Article

Joint Unsupervised Learning of Depth, Pose, Ground Normal Vector and Ground Segmentation by a Monocular Camera Sensor

Lu Xiong, Yongkun Wen, Yuyao Huang, Junqiao Zhao and Wei Tian *

Institute of Intelligent Vehicles, School of Automotive Studies, Tongji University, Shanghai 201804, China; xiong_lu@tongji.edu.cn (L.X.); 1933550@tongji.edu.cn (Y.W.); huangyuyao@tongji.edu.cn (Y.H.); zhaojunqiao@tongji.edu.cn (J.Z.)

* Correspondence: tian_wei@tongji.edu.cn

Received: 1 June 2020; Accepted: 26 June 2020; Published: 3 July 2020

Abstract: We propose a completely unsupervised approach to simultaneously estimate scene depth, ego-pose, ground segmentation and ground normal vector from only monocular RGB video sequences. In our approach, estimation for different scene structures can mutually benefit each other by the joint optimization. Specifically, we use the mutual information loss to pre-train the ground segmentation network and before adding the corresponding self-learning label obtained by a geometric method. By using the static nature of the ground and its normal vector, the scene depth and ego-motion can be efficiently learned by the self-supervised learning procedure. Extensive experimental results on both Cityscapes and KITTI benchmark demonstrate the significant improvement on the estimation accuracy for both scene depth and ego-pose by our approach. We also achieve an average error of about 3° for estimated ground normal vectors. By deploying our proposed geometric constraints, the IOU accuracy of unsupervised ground segmentation is increased by 35% on the Cityscapes dataset.

Keywords: unsupervised learning; scene depth; ego-motion; ground segmentation; ground normal vector

1. Introduction

Estimation of scene depth, ground plane and ground normal vector by camera sensors has been playing a key role in the field of automated driving and robotics. Together with the inter-frame motion estimation, they can provide a priori knowledge about a scene structure. Since such information is essential to tasks such as tracking [1], 3D object detection [2,3] and camera pose estimation [4], numerous approaches related to scene structure prediction and analysis have been proposed in recent years.

Generally, scene structures can be estimated either by the traditional geometric vision methods or by deep learning based approaches. For scene depth and ego-motion estimation, traditional methods typically leverage extensive feature matching in accordance with multi-view geometry while the results strongly rely on the matching accuracy and measured camera parameters [5–8]. In deep learning based approaches, especially in unsupervised learning, the input data is only continuous RGB video streams, and no additional supervised signals or labels are required. The inter-frame motion and scene depth can be estimated by the network itself through reconstructing the image from one frame to another w.r.t. photometric errors. In spite of impressive results achieved [9–13], one persisting problem for such methods is that dynamic objects do not conform to the original static scene assumption. In terms of ground structure estimation, traditional methods mainly leverage multi-view geometry, e.g., by fitting a large plane to the obtained 3D point cloud [14], or by estimating vanishing points and horizons using parallel lines on the ground [7]. These methods either require guidance by additional signals such

as lidar points or strongly rely on intermediate scene element predictions, which are inherently less robust and difficult to apply to a wider range of scenarios. In deep learning methods, ground plane predictions are typically learned in a supervised fashion [15], yet the required supervision signal is usually cumbersome in data collection. In addition, the link between scene structures has been taken into consideration in recent researches [16,17], which is beneficial for the entire scene understanding. Given these circumstances, we propose a unified and completely unsupervised learning framework, which can estimate the scene depth, ego-motion, ground normal vector and ground segmentation simultaneously. This framework only needs continuous video streams and, thus, no annotation for the scene is required. To make full use of the information of different scene structures, we propose a joint learning process. In this process, the estimated depth is used to restore 3D coordinates for the corresponding image points. The ground segmentation network is trained using point labels refined by estimated ground plane and its normal vector. The corresponding loss is called ground self-learning loss. To make use of the ground information in dynamic scenes (with ego-motion), we propose the plane photometric loss, which considers plane transform errors between frames by the homography matrix. We also propose another loss to punish depth abnormality in regions near the vanishing point obtained by the ground normal vector. By filtering out points with abnormal depth values, the network model can be further optimized. We validate our approach on both Cityscapes [18] and KITTI [19] benchmark and a significant improvement on estimation accuracy for both scene depth and ego-pose has been proven. In comparison with other unsupervised methods, we achieve an accuracy gain of about 1.1° in ground normal vector estimation and 35% in unsupervised ground segmentation, which further demonstrate the efficacy of our approach.

In this paper, our contributions are summarized as follows:

- An unsupervised learning framework is proposed, which can estimate the scene depth, ego-motion, ground normal vector and ground segmentation simultaneously.
- A joint learning process is proposed, which uses heterogeneous loss functions to boost the mutual information flow between the estimation of different scene structures.
- Extensive comparison experiments and ablation studies on public datasets are conducted and demonstrate the improvement of proposed approach on estimation of scene structures such as the depth, ego-pose, ground segmentation and ground normal vector.

2. Related Work

In accordance with the focus of this paper, related works are reviewed in following three aspects: depth and ego-motion estimation, unsupervised semantic segmentation, and ground normal vector estimation.

2.1. Depth and Ego-Motion Estimation

Traditional depth estimation methods mainly rely on multi-view geometry to perform 3D restoration through inter-frame registration [7]. With the development of deep learning, Convolutional Neural Networks (CNNs) are used for depth prediction. Eigen et al. [20] proposed a supervised approach with two networks to estimate depth in a coarse-to-fine order: one network makes global prediction of the entire graph, while the other refines the local information. Similar works can be seen in [21–24]. Although such methods work well, the cost of depth labels restricts these methods. To bypass this restriction, stereo image pairs are used in other works, where the left frame is reconstructed through the predicted depth and the known inter-frame pose, and the photometric error is used as the supervision signal during training [25,26]. Despite no explicit supervision, these methods require more accurate sensor calibration. Zhou et al. [9] first proposed an unsupervised framework only requiring video streams. In their approach, images are reconstructed using depth and inter-frame motion, which is estimated by a self-motion estimation network. However, this method is based on the assumption of static scene, where dynamic objects can contaminate the network prediction. In order to deal with dynamic objects, optical flow [12] and dynamic object segmentation [13] are added by

researchers. These tricks improve the accuracy, but also increase the computation cost by employing many intermediate processes. Bian et al. [27] proposed the geometry consistency loss to reduce the relative depth error of the same spatial point in two frames. In their approach, the depth consistency is also used as the weight of the photometric error, reducing the impact of dynamic objects.

2.2. Unsupervised Semantic Segmentation

Unsupervised semantic segmentation can theoretically use the method of grouping. Typical approaches for grouping are information maximizing self-augmented training (IMSAT) [28] by maximizing the mutual information between data and its representation, and Deep InfoMax (DIM) [29], which maximizes the information between spatially preserved features and compact features. Deep clustering is also used in a lot of grouping approaches for image, and learns highly transferable intermediate features using overclustering as a proxy task. Yang et al. [30] iteratively learn convolutional network features and clusters with a recurrent framework. Their model offers promising performance on small datasets. However, these methods either introduce additional grouping criteria which increase the method complexity or are only proven effective on small datasets. In contrast, the unsupervised method proposed by Ji et al. [31] use mutual information loss for supervision and because the training process is not easy to degrade, it improves the ability on large datasets. In the approach of Ji et al. [31], they perform a random transform on a single image and send both the transformed image and the original one into the network respectively. After network processing, an inverse transform is performed on the transformed image. By maximizing the mutual information of corresponding pixel in both images, their model can better learn shallow semantic information.

