Journal of Computers Vol. 32 No. 6, 2021, pp. 38-51
doi:10.53106/199115992021123206004

Unsupervised Learning of Depth and Ego-Motion from Continuous Monocular Images

Zhuo Wang[1*], Min Huang[1], Xiao-Long Huang[1], Fei Ma[1],
Jia-Ming Dou[2], Jian-li Lyu[3]

[1] School of Mechanical and Electrical Engineering, Beijing Information Science and Technology University,
Beijing 100192, Beijing, China
wangzhuosy@163.com, {huangmin, HXL}@bistu.edu.cn, mashengbo1990@126.com

[2]School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture,
Beijing 100044, Beijing, China
djmmjddjmmjd@163.com

[3] Ministry of labor and wages, China Railway Beijing Group Co., Ltd, Beijing 100860, Beijing, China
ljlsj@sina.com

Received 27 February 2021; Revised 24 July 2021; Accepted 15 August 2021

Abstract. In this study, the task of estimating depth is explored and also estimates continuous monocular images and optimizing and comparing two uncontrolled neural network structures namely DispNet and DispResNet, to determine a network structure that is more optimal. Photometric loss, minimal photometric loss, mask loss and smoothness loss are all components of loss functions for training depth and pose estimation neural networks. For the computation of photometric loss error caused through object motion and object occlusion on continuous images, a minimum photometric loss calculation method is proposed: the minimum value of photometric loss for each pixel point is taken, and then the mean value is computed as the minimum photometric loss, which minimizes the calculation error caused by occlusion, as well as other factors. The KITTI dataset assessment demonstrate that: the whole seven assessment parameters of depth estimation attain optimum value. Moreover, we show that our ego-motion network is able to predict camera tracks on long sequences of videos more closely than other algorithms.

Keywords: unsupervised learning, monocular image information, depth estimation, CNN

1 Introduction

Computer vision techniques tend to be applicable to estimate each pixel's depth and camera pose in a continuous image and denote a crucial aspect of vision SLAM. The depth and camera pose estimation techniques tend to be used in variety of applications, for instance, autonomous driving and crewless aerial vehicles. They denote a relevant addition to navigation technologies including GPS and inertial gauges. The success of conventional visual odometry have been determined through the application of feature matching [1-3]. Recently, there have been recorded progress in the enhancement of robustness, create more efficient job descriptions [4] and include semantic information [5]. Nevertheless, because of intrinsic deficiencies in image processing of computer, including discrete structural features, area features adaptation problems, as well as the massive influence of light fluctuations on the results, conventional methods still have considerable shortcomings.

Deep learning is capable of working approximately in low lighting conditions and between the movement of objects, as well as having stable operation in dynamic environments. Researchers have applied an in-depth learning to visual odometry. The first progress has been recorded with deep neural

[*] Corresponding Author

networks based on supervised learning. For instance, DeepVO [6], which combines convolutional neural networks, as well as recurrent neural networks to estimate the camera pose, utilizes optical flow to extract beneficial feature representations from images, solving to certain extent the problems of scale drift, as well as extraction of high-quality features. The training method of unsupervised learning, which solely requires data that are readily available, for example, images and camera gasoline matrices, has drawn the attention of more researchers to conduct study and come up with several outstanding models, e.g. SfMLearner [7], GeoNet [8], etc. DispNet [9] or DispResNet [10] are always adopted by these models to serve as the depth estimation network. The convolution operation is initially utilised by the network in minimizing the size of the input image, then the image size is amplified using a de-convolution operation, and the enlarged image is subsequently convolved to provide an estimate of the depth.

Some researchers have depicted to introduce optical flow or add some effective loss calculation functions to improve the accuracy of estimation results (see Sect. 2). However, most of them use DispNet and DispResNet as the depth estimation network structure without any optimization. This paper optimizes the optimal structure through joint training of the depth estimation and the camera pose estimation (see Sect. 3). The loss function commonly utilized in previous unsupervised training comprises of photometric loss and smoothness loss. Among these, loss of brightness has the greatest weight and most suitably provides reflection of network performance. As a result of the object motion alongside object obstruction, there is a significant error when the luminosity loss is directly computed. A method of calculating the minimum photometric loss that is capable of minimizing the number of defective pixels is proposed in this paper (see Sect. 4). Several assessments on the KITTI dataset [11] reveal that both depth estimation and camera pose evaluation outperform previous methods, demonstrating the network structure's superiority, as well as the minimum photometric loss calculation method described in this paper (see Sect. 5).

