth is obtained by projecting the Velodyne laser scanned points into the image plane. During the training, the input images are resized to resolution of 640 × 192, and the camera intrinsic matrix are known. During the validating and testing, the input images use the resolution of 1216 × 352. KITTI Odometry dataset has 00–10 sequences with pose labels. We follow SfmLearner [10], and split sequences 00–08 for training and 09–10 for testing.

Evaluation metric

We use the depth evaluation metric of Eigen et al. [7]. The explanation of each metric adopted in our evaluation is specified in Table 1, where D^* and D represent the ground truth and estimated depths respectively.

We use absolute trajectory error (ATE) [38] to evaluate camera motion. ATE first aligns the estimated camera motion with the ground truth pose and then evaluates the relative error of camera pose.

Table 1. Depth evaluation metric.

Abs Rel:$\frac{1}{\vert T\vert}\sum_{D\in T}\frac{\vert D-D^*\vert}{D^*}$
Sq Rel: $\frac{1}{\vert T\vert}\sum_{D\in T}\frac{\vert D-D^*\vert^2}{D^*}$
RMSE log: $\sqrt{\frac{1}{\vert T\vert}\sum_{D\in T}\vert\log\frac{D}{D^*}\vert^2}$
RMSE:$\sqrt{\frac{1}{\vert T\vert}\sum_{D\in T}\vert D-D^*\vert^2}$
δ_t:% of $D\in T\max(\frac{D^*}{D},\frac{D}{D^*})<t$

4.2. Depth Estimation Results

Quantitative comparison results of our method and previous methods are shown in Table 2. The mono column denotes whether stereo camera is used, M means monocular, S indicates stereo. The supervised column denotes whether additional supervised information is used. In the first row, the upward arrow ↑ indicates higher is better, the downward arrow ↓ means lower is better. The best results in each category are printed in bold. Following other traditional methods [7,10], we limit the maximum depth to 80 m. Depth estimation in an unsupervised manner from monocular videos obtains related depth. So, we multiply the estimated depth by the median scale factor $s = median(D^*)/median(D)$ [10] for comparison with absolute depth generated from stereo camera or supervised methods. Our method outperforms previous supervised methods [7,39] and unsupervised methods [6,9–11,13–17,40,41]. Compared with these works mentioned above, our method reduces Abs Rel by 0.017–0.088, Sq Rel by -0.039–0.700, RMSE by 0.187–1.752 and RMSE log by 0.015–0.083. Compared with Struct2depth(M) [13], which uses motion model, our result is 0.021 better than Struct2depth(M) in terms of Abs Rel, 0.187 better than that in terms of RMSE, 0.015 better than that in terms of RMSE log, 0.052 better than that in terms of $\delta < 1.25$, and 0.009 better than that in terms of $\delta < 1.25^2$, except in Sq Rel and $\delta < 1.25^3$. It is also worth noting that on the metric of Abs Rel, our method outperforms other methods. This metric measures the ratio of prediction error over the ground truth value and can be used to compare the reliability of different depth measurement results. The good performance under this metric indicates that our method produces consistent depth at long and short distances.

Our DepthNet and MotionNet are the same as those of methods in literature [10,13], so the network inference time is also the same. Our test results show that for predicting depth it takes 3.972 s to load model and initialize, 0.020 s for network inference and 0.003 s for postprocessing. For predicting camera egomotion, it takes 4.132 s to load model and initialize, 0.005 s for network inference and 0.002 s for postprocessing.