2.3. Ground Normal Vector Estimation

Multi-view geometry is broadly used in traditional methods to estimate the structural feature of the plane in monocular images [32], e.g., by detecting the vanishing point and the horizon through parallel lines on the plane. However, this type of method heavily relies on the performance of manual features and is less robust. Other methods attempt to learn ground normal vectors for each pixel and cluster the results to obtain planes [16,17,33,34]. There are also indirect methods for obtaining normal vectors through 3D reconstruction [32,35,36]. But these methods are either for indoor scenarios or require surface labels. In a recent work, Man et al. [15] estimate the ground normal and depth in two separate streams. A consistency loss is added to enhance the accuracy of the normal vector. Although this work obtains better accuracy, it still requires labels for depth, normal vector and ground segmentation.

Our approach is based on preliminary works [27,31]. To enhance the ground segmentation performance, we add geometric priors (the ground normal) in the joint training process. Additionally, we add information of static ground plane and through the plane photometric loss we improve the performance of depth and pose estimation. The ground normal vector is naturally calculated by ground separation and its depth. Under the mutual promotion between subtasks, the estimation of depth, pose, ground normal vector and ground segmentation can be improved.

3. Proposed Method

Mutual information loss (MI) has been used for unsupervised segmentation in preliminary work [31]. However, for complicated traffic scenes, the result becomes unreliable, as shown in the middle of Figure 1. Because the supervision signal is insufficient, it can only learn shallow semantic features and give a very rough hypothesis about the ground. In our approach, we merge the ground segmentation with a structure from motion (SfM) framework, so that the tasks of the estimation of depth, ego-motion, ground segmentation and ground normal vector can promote each other. For this purpose, we add the ground self-learning loss L_r , the plane photometric loss L_H , and the depth abnormal punishment loss L_v . We also use the ground normal vector to improve the origin loss in the SfM framework.

Figure 1. From left to right are: the input image, the ground segmentation result obtained only using the mutual information loss (MI), and the result by joint learning which uses the ground self-learning loss (GSFL). It can be seen that after adopting the geometric prior, the ground segmentation results have been enhanced.

The training of our approach consists of two stages. The first stage is for pre-training the ground segmentation network (Figure 2) and the SfM framework (Figure 3), respectively. In the second stage, we perform the joint learning (Figure 4) with the aforementioned three losses and use the normal vector to improve the origin loss in SfM framework. Thus, our entire loss function is defined as

$$L = L_i + L_d + L_H + L_r + L_v, \quad (1)$$

where L_i and L_d respectively denote the mutual information loss and the scene reconstruction loss from the first training phase. In the second phase, the L_d is improved to L'_d which will be explained in the Joint learning part, and L_H, L_r, L_v are adopted. Note that for an intuitive expression, the weight for each loss is implicitly integrated in their formulations (introduced in following subsections). By the above training procedure, the SfM framework can guide the learning of the ground segmentation. Through the ground self-learning loss and the depth abnormal punishment loss, the learned ground-related information feeds back to the SfM framework, and finally obtains a stronger scene structure estimation.

Figure 2. The original image I_a undergoes a random transform to generate $g(I_a)$. Both of them are then passed separately into the segmentation net, and yield the probability feature map $\Phi(g(I_a))$ and $\Phi(I_a)$ to calculate the mutual information loss L_i .

Figure 3. The unsupervised depth learning framework in our approach. Detailed description about its structure can be seen in the text.

Figure 4. New losses L_r , L_H and L_v are added in the second learning stage. In comparison with the first learning stage, the normal vector is fused to improve the calculation of loss L_d , forming the new loss L'_d . Details about this learning stage can be seen in the text.

In following parts, the calculation of corresponding losses as well as training details are introduced. Specifically, the loss L_i and L_d used in the first learning stage will be introduced in Sections 3.1 and 3.2, respectively. In Section 3.3, the loss L_r , L_H , L_v will be represented, which are added in the second stage, and the improved loss L'_d will be described. Finally, in Section 3.4, a small summary of the entire framework will be expressed.

3.1. Self-Supervised Ground Segmentation

Here we adopt the mutual information loss to make a coarse prediction for the ground. By maximizing the mutual information loss of image pairs, image regions within the same semantic class should have similar probability distribution, and thus belong to the same semantic segment.

Given two discrete probability distributions $\mathbf{z}$ and $\mathbf{z}'$, the joint probability distribution matrix $\mathbf{P}$ can be calculated by function h as

$$\mathbf{P} = h(\mathbf{z}, \mathbf{z}') = \frac{\mathbf{z}\mathbf{z}'^\top + \mathbf{z}'\mathbf{z}^\top}{2}. \quad (2)$$

Thus, the mutual information [31] is expressed as

$$MI(\mathbf{P}) = \sum_{c=1}^C \sum_{c'=1}^C \mathbf{P}_{cc'} \ln \frac{\mathbf{P}_{cc'}}{\mathbf{P}_c \mathbf{P}_{c'}}, \quad (3)$$

where $\mathbf{P}_c$ represents the marginal probability of class c in $\mathbf{z}$ and $\mathbf{P}_{c'}$ represents the marginal probability of class c' in $\mathbf{z}'$. $\mathbf{P}_{cc'}$ denotes the joint probability for c and c' . To extend this concept, the joint probability distribution matrix $\mathbf{P}_{I_a}$ for the image I_a and its transformed version $g(I_a)$ (see Figure 2) can be defined as

$$\mathbf{P}_{I_a} = \frac{1}{|V_i|} \sum_{(x,y) \in V_i} h(\Phi(I_a)_{xy}, g^{-1}(\Phi(g(I_a)))_{xy}), \quad (4)$$

where $g(\cdot)$ represents a random set of image transform, including translation, rotation, hue change, etc. And $g^{-1}(\cdot)$ denotes the inverse transform. Term $\Phi(\cdot)$ indicates the probability map generated by a segmentation network. Subscript xy denotes the image coordinate of point (x, y) . Set V_i contains valid points which remain in the image after the transform $g(\cdot)$ or $g^{-1}(\cdot)$. And $|V_i|$ denotes the number of points in this set.

For semantic segmentation, the mutual information loss L_i can be expressed as

$$L_i = -MI(\mathbf{P}_{I_a}). \quad (5)$$

This method can only yield a very rough cluster as shown in the middle of Figure 1, which is far from enough in complex scenes. Thus, in our approach, this coarse ground segmentation is combined with the depth information to expand the ground part, i.e., to optimize the ground segmentation result by compensating the geometric information. The depth information is estimated by a SfM framework introduced below.

3.2. SfM Framework

The overall SfM framework in our approach is illustrated in Figure 3. The depth D_a of frame I_a is estimated through the DepthNet. Two consecutive frames I_a, I_b are sent to the PoseNet to estimate the ego-motion $[\mathbf{R}_a^b, \mathbf{t}_a^b]$. Then, the depth D_a , pose $[\mathbf{R}_a^b, \mathbf{t}_a^b]$ and frame I_b are used to reconstruct frame $I_{a'}$ to calculate the photometric loss. The smooth loss and geometry consistency loss are also added in the training. Therefore, the scene reconstruction loss function is defined as

$$L_d = \alpha L'_p + \beta L_s + \gamma L_{GC}, \quad (6)$$

where L'_p is the depth constrained photometric loss, L_s is the smooth loss and L_{GC} is the geometric consistency. And α, β, γ are hyper-parameters for trade-off between individual losses.