The major contributions of this paper are listed below:

(1) The DispNet and DispResNet are optimized, as well as proposing the optimal network structure and to prove the network superiority, ablation experiments are conducted.

(2) A minimum brightness/luminosity loss is proposed that tend to maximally solve the error caused by objects occlusion and objects motion when re-projecting image.

2 Related Work

The feature matching used in conventional approach is usually effective. Nevertheless, it has several challenges, for instance, the need for at least two images, the high light environment requirements, and image texture features. Researchers initially intend to introduce a deep learning to replace one or several intermediate aspects of the traditional methods, including feature extraction, feature recognition, as well as camera pose estimation. Subsequently, Eigen et al [12] predicted depth estimation information from a single image using a convolutional neural network, through the application of real spatial information measured by a distance sensor as a supervised signal, and trained a coarse-to-fine neural network in forecasting the depth of individual pixel points in a monocular image.

Considering the challenges of obtaining high-quality depth estimation label data, A completely autonomous end-to-end learning framework, SfMLearner is proposed by Zhou et al. [7], for constructing two networks that estimate image depth, as well as position of camera, with training of both networks concurrently taking place. The computation of Photometric losses is conducted through the application of image reconstruction techniques [13] in addition to the parallax smoothness forecasted combined to serve as a supervised signal. Moving objects in continuous images does not meet the prerequisite that photometric loss should be calculated in a static scene, therefore Zhou et al. [7] have designed a novel neural network that can generate masks to obscure moving objects. Nevertheless, this model generates a coarse network of masks, which provides a relatively limited network accuracy improvement and enhances the difficulty of the network training.

The GeoNet model [8] introduces the optical flow method for solving object motion between two adjacent frames. The entire model is classified into two parts: 1) a rigid structure reproducer; and 2) a non-rigid motion localiser. The rigid structure reproducer shares similarity with the structure of the SfMLearner in that it comprises of a depth estimation network and a camera pose estimation network, which result in output depth estimation and camera pose estimation results. The precision of forecast results is enhanced using the GeoNet model. Nevertheless, the introduction of the optical flow network makes the training of the network more tedious.

Absolute precision in-camera poses can be guaranteed by the UnDeepVO model [14], utilizing binocular images and monocular images for training and assessment respectively, and to some extent enhancing detection precision, however at the expense of varying camera movements in sequences of the video.

Bian et al. [15] proffer a solution to the loss of geometric consistency in scale uniformity that enables the network to use continuous video sequences as a training dataset, with leading results in assessing the KITTI dataset. This method suffers from the similar problem as all other methods: the utilization of masks to reject pixel points in the presence of object motion, object occlusion, which could cause a reduction in the total number of supervisable pixel points. This article suggests a method of calculating minimum photometric loss to minimize errors brought through object movement and occlusion and to solve this problem.

From this study, it is discovered that it is feasible to optimize the other half of these two network models', thus to determine the optimal network model, the networks are adjusted and ablation experiments are performed. For handling moving objects, recent work [7] introduces an extra motion segmentation network and [8] proposes to introduce an additional optical flow network. Even more recently [15] propose a geometry consistency constraint. Although they show significant performance improvement, there is a huge additional computational cost added into the basic framework, yet they still suffer from the occluded part issue. We reconstruct the warping results by calculating the minimum value and calculate the luminosity loss, which can solve the error caused by the object motion to the maximum extent.

3 Method

Components of our system consist of a depth estimator, as well as a camera pose estimator respectively. As depicted in Fig. 1, series of input images are taken by the pose estimation network and also predicts the 6-DoF transformations between them. To create a mask which is classified into the pose estimation network, a branch is separated from the pose estimation network. The depth estimation network uses a monocular image as an input for estimating depth map.