In Figure 4, our experimental results are compared with Sfmlearner, DDVO, GeoNet and Monodepth methods. The first line is the original image and the following is the depth maps generated by each method. The higher intensity of red in the depth map, the closer the distance. The blue boxes in Figure 4 are the areas we focus on, which include objects with broad shape as well as thin objects. Compared with other methods, the depth maps produced by our method are clearer and the edges are sharper in both cases. In the blue boxes of first column images, there is a farther vehicle. DDVO, GeoNet and Monodepth do not estimate its depth, but our method estimates its depth accurately. The boxes in the second and third columns of images include slender pillars, and the boundaries of these objects estimated by other methods are blurry. The green dotted boxes in the image indicate obvious defects in other baselines. We can see that our models generate higher quality outputs and do not produce "holes" in the depth maps. There are holes in the ground in the results of the SfmLearner, which may lead to autonomous vehicles misjudge the passing area. In the results of Monodepth, the depth estimation of the edge area of the image is wrong, which may be caused by the lack of covisible areas in the edge of the stereo images. As shown in Figure 5a, a black region obtained from OAM indicates a possible occlusion in the previous frame. Figure 5b obviously indicates dynamic objects in the scene learned from the MaskNet.

Table 2. Depth estimation quantitative results on Eigen [7] split of KITTI raw dataset [37], capped at 80 m. These methods are all trained on KITTI raw dataset. The camera column denotes whether stereo camera is used, M means monocular, S indicates stereo. The supervised column denotes whether using additional supervised information. In the first row, the upward arrow ↑ indicates higher is better, the downward arrow ↓ means lower is better. Best results in each category are in bold.

Method	Supervied	Camera	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Eigen [7]	Depth	M	0.203	1.548	6.307	0.282	0.702	0.890	0.890
Liu [39]	Depth	M	0.201	1.584	6.471	0.273	0.680	0.898	0.967
SfmLearner [10]	-	M	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Yang [15]	-	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Vid2depth [16]	-	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
LEGO [14]	-	M	0.162	1.352	6.276	0.252	0.783	0.921	0.969
GeoNet [11]	-	M	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DDVO [17]	-	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
Monodepth [9]	Pose	S	0.148	1.344	5.927	0.247	0.803	0.922	0.964
CC [40]	-	M	0.148	1.149	5.464	0.226	0.815	0.935	0.973
EPC++ [41]	-	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2depth(M) [13]	-	M	0.141	**1.026**	5.291	0.215	0.816	0.945	**0.979**
SC-SfmLearner [6]	-	M	0.137	1.089	5.439	0.217	0.830	0.942	0.975
Ours	-	M	**0.120**	1.065	**5.104**	**0.200**	**0.868**	**0.954**	0.978

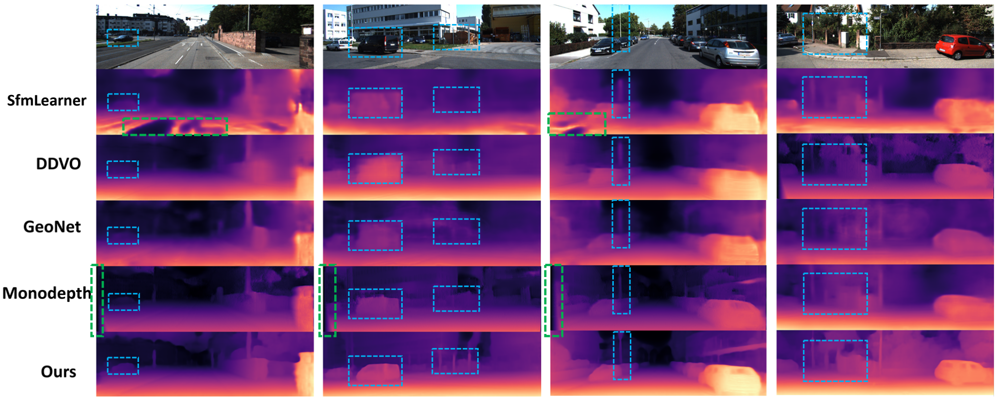

Figure 4. Qualitative KITTI results. Our method is compared with the results of SfmLearner [10], DDVO [17], GeoNet [11] and Monodepth [9]. The higher intensity of red in the picture, the closer the distance. The results in the blue dashed boxes are the areas we focus on. The results in the green dashed boxes are "holes" in the depth maps.