The loss based on Structural Similarity (SSIM) [37] comprehensively considers the difference of lighting, contrast, and image structure, while the simple L_1 distance is more sensitive to illumination. The superiority of SSIM in image reconstruction has been proven in Bian et al. [27]. Thus, the loss based on SSIM is more suitable to be integrated in the photometric loss function. (Details can be seen in Appendix C.) So, L_p is calculated as

$$L_p = \frac{1}{|V_d|} \sum_{p \in V_d} \left((1 - \lambda_s) \|I_a(p) - I_{a'}(p)\|_1 + \lambda_s \frac{1 - \text{SSIM}_{aa'}(p)}{2} \right). \quad (7)$$

Here $|V_d|$ represents the number of successfully transformed image point p , $I_{a'}$ is the reconstructed image (by obtained pose $[\mathbf{R}_{a'}^b, \mathbf{t}_{a'}^b]$, next frame I_b and depth D_b in Figure 3) of image I_a . The loss based on SSIM describes the structural similarity between image $I_{a'}$ and I_a . λ_s is a hyper-parameter.

The smooth loss item L_s is used to punish discontinuity of depth in local region and formulated by

$$L_s = \sum_p \left(e^{-\nabla I(p)} \cdot \nabla D(p) \right)^2 \quad (8)$$

with ∇ as a gradient operation.

The geometric consistency constraint L_{GC} [27] is added to get better scale consistency. It is calculated by following equation:

$$D_{\text{diff}}(p) = \frac{|D_{a'}(p) - D_a(p)|}{D_{a'}(p) + D_a(p)}, \quad (9)$$

$$L_{GC} = \frac{1}{|V_d|} \sum_{p \in V_d} D_{\text{diff}}(p). \quad (10)$$

$D_{a'}$ is the reconstructed depth of I_a and obtained by transforming D_b (the depth of next frame I_b) according to the pose $[\mathbf{R}_{a'}^b, \mathbf{t}_{a'}^b]$ and itself. This consistency constraint is used to optimize the original photometric loss L_p by

$$L'_p = \frac{1}{|V_d|} \sum_{p \in V} ((1 - D_{\text{diff}}) \cdot L_p(p)). \quad (11)$$

3.3. Joint Learning

In this part, the learning process to merge ground estimation with the SfM framework by those three losses L_r , L_H and L_v will be introduced in details. The whole joint learning procedure is shown in Figure 4.

With the rough ground segmentation $\Phi(I_a)$ and estimated depth D_a , the normal vector $\mathbf{n}$ of the ground plane is calculated by the Random Sample Consensus (RANSAC) method. A self-learning ground label I_{ground} is obtained from points within a valid distance to the plane. I_{ground} and $\Phi(I_a)$ are used to calculate the cross-entropy loss L_r . By using this loss, the road surface segmentation is re-learned according to the geometry information. The normal vector $\mathbf{n}$ is used to calculate the vanishing point and an ROI. Abnormal depth values in this ROI are punished through L_v and also eliminated in the calculation of L'_p . At the end, the plane photometric loss L_H is calculated with normal vector $\mathbf{n}$, ground segmentation $\Phi(I_a)$, successive frames I_a, I_b , and pose estimation $[\mathbf{R}_{a'}^b, \mathbf{t}_{a'}^b]$. This loss is used to optimize the pose model. More details about this learning procedure are described in following parts.

3.3.1. Ground Self-Learning Loss

The depth itself actually contains a lot of structural information, especially for the ground. Using this loss, the coarse ground segmentation is combined with the depth information to refine the self-learning labels for ground segmentation network so that a better segmentation result can be restored.

Here the output of the ground segmentation network is processed by the softmax layer to obtain the probability map $\Phi(I_a)$. Here we only consider a binary classification problem (i.e., ground and non-ground). Thus, a ground mask M_g can be obtained by checking the ground probability at the image coordinate (x, y) by $M_g(x, y) = \text{sign}(\Phi(I_a(x, y)) > 0.5)$. Then, for a ground point $\mathbf{p}_{xy}^c$ in camera coordinate system, we have

$$\frac{\mathbf{n}^\top}{d} \mathbf{p}_{xy}^c + 1 = 0, \quad (12)$$

$$\mathbf{K}^{-1}D_{xy}\mathbf{p}_{xy} = \mathbf{p}_{xy}^c, \quad (13)$$

where $\mathbf{n}$ is the ground unit normal vector in camera coordinate system and d is the distance from the camera to the ground. In the second equation, the homogeneous image point coordinate $\mathbf{p}_{xy}$ is transformed into camera coordinate $\mathbf{p}_{xy}^c$ by the intrinsic parameter matrix $\mathbf{K}$ and corresponding depth D_{xy} . The subscript indicates that the image coordinate of the point is (x, y) . Thereafter, using the RANSAC based least square method in the ground area (mainly on the lower half image), we can estimate

$$\frac{\mathbf{n}}{d} = \text{RANSAC}(\mathbf{K}, \left\{ \mathbf{p}_{xy}, D_{xy} \mid M_g(x, y) = 1, y > \frac{h}{2} \right\}) \quad (14)$$

with image height h .

After obtaining $\frac{\mathbf{n}}{d}$, we calculate the average offset o of fitting points from the ground plane, as below:

$$o = \frac{1}{|V_r|} \sum_{(x,y) \in V_r} \frac{|\mathbf{n}^\top \mathbf{p}_{xy}^c + d|}{\|\mathbf{n}\|_2} \quad (15)$$

$$s.t. \quad V_r = \{(x, y) \mid R(x, y) = 1\}. \quad (16)$$

In above equation, V_r is the set of selected points in the RANSAC fitting phase. A point is only selected for fitting, when we have the inlier indicator $R(i, j) = 1$. Points that are smaller than $\lambda_r \cdot o$ are selected as ground points to form the new label l_{ground} ($\mathbf{p}_{xy}^c$ and $\mathbf{p}_{xy}$ are interchangeable by Equation (13)).

$$l_{ground} = \left\{ \mathbf{p}_{xy} \mid \frac{|\mathbf{n}^\top \mathbf{p}_{xy}^c + d|}{\|\mathbf{n}\|_2} < \lambda_r \cdot o \right\}. \quad (17)$$

With such a geometric prior, the new label is used in the cross entropy (CE) loss to re-learn the ground segmentation network. Thus, we have the ground self-learning loss.

$$L_r = 0.1 \cdot e^{\lambda_c \cdot s} \cdot CE(\Phi(I_a), l_{ground}). \quad (18)$$

Due to the inaccuracy of the initial ground hypothesis, we slowly increase the weight of CE by the training step number s . In previous section, we are known that the original mutual information loss L_i of the unsupervised method is learned through shallow semantics. Since the ground self-learning loss L_r is learned through geometric information, combining those two losses can make their learning complementary to each other. Therefore, better ground segmentation results can be obtained (shown in the right of Figure 1).