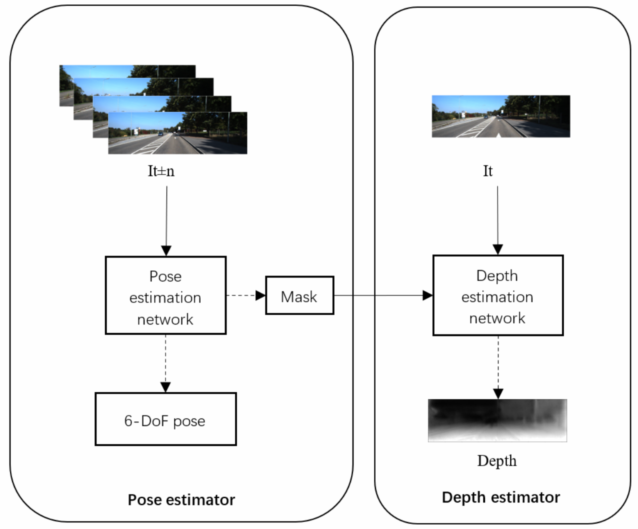


Fig. 1. Overview of our system. It consists of a depth estimator and a camera pose estimator

3.1 Depth Estimator

Using higher resolution images as input tends to considerably enhance detection accuracy and reduce errors [15]. As oppose most unsupervised depth estimation models, this paper provides direct estimation of the depth of each pixel point in the image, instead of estimating the parallax map first and subsequently taking the number of arrivals.

It was discovered that simply putting the binoculars together on the colour channel and feeding them into the neural network would affect the outcome of the prediction. Assuming the binocular images were individually fed into the neural network, the number of network parameters would increase and the training of the network would be more complicated, which is incomplete processing compared to the enhanced precision. Thus, continuous monocular image sequences are used as input to the neural network in this article.

The first half of the network comprises of alternating step-2 and step-1 convolutional layers. The step-2 convolutional layer acts as a size reduction, and the second half comprises of a step-2 deconvolutional layer and a step-1 convolutional layer, with the deconvolutional layer acting as a size-sized enlargement. Applying the ReLU activation splice depth estimation layer, for preventing the output from appearing as a Nan value, the prediction is subjected to a scaling constant a subsequently accompanied by a constant c = 0.01:

$$D_t = a * sigmoid(x) + x\,, \tag{1}$$

where D_t represents the depth map, and x represents the output of depth estimation network.

For replacing the precipitation layer on DispNet, a residual module is applied by DispRseNet which reduces detection errors and enhances detection precision. This article will adapt and experiment with both network models. The results of the optimization of the model are presented using the example of DispNet. There has been an extension of the first half of the network from 14 to 16 layers and the structure of the second half of the network has been optimized. The results are depicted in Fig. 2. The second half of the network has two operational layers. The deconvolution layer: the step size is 2, the role is to amplify the image feature size to 2 times. Depth estimation layer: Use the convolution layer' output as input for computing the depth using Equation (1).