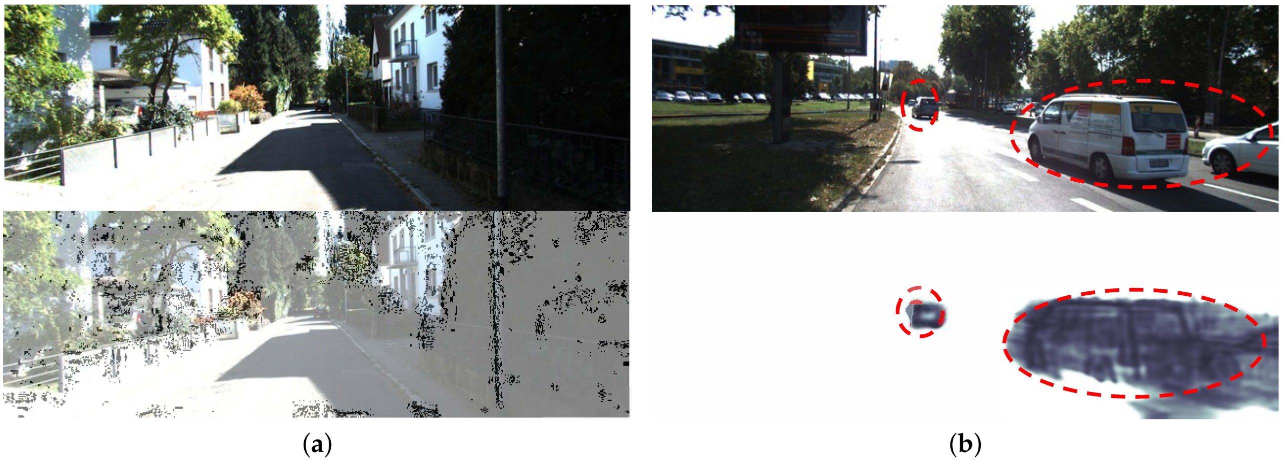

Figure 5. (**a**) occlusion mask (**b**) moving objects mask.

4.3. Camera Pose Estimation Results

Our method is compared not only with the traditional visual SLAM method [42] but also with other deep learning methods [10,29]. The quantitative evaluation of camera egomotion estimation is shown in Table 3. Table 3 shows that our camera egomotion results exceed unsupervised learning method monodepth2 [29] and SfmLearner [10] in 09 and 10 sequences in terms of the ATE [38]. We compare our egomotion estimation with two variants of monocular ORB-SLAM [42]. The results show that our method has an advantage over ORB-SLAM(short), which runs on five-frame snippets. Our results are not as good as ORB-SLAM(full) because ORB-SLAM(full) is a complete SLAM system including loop closure and relocalization, which uses all images in the sequence.

Table 3. Absolute Trajectory Error (ATE) on the KITTI Odometry sequences 09 and 10 (lower is better).

Method	**Seq.09**	**Seq.10**
ORB-SLAM(full) [42]	**0.014 ± 0.008**	**0.012 ± 0.011**
Mean Odometry	0.032 ± 0.026	0.028 ± 0.023
ORB-SLAM(short)	0.064 ± 0.141	0.064 ± 0.130
Monodepth2 [29]	0.023 ± 0.013	0.018 ± 0.014
SfmLearner [10]	0.021 ± 0.017	0.020 ± 0.015
Ours	**0.019 ± 0.009**	**0.013 ± 0.010**

4.4. Ablation Study

We measure the impact of each contribution on performance and show the results of ablation study in Table 4 to understand which part of our method contributes to the performance. In Table 4, the baseline model following recent works [10,11] does not contain any of our contributions; +F represents the contribution of FPML; +OM indicates the contribution of OAM. Comparing with the baseline, the performance is improved by adding the FPML or OAM. In the main metric Abs Rel, the contribution of FPML is 0.01 better than that of the baseline. Moreover, the contribution of OAM is 0.013 better than that of the baseline. The combination of these contributions improves performance by 0.02 better than the baseline in terms of Abs Rel.

Table 4. Ablation studies on FPML and OAM. +F represents the contribution of FPML. +OM indicates the contribution of OAM. Each of our contributions improves performance.