3.3.2. Plane Photometric Loss

Dynamic objects can cause problem for pose estimation. In turn, static objects are more valuable in estimating ego-poses. In the image, the ground can be considered as a huge static object and occupies a large image proportion. Thus, it has many good features, such as corner points and lane lines. In this approach, we propose a plane photometric loss which reconstructs the ground plane from one frame into another. For a certain plane point, the pixel coordinate transform between two frames I_a and I_b is given by

$$\mathbf{p}^b \sim \mathbf{K}^{-1}(\mathbf{R}_a^b - \frac{\mathbf{t}_a^b \mathbf{n}^\top}{d}) \mathbf{K} \mathbf{p}^a, \quad (19)$$

where $\mathbf{R}_a^b$ and $\mathbf{t}_a^b$ are inter-frame rotation and translation from I_a to I_b , which are predicted by the pose network. $\mathbf{n}$ is the ground normal vector and d is the height of the camera. We have obtained $\frac{\mathbf{n}}{d}$ using the RANSAC method. For more details, please see Appendix A.

Then, the plane photometric loss is defined as

$$L_H = \frac{1}{|V_H|} \sum_{p \in V_H} (\|I_a(p) - I_{a'}(p)\|_1 \cdot f(p)) \quad (20)$$

with $f(p)$ as the probability of point p on the ground and obtained from the ground segmentation network. V_H is the set of ground points which are successfully transformed between I_a and I_b . Through this loss, the pose learning is strengthened through the static ground. Therefore, the ground segmentation results can help optimize the original SfM framework through the pose model. But this is not the case for dynamic objects, which leads to our third loss, the depth abnormal punishment loss.

3.3.3. Depth Abnormal Punishment Loss

The self-supervised depth estimation method mainly relies on the position change between frames to estimate the depth information of the scene. For dynamic objects near the vanishing point in the image, their position change between frames is very small and thus can be mistaken for the same infinite as the vanish point. It is very common in the vehicle-following scene in the transportation task. However, since the ground normal direction has been obtained, it can be used to calculate out the ground horizon by

$$\mathbf{K}^{-\top} \frac{\mathbf{n}}{d} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad (21)$$

$$\frac{a}{c}x + \frac{b}{c}y + 1 = 0. \quad (22)$$

Equation (22) represents the ground horizon. For details about the proof of this part, please see the Appendix B.

Assuming that (x_v, y_v) is the upper marginal center of the region of interest and r_w, r_h are the width and height, $(x_v - \frac{r_w}{2}, y_v) \cdot (x_v + \frac{r_w}{2}, y_v + r_h)$ are the corners in the diagonal direction of the region. In the early stage of training, since the normal vector is not stable, (x_v, y_v) is initialized by the image center point $\frac{w}{2}, \frac{h}{2}$ and w, h are the width and height of the image. At the end of training, (x_v, y_v) is chosen as the vanishing point in the same direction of the road and calculated by Equation (22). We obtain

$$y_v = (1 - \frac{ax_v}{c}) \frac{c}{b}. \quad (23)$$

Here we simply choose $x_v = \frac{w}{2}$, which is valid in our datasets. In the region of interest, the abnormal points produce relatively sharp increases in the depth, which is more obvious in the lateral direction of the image (e.g., comparing a vehicle with its nearby road surface). The set of abnormal points V_{lv} can thus be found through the depth prediction as below:

$$V_{lv} = \left\{ (x, y) \left| D_{xy} > \frac{\lambda_{lv}}{r_w} \sum_{x'=x_v-\frac{r_w}{2}}^{x_v+\frac{r_w}{2}} D_{x'y} \right. \right\} \quad (24)$$

and λ_{lv} is a weight hyper-parameter. For such anomalies, they are removed from the original point set V_d when calculating the photometric loss, expressed as

$$L_p^v = \frac{1}{|V_d - V_{lv}|} \sum_{p \in V_d / V_{lv}} \left((1 - D_{diff}) \cdot L_p(p) \right). \quad (25)$$

Since L_p was replaced by L_p^v , loss L_d in Equation (6) is changed in the second stage, and this new loss is called L_d' . At the same time, the abnormalities are punished in the loss

$$L_v = \frac{1}{|V_{lv}|} \sum_{(x,y) \in V_{lv}} |D_{xy} - \frac{\lambda_{lv}}{r_w} \sum_{x'=x_v-\frac{r_w}{2}}^{x_v+\frac{r_w}{2}} D_{x'y}|. \quad (26)$$

By the understanding of the scene structure like vanishing point and the horizon lines, we can reduce the natural contradiction between the assumption of static objects between frames. In this way, the scene structure helps learning the SfM framework.

3.4. Entire Learning and Inference Framework

Our entire learning and inference framework is shown in Figure 5. The loss L_i and L_d are used in the first learning stage (described in Sections 3.1 and 3.2) while loss L_v , L_r and L_H are added in stage two. Additionally, the normal vector is used to improve the loss L_d , forming the modified loss L_d' (presented in Section 3.3). In the inference stage, as shown by the dotted line in Figure 5, the network predicts the scene depth, ego-motion, and ground segmentation. Based on the results of ground segmentation and depth estimation, the ground normal vector is calculated using Equation (14). Through these methods, a powerful scene structure estimator is constructed, and its sub-tasks promote each other through our proposed losses.

Figure 5. An overview of our entire learning and inference framework. Detailed description about the framework can be seen in the text.

4. Experiment

4.1. Dataset

In the research field of autonomous driving, the KITTI data set (<http://www.cvlibs.net/datasets/kitti/index.php>) is currently one of the largest evaluation datasets for computer vision algorithms such as stereo imaging, optical flow, visual odometry, 3D object detection and tracking, depth and pose estimation by vehicle-mounted cameras and lidar. In our experiments, we use the KITTI raw data and KITTI odometry data. The KITTI raw data is composed of 78 short sequences of about 45,000 images. We follow the protocol of Eigen's Split [20] same as in works [9–13,27,38,39] and divide these 78 sequences into two subsets: 697 images for testing, and the rest are for training and validation.

The Cityscapes (<https://www.cityscapes-dataset.com>) is a large-scale benchmark for segmentation tasks of traffic scene. It is composed of 5488 short sequences in 50 cities, about 70,000 images, of which 5000 are finely marked and 20,000 are roughly marked for segmentation task. Although Cityscapes is unsuitable for evaluating the estimation performance of depth, pose and ground normal vector, it contains video streams. Thus, it can be used in the unsupervised pre-training process like in works [12,13,27]. Moreover, the Cityscapes dataset provides annotations for the ground segmentation which is not available in the KITTI dataset. Hence, we use Cityscapes (i.e., the official validation set of 500 marked images) for evaluating our ground segmentation subnetwork. Since we only focus on the binary ground segmentation task, we also make adaption of the official evaluation tool in our experiments.

4.2. Implementation Details

4.2.1. Training Configuration

The hyper-parameters used in the training are as follows. In the first stage, for the coefficients in loss L_d (Equation (6)), we follow the work [27,31], so that the experimental comparison with them should be fair. Thus, coefficients α , β , γ are respectively set to 1.0, 0.1, 0.5, and $\lambda_s = 0.85$. In the second stage, the existing hyper-parameters in the first stage remain the same, and we set $\lambda_r = 0.3$, $\lambda_c = \frac{\ln 10}{50000}$, $\lambda_{lv} = 2$, which are searched through experiments on the validation set. Regarding the ROI selection, according to the statistics of used images, we found that an area near the vanishing point with a size of $\frac{w}{3} \times \frac{h}{4}$ can well handle the abnormal depth values (h and w are respectively the height and width of the image). Because we use a unified image size of 832×256 , we set $r_w = 277$ and $r_h = 64$. In the training process, we randomly select 1000 batches as an epoch. In the first training stage we train 200,000 iterations which is the same in Bian et al. [27] while in the second training stage we train 80k iterations. This configuration is based on experiments on the validation set. In the second phase, since the ground segmentation is not accurate enough in the early period of training, the location of the vanishing point calculated by the normal vector fluctuates too much, which will damage the depth estimation. Hence, the upper marginal center of ROI is approximated by the image center in the first 50,000, but in the later period when the road surface estimation becomes gradually stable and accurate, the calculation of vanishing point also becomes more accurate. Therefore, in the last 30,000, the vanishing point is calculated by the ground normal vector. The learning rate is set to 10^{-4} in the first stage and reduces to 5×10^{-5} in the second stage in order to make the learning more stable. The batch size is set to 4.