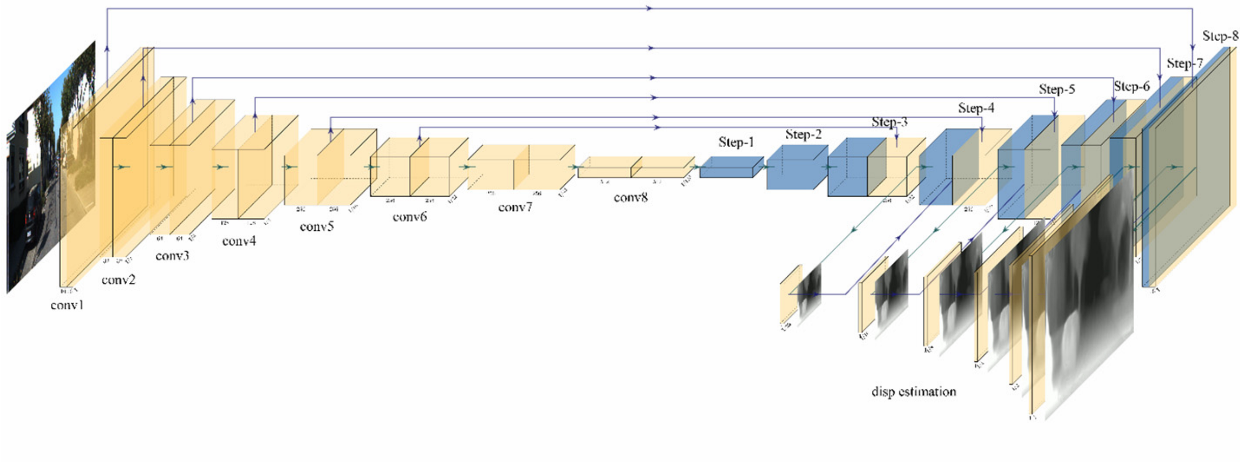


Fig. 2. Depth estimation network structure

The depth estimation layers are capable of estimating the depth of multiple size images, and more depth estimation layers can improve the network results. The input to the deconvolution layer can be either the stitching layer or the previous stage of deconvolution. Notwithstanding, the detection of the results is impacted by these two input methods to higher complexity, and following comparative experiments, the network structure depicted in Fig. 2 is found to have an optimal performance. Section 5.3 presents results of the comparative tests. Camera Pose Estimator It was discovered that simply putting the binoculars together on the colour channel and feeding them into the neural network would affect the outcome of the prediction. Supposing the binocular images are fed separately into the neural network, it would lead to an increase in network parameters number, as well as adding more complication to the network training, whose processing is incomplete in contrast to the improved accuracy. Therefore, continuous monocular image sequences are utilized as input to the neural network in this article.

3.2 Camera Pose Estimator

The camera pose is simply relevant when the camera is moving, thus minimum of two or more images are required as input for the camera pose estimator. To complement the depth estimator, a frame is determined as the current image I_t, while making the number of forward and backward changing images is equivalent to n images I_(t±n) (2n images in total) prior to and after this frame are chosen to be stitched with the current image as input to the pose estimation network, which estimates 2n pairs of poses of the current image to the before and after images. The component of each pair of poses consists of 3 translation components and 3 rotation components. Rotation angles are very non-linear and more tedious to train in contrast to translations, hence translations are evaluated separately from rotations. The functions of the image first 7 layers of the network extract and subsequently three branches are generated from this node, namely: a translation estimation branch, a rotation estimation branch and a mask generation branch.

The network structure of the translation estimation branch is similar to that of the rotation estimation branch, comprising of one 3×3 convolutional layer and two 1×1 convolutional layers. The output T_n of the translational estimation branch and the output R_n of the rotational estimation branch are spliced into the pose P_{cn}. Zhou et al [7] proposed that the output bit pose vector is scaled by a minimal constant (0.01) for the facilitation of training. The final output pose $P_n = 0.01 \times P_{cn}$.

The mask generation branch utilizes a 7-layer deconvolution layer with a step size of 2 to reverse scale the image features to the similar size as the original image.

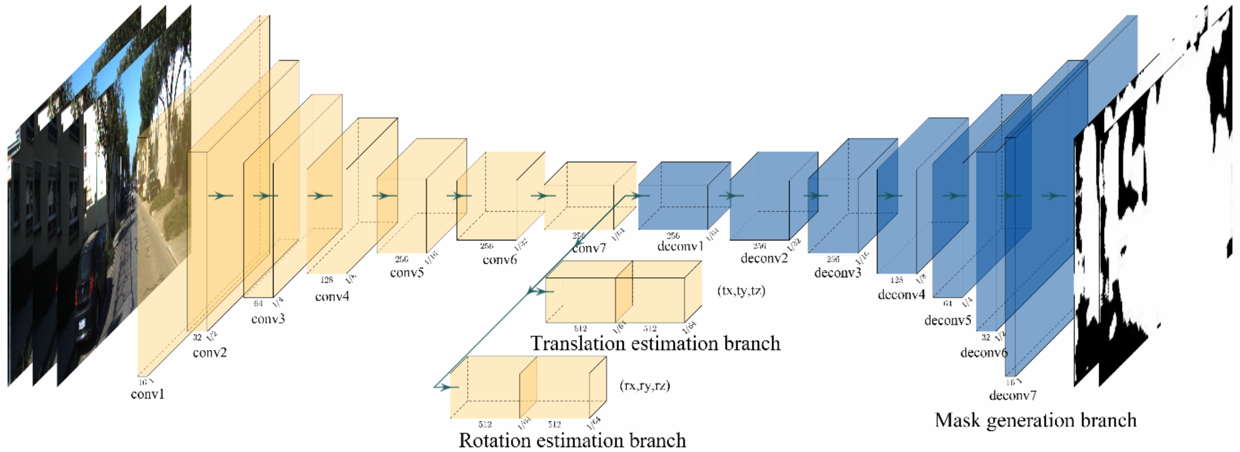


Fig. 3. Ego-motion estimation network structure

4 Unsupervised Training Loss

Components of the paper loss function include photometric loss, minimal photometric loss, mask loss, as well as loss of smoothness. In this article, the main focus is on the depth estimate result for the original image, thus only the depth estimate loss is calculated from the original size.