Method	**Abs Rel**	**Sq Rel**	**RMSE**	**RMSE log**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.140	1.610	5.512	0.223	0.852	0.946	0.973
+F	0.130	0.974	5.197	0.208	0.840	0.948	0.979
+OM	0.127	0.957	5.163	0.202	0.852	0.953	0.980
+OM+F	0.120	1.065	5.104	0.200	0.868	0.954	0.978

5. Conclusions

We propose an unsupervised learning framework that achieves monocular depth and egomotion estimation via FPML and OAM. The introduced FPML captures the local and global information and reduces the influence of brightness variation and non-Lambertian surface. In addition, the proposed OAM predicts not only dynamic objects but also occluded pixels in an innovative manner according to change of masks. As a result, FPML and OAM address the problem of photometric inconsistency and improve accuracy of depth and camera pose estimation. On the KITTI dataset, our results are better than the state-of-the-art unsupervised methods and even some supervised methods, both qualitatively and quantitatively. Especially, compared with previous methods, our method reduces Abs Rel by 0.017–0.088, which is the most important metric in the literature.

In our future works, we will estimate the 3D motion of the dynamic rigid object in the image to help the robot better understand the 3D environment. Furthermore, the camera and LiDAR information will also be fused to achieve real-time accurate depth estimation, which is used for localization and mapping.

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

Funding: This work was supported by the National Natural Science Foundation of China under Grant NSFC 61671153.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

FPML	Feature Pyramid Matching Loss
OAM	Occlusion-Aware Mask
Abs Rel	Absolute Relative Error
SAD	Sum of Absolute Differences
SSIM	Structural Similarity
SAD	Sum of Absolute Differences
SOTA	State-Of-The-Art
CNN	Convolutional Neural Network
Mask R-CNN	Mask Region—Convolutional Neural Networks
Sq Rel	Squared Relative Error
RMSE	Root Mean Squard Error
ELU	exponential linear unit
RELU	Rectified Linear Unit

GPS	Global Position System
IMU	Inertial Measurement Unit
ATE	Absolute Trajectory Error

References

1. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.A.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.J.; et al. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; Pierce, J.S., Agrawala, M., Klemmer, S.R., Eds.; ACM: New York, NY, USA, 2011; pp. 559–568. [CrossRef]
2. Lin, J.; Zhang, F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, 31 May–31 August 2020; pp. 3126–3131. [CrossRef]
3. Rosin, P.L.; Lai, Y.K.; Shao, L.; Liu, Y. *RGB-D Image Analysis and Processing*; Springer: Berlin/Heidelberg, Germany, 2019.
4. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26–30 May 2015; pp. 2174–2181. [CrossRef]
5. Bloesch, M.; Czarnowski, J.; Clark, R.; Leutenegger, S.; Davison, A.J. CodeSLAM—Learning a Compact, Optimisable Representation for Dense Visual SLAM. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2560–2568. [CrossRef]
6. Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.; Reid, I.D. Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; 2019; pp. 35–45.
7. 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 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; 2014; pp. 2366–2374.
8. Eigen, D.; Fergus, R. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; pp. 2650–2658. [CrossRef]
9. Godard, C.; Aodha, O.M.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 6602–6611. [CrossRef]
10. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised Learning of Depth and Ego-Motion from Video. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 6612–6619. [CrossRef]
11. Yin, Z.; Shi, J. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1983–1992. [CrossRef]
12. Vijayanarasimhan, S.; Ricco, S.; Schmid, C.; Sukthankar, R.; Fragkiadaki, K. Sfm-net: Learning of structure and motion from video. *arXiv* **2017**, arXiv:1704.07804.
13. Casser, V.; Pirk, S.; Mahjourian, R.; Angelova, A. Depth Prediction without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019; AAAI Press: Menlo Park, CA, USA, 2019; pp. 8001–8008. [CrossRef]
14. Yang, Z.; Wang, P.; Wang, Y.; Xu, W.; Nevatia, R. LEGO: Learning Edge With Geometry All at Once by Watching Videos. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 225–234. [CrossRef]
15. Yang, Z.; Wang, P.; Xu, W.; Zhao, L.; Nevatia, R. Unsupervised Learning of Geometry From Videos With Edge-Aware Depth-Normal Consistency. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; AAAI Press: Menlo Park, CA, USA, 2018; pp. 7493–7500.
16. Mahjourian, R.; Wicke, M.; Angelova, A. Unsupervised Learning of Depth and Ego-Motion From Monocular Video Using 3D Geometric Constraints. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5667–5675. [CrossRef]
17. Wang, C.; Buenaposada, J.M.; Zhu, R.; Lucey, S. Learning Depth From Monocular Videos Using Direct Methods. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2022–2030. [CrossRef]
18. Zou, Y.; Luo, Z.; Huang, J. DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency. In *Computer Vision–ECCV 2018, Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018*; Proceedings, Part V; Lecture Notes in Computer Science; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11209, pp. 38–55. [CrossRef]