4.2.2. Network Structure

For the depth and pose estimation network, we adopt the DispRes-Net [27], and PoseNet [27] as backbone. These backbone structures are the same as in works [13,27], so that the subsequent experimental comparison should be fair. The ground segmentation network has the same structure as the depth network, with the output header divided into 5 categories due to the reason that more categories can be conducive to mutual information learning, as pointed out in [31]. Here one category is for the ground, and the other four together form the non-ground category.

4.2.3. Environment

The environment for training and testing our approach is a desktop with Intel(R) Xeon(R) CPUs E5-2667 v3 of 3.20 GHz by Intel and a memory of 64G DDR4 (Samsung, Seoul, South Korea). The GPU we use is GTX 1080 Ti (Nvidia, Santa Clara, California, United States). The network is implemented by PyTorch-1.2.0. The version of python is 3.6.8 and the operating system is Ubuntu 16.04.6 LTS.

4.3. Experimental Results

This part mainly carries out comparison experiments from four aspects, namely the depth estimation, pose estimation, ground normal vector quantitative measurement, and unsupervised ground estimation.

4.3.1. Depth Estimation Results

For testing the depth estimation network, the KITTI raw data is divided into training and test set according to the Eigen's Split [20] same as in related works [9,11–13]. Additionally, we also test the performance of the network with pre-training on the Cityscapes data.

The test results are given in Table 1. For the depth evaluation, the error metrics used are the absolute relative error (AbsRel), the square relative error (SqRel), the root mean square error (RMS), and the root mean square logarithmic error (RMSlog). For the accuracy, three thresholds are used for evaluation. Above metrics are widely used in depth estimation tasks to comprehensively consider errors and accuracy. For calculation details about these metrics, please refer to the Appendixes A–D. The compared methods are mainly from three groups: using point cloud as groundtruth (GT) to directly perform supervised (S) learning, using binocular camera calibration information for semi-supervised (SS) learning, and using video streams for unsupervised (US) learning. In comparison with other methods, it can be seen that whether our approach is only trained on the KITTI or on Cityscapes + KITTI, it achieves significant improvement on depth estimation in terms of unsupervised approaches. Compared with the baseline model SC-SfMlearner [27], we outperform it in all depth-related error and accuracy metrics. This is due to the employment of our depth abnormal punishment loss, which uses the structural information of the scene, i.e., the normal vector, to suppress the abnormal depth. Compared with the semi-supervised methods, except for a slight lag in the absolute relative error (AbsRel), our approach (trained on CS + K) surpasses the best method [25] on other metrics. In comparison with supervised depth estimation methods, our approach achieves the second place, only with a minor gap to the top method [22]. For a more intuitive impression, we select some test examples as qualitative results shown in Figure 6. However, due to the fact that the lidar cloud is very sparse while the evaluated method generates a dense depth map, a direct qualitative comparison will not be intuitive. Therefore, we resort to the depth completion method by DeepLidar [40]. This method uses an encoder-decoder structure to effectively fuse the dense color image and sparse lidar points. It also uses a network to estimate surface normals as the intermediate representation for dense depth map completion. It is one of the top performed methods in the KITTI depth completion benchmark for lidar data. Due to the lack of point cloud data in the upper part of the image, the corresponding completion cannot be performed. Thus, we truncate this part. Among the compared methods, Zhou et al. [9] were the first to use monocular image for unsupervised depth estimation. In order to solve the problem of dynamic objects, they use explainability to mask dynamic objects. Wang et al. [39] use a differentiable implementation for direct visual odometry, along with a depth normalization strategy. The CC [13] method uses four sub-networks and uses a training method similar to the EM algorithm. Compared to these methods, the clearer outline and details in estimated depth map are more clear, this is due to the geometric information such as ground surface and scene structure used in our network learning and demonstrate the effectiveness of our approach.

Table 1. The result of depth estimation. K represents the use of KITTI raw data for training while CS + K represents the use of both the Cityscapes and the KITTI raw data. B represents the training with binocular images. D represents using depth groundtruth as supervision signal, and M is for monocular video. S represents the supervised method, SS represents semi-supervised method and US represents unsupervised method. For the error, the lower is better and for the accuracy, the higher is better. The details of the indicators can be seen in Appendix D.

Learning	Method	Datasets	Error ↓				Accuracy ↑		
			AbsRel	SqRel	RMS	RMSlog	<1.25	<1.25 ²	<1.25 ³
S	Eigen et al. [20]	K (D)	0.203	1.548	6.307	0.282	0.702	0.890	0.958
	Liu et al. [41]	K (D)	0.202	1.161	6.523	0.275	0.678	0.895	0.965
	Kuznetsov et al. [22]	K (B + D)	0.113	0.741	4.621	0.189	0.862	0.960	0.986
SS	Garg et al. [42]	K (B)	0.152	1.226	5.849	0.246	0.784	0.921	0.967
	Zhan et al. [26]	K (B)	0.144	1.391	5.869	0.241	0.803	0.928	0.969
	Godard et al. [25]	K (B)	0.148	1.344	5.927	0.247	0.803	0.922	0.964
	Godard et al. [25]	CS + K (B)	0.124	1.076	5.311	0.219	0.847	0.942	0.973
US	Zhou et al. [9]	K (M)	0.208	1.768	6.856	0.283	0.678	0.885	0.957
	Yang et al. [38]	K (M)	0.182	1.481	6.501	0.267	0.725	0.906	0.963
	Mahjourian et al. [10]	K (M)	0.163	1.240	6.220	0.250	0.762	0.916	0.968
	Wang et al. [39]	K (M)	0.151	1.257	5.583	0.228	0.810	0.936	0.974
	Geonet-VGG [11]	K (M)	0.164	1.303	6.090	0.247	0.765	0.919	0.968
	Geonet-Resnet [11]	K (M)	0.155	1.296	5.857	0.233	0.793	0.931	0.973
	DF-Net [12]	K (M)	0.150	1.124	5.507	0.223	0.806	0.933	0.973
	CC [13]	K (M)	0.140	1.070	5.326	0.217	0.826	0.941	0.975
	SC-SfMLearner [27]	K (M)	0.137	1.089	5.439	0.217	0.830	0.942	0.975
	ours	K (M)	0.135	1.006	5.336	0.212	0.833	0.944	0.977
US	Zhou et al. [9]	CS + K (M)	0.198	1.836	6.565	0.275	0.718	0.901	0.960
	Yang et al. [38]	CS + K (M)	0.165	1.360	6.641	0.248	0.750	0.914	0.969
	Mahjourian et al. [10]	CS + K (M)	0.159	1.231	5.912	0.243	0.784	0.923	0.970
	Wang et al. [39]	CS + K (M)	0.148	1.187	5.496	0.226	0.812	0.938	0.975
	Geonet-Resnet [11]	CS + K (M)	0.153	1.328	5.737	0.232	0.802	0.934	0.972
	DF-Net [12]	CS + K (M)	0.146	1.182	5.215	0.213	0.818	0.943	0.978
	CC [13]	CS + K (M)	0.139	1.032	5.199	0.213	0.827	0.943	0.977
	SC-SfMLearner [27]	CS + K (M)	0.128	1.047	5.234	0.208	0.846	0.947	0.976
	ours	CS + K (M)	0.126	0.943	5.084	0.203	0.849	0.949	0.978

Figure 6. Qualitative results of scene depth estimation on KITTI raw dataset. Compared to other methods, our method has clearer outline and more details in estimated depth map. This can be credited to the use of geometric information in our method.

4.3.2. Ego-Pose Estimation Results

In this part we conduct the experiment on the KITTI odometry dataset [18]. The model is trained and verified on the sequence 00–08 and tested on sequence 09–10. For clarity, the sequence 09 and 10 are respectively renamed as test-1 and test-2 in Figure 7 and Table 2. The experimental results are shown in Table 2. The Oriented FAST and Rotated BRIEF-Simultaneous Localization and Mapping (ORB-SLAM) [43] system (without loop closure) is reported as a reference. It can be seen that our method achieves a certain improvement on the pose estimation in terms of unsupervised methods, especially compared with baseline model SC-SfMlearner [27]. This is because our plane photometric loss improves the weight of static objects in the optimization process. Thanks to the static ground features, the pose model is better optimized, which shows that the understanding of the scene geometry does help the optimization of the model. However, compared with the traditional geometry based method, i.e., the ORB-SLAM, there is still a gap for deep learning based methods in pose estimation. A qualitative comparison result is given in Figure 7, from which it can also be seen that compared to other unsupervised methods, our estimated trajectory is relatively closer to the GT. Especially compared with the baseline model SC-SfMlearner [27], in both sequence test-1 and test-2, our accuracy has been significantly improved. This is also consistent with the results in Table 2.

Table 2. The estimation results on KITTI odometry dataset. t_{err} is average translational drift error. r_{err} is average rotational drift error. The ORB-SLAM is a traditional visual SLAM method and is used as a reference. The best results for methods based on unsupervised learning is highlighted.

Methods	Test-1 (Sequence 09)		Test-2 (Sequence 10)	
	$t_{err}(\%)$	$r_{err}(^{\circ}/100m)$	$t_{err}(\%)$	$r_{err}(^{\circ}/100m)$
ORB-SLAM [43]	15.30	0.26	3.68	0.48
Zhou et al. [9]	17.84	6.78	37.91	17.78
Zhan et al. [26]	11.93	3.91	12.45	3.46
SC-SfMlearner [27]	11.2	3.35	10.1	4.96
ours	9.36	2.61	10.25	3.84

Figure 7. Qualitative results on the KITTI odometry test set. (a) is the result on test-1, (b) is the result on test-2. In comparison with unsupervised methods, our estimated trajectory is relatively closer to the GT in both sequences.

4.3.3. Ground Normal Vector Estimation Results

In this experiment, we use exactly the same dataset as for the depth estimation model. There is only a few works to estimate ground normal vectors from monocular video. We compare the existing works with our approach and report the result in Table 3. We surpass the hidden markov model (HMM) method [44] by about 0.9° . With training on Cityscapes+KITTI, the error is further reduced by 0.2° . However, there is still a gap between our method and the supervised GroundNet [15]. The GroundNet requires labels for the ground, depth and normal vector, while we only rely on continuous video streams. We also give the error distribution of our estimated normal vector on the test set in Figure 8a,b. It can be seen that errors are mainly concentrated around 3° . In order to perceive the estimation result of normal vector more intuitively, we project the normal vector into the original image, as shown in Figure 9a. Since the error is small, it is difficult to detect the change especially in the pitch angle, which is yet important in the field of automated driving. Therefore, we transform the image to a bird's eye view (BEV). Through the parallelism of the lane lines, we can intuitively judge the quality of the pitch angle. The result is shown in Figure 9. It can be seen that the deviation of parallelism by our approach is relatively small. Note that the reference value (Figure 9b) directly calculated by the given extrinsic parameters is sometimes inaccurate, which implies that our actual error is likely to be smaller than what we reported in Table 3. In the absence of labeled data, our normal vector estimation has achieved relatively good results.

Figure 8. The error distribution of predicted ground normal vector. (a) shows prediction errors of the model trained on the KITTI raw data, while in (b) the corresponding model is trained on both Cityscapes and KITTI raw data.

Figure 9. (a) is the original image. The red arrow denotes our predicted ground normal vector, and the blue one is the groundtruth calculated using extrinsic parameters. (b) is the BEV image transformed using our predicted normal vector while (c) is the BEV transformed using the groundtruth normal vector. The parallelism of the lane lines near the red circle is relatively low, indicating that the groundtruth value is sometimes not accurate enough.

Table 3. Comparison of normal vector estimation. The groundtruth value is calculated using extrinsic parameters. K means only using KITTI raw data for training, and CS + K means using Cityscapes and KITTI raw data.

Methods	Error/Deg
GroundNet [15] (Supervised)	0.70
HMM [44] (Unsupervised)	4.10
ours (K) (Unsupervised)	3.23
ours (CS + K) (Unsupervised)	3.02

4.3.4. Unsupervised Ground Segmentation Results

Due to the lack of pixel-level annotations on KITTI data, this part of experiment is conducted on the Cityscapes. Here we take all points belonging to the ground plane as positive samples and test our approach on the official validation set.

To demonstrate the effectiveness of our joint learning, we also extract the ground segmentation network from our architecture and perform supervised training on Cityscapes. The results are given in Table 4. It can be seen that even compared with the supervised method, we are still ahead of 9% in terms of the IOU accuracy. This is because our mutual information loss allows the network to learn better feature representations and improve the generalization ability of the network. For the simple supervised method, the representation learning is not sufficient, so the result is not as good as unsupervised methods with geometric correction. This also proves that the learning of semantic information can be promoted by adding geometric prior.

Table 4. Test results for ground segmentation using our network structure with or without supervision.

Methods	IOU
Supervision	0.74
Unsupervised	0.83

4.4. Ablation Experiment

Here we give a deep exploration on the effectiveness of those three losses used in the second training phase by ablation studies. Since the plane photometric loss (PPL) needs the ground segmentation result, which strongly depends on the ground self-learning loss (GSFL), we combine both losses together as GSFL + PPL. The depth abnormal punishment loss here is abbreviated as APL. We compare the performance of GSFL + PPL and APL, which are separately integrated with our approach. The depth estimation result is shown in Table 5, the improvement of APL is very obvious, while the effect of GSFL + PPL is relatively weak. The pose estimation result is shown in Table 6, we can notice that GSFL + APL does improve the estimation of the ego-motion, while the improvement by APL is not so obvious. This is consistent with our expectation. We expect that GSFL + PPL can be used to enhance the optimization of static objects in ego-motion estimation, and the abnormal depth can be improved through APL. The result of ground segmentation is given in Table 7 and shows that the effect of GSFL is very obvious with the IOU accuracy increased by 35%. This fully illustrates the importance of geometric prior information. We can observe this improvement qualitatively in the right part of Figure 1, in which the shadow area is also detected by adding geometric prior constraints through GSFL.

Table 5. The depth ablation experiments are carried out on the KITTI raw data. GSF, PPL, and APL respectively represent the ground self-learning loss, the plane photometric loss and the depth abnormal punishment loss.