19. Yang, N.; von Stumberg, L.; Wang, R.; Cremers, D. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; pp. 1278–1289. [CrossRef]
20. 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]
21. Li, J.; Klein, R.; Yao, A. Learning fine-scaled depth maps from single rgb images. *arXiv* **2016**, arXiv:1607.00730.
22. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. In Proceedings of the Fourth International Conference on 3D Vision, 3DV 2016, Stanford, CA, USA, 25–28 October 2016; pp. 239–248. [CrossRef]
23. Xu, D.; Ricci, E.; Ouyang, W.; Wang, X.; Sebe, N. Multi-scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 161–169. [CrossRef]
24. Mayer, N.; Ilg, E.; Fischer, P.; Hazirbas, C.; Cremers, D.; Dosovitskiy, A.; Brox, T. What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation? *Int. J. Comput. Vis.* **2018**, *126*, 942–960. [CrossRef]
25. Garg, R.; Kumar, B.G.V.; Carneiro, G.; Reid, I.D. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In *Computer Vision–ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part VIII; Lecture Notes in Computer Science; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9912, pp. 740–756. [CrossRef]
26. Harltey, A.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2006.
27. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
28. Yang, Z.; Wang, P.; Wang, Y.; Xu, W.; Nevatia, R. Every Pixel Counts: Unsupervised Geometry Learning with Holistic 3D Motion Understanding. In *Computer Vision—ECCV 2018 Workshops, Munich, Germany, 8–14 September 2018*; Proceedings, Part V; Lecture Notes in Computer, Science; Leal-Taixé, L., Roth, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11133, pp. 691–709. [CrossRef]
29. Godard, C.; Aodha, O.M.; Firman, M.; Brostow, G.J. Digging Into Self-Supervised Monocular Depth Estimation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October–2 November 2019; pp. 3827–3837. [CrossRef]
30. Janai, J.; Güney, F.; Ranjan, A.; Black, M.J.; Geiger, A. Unsupervised Learning of Multi-Frame Optical Flow with Occlusions. In *Computer Vision—ECCV 2018, Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018*; Proceedings, Part XVI; Lecture Notes in Computer Science; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11220, pp. 713–731. [CrossRef]
31. Sun, D.; Yang, X.; Liu, M.; Kautz, J. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8934–8943. [CrossRef]
32. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December2015; 2015; pp. 2017–2025.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
34. Clevert, D.; Unterthiner, T.; Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.
35. 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 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
36. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
37. 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. [CrossRef]
38. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems—IROS 2012, Vilamoura, Algarve, Portugal, 7–12 October 2012; pp. 573–580. [CrossRef]
39. 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] [PubMed]
40. 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 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 12240–12249. [CrossRef]

41. Luo, C.; Yang, Z.; Wang, P.; Wang, Y.; Xu, W.; Nevatia, R.; Yuille, A.L. Every Pixel Counts ++: Joint Learning of Geometry and Motion with 3D Holistic Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2624–2641. [CrossRef] [PubMed]
42. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]