Methods	Datasets	Error ↓				Accuracy ↑		
		AbsRel	SqRel	RMS	RMSlog	<1.25	<1.25 ²	<1.25 ³
Basic	K	0.137	1.091	5.441	0.217	0.830	0.942	0.975
Basic + GSFL + PPL	K	0.136	1.103	5.417	0.215	0.835	0.944	0.976
Basic + GSFL + PPL + APL	K	0.135	1.006	5.336	0.212	0.833	0.944	0.977

Table 6. The ego-motion estimation ablation experiments are carried out on the KITTI odometry dataset. GSF, PPL, and APL respectively represent the ground self-learning loss, the plane photometric loss and the depth abnormal punishment loss.

Methods	Test-01 (Sequence 09)		Test-02 (Sequence 10)	
	$t_{err}(\%)$	$r_{err}(^{\circ}/100m)$	$t_{err}(\%)$	$r_{err}(^{\circ}/100m)$
Basic(K)	11.24	3.34	10.07	4.91
Basic + GSFL + PPL(K)	9.34	2.63	9.51	3.97
Basic + GSFL + PPL + APL(K)	9.36	2.61	10.14	3.84

Table 7. Ground segmentation results with or without the GSFL loss.

Methods	IOU
Basic	0.48
Basic + (GSFL)	0.83

5. Conclusions

Estimation of scene structures (including scene depth, ego-pose, ground normal vector and ground segmentation) by a camera sensor is a crucial task for automated driving and robotics. In supervised methods, learning scene structures often requires groundtruth labels obtained by expensive sensors such as lidar, which limits their application. In existing unsupervised methods, estimation of the scene depth can also be affected by dynamic objects which do not meet the static assumption. In order to alleviate these problems, we propose a completely unsupervised learning framework for scene structure estimation by monocular camera. In this approach, we innovatively proposed three losses in

the joint learning process. Through these three losses, the ground segmentation results can be corrected by the estimated geometric prior while the depth estimation is improved by reducing the error of transformed ground plane. And the impact of dynamic objects can be suppressed using the estimated ground normal vector. Experimental results on the KITTI and Cityscape datasets demonstrate that the depth and pose estimation results have been significantly improved by our approach. We also achieve better ground segmentation and normal vector estimation results in terms of unsupervised learning methods. In the future, we will continue to explore how to combine more scene structure priors to further improve the framework.

Author Contributions: Conceptualization: L.X., Y.W., Y.H., J.Z., W.T.; Supervision: L.X., J.Z., W.T.; Writing—review: Y.W., Y.H., W.T.; Experiments: Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China under Grant 2016YFB0100901.

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

Appendix A. Homography Transformation

Given two consecutive image frames, a point $\mathbf{p}^c$ on the first frame within the camera coordinate system applies the camera imaging model

$$\mathbf{p}_1 = \mathbf{K}\mathbf{p}_1^c \quad (\text{A1})$$

with intrinsic matrix $\mathbf{K}$. Here homogeneous image coordinates $\mathbf{p}_1$ are used. The motion between the first frame and the second frame is represented by $[\mathbf{R}, \mathbf{t}]$, which denote the rotation and translation, respectively. Then, the point $\mathbf{p}_1^c$ is transformed into the camera coordinate system of the second frame, satisfying following equation

$$\mathbf{p}_2 = \mathbf{K}(\mathbf{R}\mathbf{p}_1^c + \mathbf{t}). \quad (\text{A2})$$

Considering point $\mathbf{p}_1^c$ on the ground plane, it satisfies equation

$$\mathbf{n}^\top \mathbf{p}_1^c + d = 0 \quad (\text{A3})$$

with the camera height d and ground normal vector $\mathbf{n}$. The above equation can be further modified as

$$-\frac{\mathbf{n}^\top \mathbf{p}_1^c}{d} = 1, \quad (\text{A4})$$

$$\mathbf{t} = -\frac{\mathbf{t}\mathbf{n}^\top \mathbf{p}_1^c}{d}. \quad (\text{A5})$$

Introducing above equation into Equation (A2), we obtain

$$\mathbf{p}_2 = \mathbf{K}(\mathbf{R}\mathbf{p}_1^c - \frac{\mathbf{t}\mathbf{n}^\top \mathbf{p}_1^c}{d}), \quad (\text{A6})$$

$$\mathbf{p}_2 = \mathbf{K}(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^\top}{d})\mathbf{p}_1^c. \quad (\text{A7})$$

Introducing Equation (A1) into above equation, we can get

$$\mathbf{p}_2 = \mathbf{K}(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^\top}{d})\mathbf{K}^{-1}\mathbf{p}_1, \quad (\text{A8})$$

which is the formula for transforming planes between frames.

Appendix B. Calculation of the Horizon

Considering the set V of infinity points on the ground in the camera coordinate system, a point $\mathbf{q}^c \in V$ satisfies the camera imaging model

$$\mathbf{p} = \mathbf{K}\mathbf{q}^c \quad (\text{A9})$$

with homogeneous image coordinate $\mathbf{p}$, which indicates the infinity point on the ground plane. Since the line between $\mathbf{q}^c$ and the camera origin is perpendicular to $\mathbf{n}$, we have

$$\mathbf{q}^{c\top} \mathbf{n} = 0. \quad (\text{A10})$$

Introducing Equation (A9) into above equation, we have

$$\mathbf{p}^\top \mathbf{K}^{-\top} \mathbf{n} = 0. \quad (\text{A11})$$

In this paper, there is an additional coefficient on the left side of Equation (A11), but it does not affect the result, because the right side is 0. Since $\mathbf{p}$ is the coordinate of the infinity point on the ground, all such points constitute the horizon in the image. Thus, the horizon equation is

$$\mathbf{K}^{-\top} \mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad (\text{A12})$$

where the right side of above equation denotes the parameters of the horizon line.

Thus, the horizon equation can be formulated with image point $\mathbf{p}$ by

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (\text{A13})$$

$$\frac{a}{c}x + \frac{b}{c}y + 1 = 0. \quad (\text{A14})$$

Appendix C. Calculation of the Loss Based on SSIM

Given two small patches $\mathbf{x}$ and $\mathbf{y}$, the SSIM mainly considers three aspects, namely the luminance comparison $l(\mathbf{x}, \mathbf{y})$, contrast comparison $c(\mathbf{x}, \mathbf{y})$, and structure comparison $s(\mathbf{x}, \mathbf{y})$. The similarity for two patches is represented as

$$S(\mathbf{x}, \mathbf{y}) = l(\mathbf{x}, \mathbf{y}) \cdot c(\mathbf{x}, \mathbf{y}) \cdot s(\mathbf{x}, \mathbf{y}). \quad (\text{A15})$$

The luminance comparison is defined as follows

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (\text{A16})$$

where μ_x, μ_y respectively denote the mean of pixel value for small patch $\mathbf{x}$ and $\mathbf{y}$, which are of 5×5 in our work. And C_1 is a constant and set to 0.02 as in the work [37].

The contrast comparison is defined as

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (\text{A17})$$

where σ_x, σ_y respectively denote the standard deviation of pixel values in patch $\mathbf{x}$ and $\mathbf{y}$. The constant C_2 is set to 0.06, same as in the work [37].

The structural comparison is defined as

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \quad (\text{A18})$$

and the σ_{xy} is

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x) (y_i - \mu_y). \quad (\text{A19})$$

C_3 is set to $\frac{C_2}{2}$. N denotes the pixel number of the patch. Since the photometric value subtracts the mean and is divided by the standard deviation, the structural comparison excludes the effects of luminance and contrast.

In summary, the SSIM for two patches can be expressed as

$$\text{SSIM}(\mathbf{x}_j, \mathbf{y}_j) = \frac{(2\mu_{x_j}\mu_{y_j} + C_1)(\sigma_{x_j y_j} + C_2)}{(\mu_{x_j}^2 + \mu_{y_j}^2 + C_1)(\sigma_{x_j}^2 + \sigma_{y_j}^2 + C_2)}. \quad (\text{A20})$$

So the loss based on SSIM in the paper can be expressed as

$$\text{SSIM}_{aa'}(p) = \text{SSIM}(I_a(p), I_{a'}(p)). \quad (\text{A21})$$

The $I_a(p), I_{a'}(p)$ here specifically refer to the patches centered on pixel p . Based on the above description, SSIM considers more comprehensive information than other simple image difference measurements. It is more in line with the requirements in our method.

Appendix D. The Metrics Used for Depth Estimation

For depth estimation, we denote D_i^* as the ground truth and D_i as the estimated value. N is the number of involved points. The absolute relative error is defined as

$$\text{AbsRel} = \frac{1}{N} \sum_{i=1}^N \frac{|D_i - D_i^*|}{D_i^*}. \quad (\text{A22})$$

The square relative error is

$$\text{SqRel} = \frac{1}{N} \sum_{i=1}^N \frac{|D_i - D_i^*|^2}{D_i^*}. \quad (\text{A23})$$

The root mean square error is

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N |D_i - D_i^*|^2}. \quad (\text{A24})$$

The root mean square logarithmic error is

$$\log \text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N |\lg D_i - \lg D_i^*|^2}. \quad (\text{A25})$$

The accuracy is defined as

$$\frac{1}{N} \sum_i^N \text{sign} \left(\max \left(\frac{D_i}{D_i^*}, \frac{D_i^*}{D_i} \right) < T \right). \quad (\text{A26})$$

with threshold T . Three different thresholds ($1.25, 1.25^2, 1.25^3$) are used in the accuracy metric as in works [9–13,27,38,39].

References

1. Khan, S.M.; Shah, M. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*; Springer: Berlin, Heidelberg, 2006; pp. 133–146.
2. Chen, X.; Kundu, K.; Zhu, Y.; Berneshawi, A.G.; Ma, H.; Fidler, S.; Urtasun, R. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; pp. 424–432.
3. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3d proposal generation and object detection from view aggregation. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 1–5 October 2018; pp. 1–8.
4. Koščeká, J.; Zhang, W. Extraction, matching, and pose recovery based on dominant rectangular structures. *Comput. Vis. Image Underst.* **2005**, *100*, 274–293. [[CrossRef](#)]
5. Forssén, P.E.; Lowe, D.G. Shape descriptors for maximally stable extremal regions. In *Proceedings of the 2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, 14–20 October 2007; pp. 1–8.
6. Bian, J.; Lin, W.Y.; Matsushita, Y.; Yeung, S.K.; Nguyen, T.D.; Cheng, M.M. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 4181–4190.
7. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
8. Bian, J.W.; Wu, Y.H.; Zhao, J.; Liu, Y.; Zhang, L.; Cheng, M.M.; Reid, I. An evaluation of feature matchers for fundamental matrix estimation. *arXiv* **2019**, arXiv:1908.09474.
9. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 1851–1858.
10. 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 Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5667–5675.
11. Yin, Z.; Shi, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1983–1992.
12. 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; pp. 36–53.
13. 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*, Long Beach, CA, USA, 16–20 June 2019; pp. 12240–12249.
14. McDaniel, M.W.; Nishihata, T.; Brooks, C.A.; Iagnemma, K. Ground plane identification using LIDAR in forested environments. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, 3–8 May 2010; pp. 3831–3836.
15. Man, Y.; Weng, X.; Li, X.; Kitani, K. GroundNet: Monocular Ground Plane Estimation with Geometric Consistency. In *ACM International Conference on Multimedia*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 2170–2178.

16. Bansal, A.; Russell, B.; Gupta, A. Marr revisited: 2d-3d alignment via surface normal prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 26–1 July 2016; pp. 5965–5974.
17. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2650–2658.
18. 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 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 26–July 1 2016; pp. 3213–3223.
19. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Rob. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
20. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2014; pp. 2366–2374.
21. Lin, G.; Liu, F.; Milan, A.; Shen, C.; Reid, I. Refinenet: Multi-path refinement networks for dense prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1228–1242. [[CrossRef](#)] [[PubMed](#)]
22. Kuznetsov, Y.; Stuckler, J.; Leibe, B. Semi-supervised deep learning for monocular depth map prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6647–6655.
23. Yin, Z.; Darrell, T.; Yu, F. Hierarchical discrete distribution decomposition for match density estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 6044–6053.
24. Tang, C.; Tan, P. Ba-net: Dense bundle adjustment network. *arXiv* **2018**, arXiv:1806.04807.
25. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 270–279.
26. Zhan, H.; Garg, R.; Saroj Weerasekera, C.; Li, K.; Agarwal, H.; Reid, I. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 340–349.
27. Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.M.; Reid, I. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2019; pp. 35–45.
28. Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; Sugiyama, M. Learning discrete representations via information maximizing self-augmented training. *arXiv* **2017**, arXiv:1702.08720.
29. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv* **2018**, arXiv:1808.06670.
30. Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.J.; Lipson, H. Understanding Neural Networks through Deep Visualization. *arXiv* **2015**, arXiv:1506.06579.
31. Ji, X.; Henriques, J.F.; Vedaldi, A. Invariant information clustering for unsupervised image classification and segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 9865–9874.
32. Hoiem, D.; Efros, A.A.; Hebert, M. Recovering surface layout from an image. *Int. J. Comput. Vis.* **2007**, *75*, 151–172. [[CrossRef](#)]
33. Ren, Z.; Jae Lee, Y. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 762–771.
34. Chen, W.; Xiang, D.; Deng, J. Surface normals in the wild. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 1557–1566.
35. Zhang, Y.; Song, S.; Tan, P.; Xiao, J. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 668–686.

36. Stekovic, S.; Fraundorfer, F.; Lepetit, V. General 3D Room Layout from a Single View by Render-and-Compare. *arXiv* **2020**, arXiv:2001.02149.
37. 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](#)]
38. Yang, Z.; Wang, P.; Xu, W.; Zhao, L.; Nevatia, R. Unsupervised learning of geometry with edge-aware depth-normal consistency. *arXiv* **2017**, arXiv:1711.03665.
39. Wang, C.; Miguel Buenaposada, J.; Zhu, R.; Lucey, S. Learning depth from monocular videos using direct methods. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2022–2030.
40. Qiu, J.; Cui, Z.; Zhang, Y.; Zhang, X.; Liu, S.; Zeng, B.; Pollefeys, M. DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene From Sparse LiDAR Data and Single Color Image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
41. 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.* **2015**, *38*, 2024–2039. [[CrossRef](#)] [[PubMed](#)]
42. Garg, R.; BG, V.K.; Carneiro, G.; Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*; Springer: Zurich, Switzerland, 2016; pp. 740–756.
43. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Rob.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
44. Dragon, R.; Van Gool, L. Ground plane estimation using a hidden markov model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 4026–4033.

© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).