4.1 Photometric Loss

Photometric loss is calculated by transforming the before and after moment images into the current image utilizing information such as the estimated pose as well as the current depth and subsequently computing the difference between the two images [7-8, 14-15].

The camera internal reference matrix is referred to as K. The current image I_t is input to the depth estimation network, and the depth estimate D_t is output. The current image It along with the subsequent (or previous) image I_{t+1} (or I_{t-1}) are entered into the positional estimation network and the 6D of poses are performed. The resultant poses are subsequently transformed into a transformation matrix T in the form of flush coordinates. Converting the 2D depth map Dt to a 3D point cloud Pt

$$P_t = K^{-1} \cdot D_t \cdot M_t, \tag{2}$$

where M_t denotes the matrix of chi-square coordinates for each pixel point. Transformation of the current image 3D point cloud P_t to the subsequent (or previous) 3D point cloud P_{t+1} (or P_{t-1}) utilizing the transformation matrix T

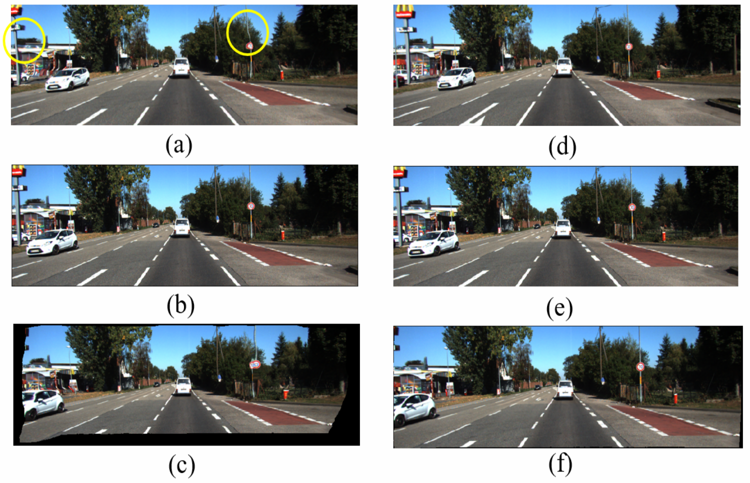
$$P_{t+1} = T \cdot P_t = T \cdot K^{-1} \cdot D_t \cdot M_t. \tag{3}$$

Determine the pixel coordinates corresponding to the 3D point cloud P_{t+1} (or P_{t-1}):

$$M_{t+1} = D_t^{\ -1} K P_{t+1} = D_t^{\ -1} T K D_t M_t. \tag{4}$$

Equation (4) describes the correspondence between each pixel point of the current image I_t and the pixel point of the image I_{t+1} (or I_{t-1}) at the following (or previous) moment. This is of course not an individual match, and there may be matches that are beyond the scope of the image. Pixel points outside the image size could be eliminated by limiting the coordinate interval and then the image I_{t+1} (or I_{t-1}) is distorted (warp) backward by spatially differentiable bilinear interpolation [17] to produce the current image $\tilde{I}_s^{rig}$, and the colour differentiation between the generated $\tilde{I}_s^{rig}$ and I_t, is the photometric loss influenced by the depth estimation map Dt and the pose conversion T. The lower the photometric loss, the more the network prediction error is low.

The value of D_t in Eq. (4) is computed from Eq. (1), where the scaling constant α in Eq. (1) has no effect on the Eq. (4) calculation and could be considered as any constant. The scaling constant is assuming to be 1 for the experiment: when the depth estimation result is in the range [0, 1], the current moment image synthesized at the final moment is depicted in Fig. 4(a), the current moment image is shown in Fig. 4(b), and the current moment image synthesized at the last moment is shown in Fig. 4(c). As readily observable, Fig. 4(c) has several pixel points at the edges that could not be composited, while Fig. 4(a) is constant with the overall current moment image, nevertheless there are distorted pixels, as revealed in the circled area. The experiment conducted by setting the constant scale to 10 is the last moment when the composite image is revealed in Fig. 4(d), the composite image of the current moment depicted in Fig. 4(e), and the current moment image last minute composite is shown in figures (f). It is readily observed that the two composite images are nearly similar to the current moment snapshot. Thus, it is essential to have a scaling constant greater than 1.



(a) (d)

(b) (e)

(c) (f)

Fig. 4. Depth estimation warpage results with different scaling constants

Potential requirements to compute photometric loss include: the scene is static, mutual occlusion does not exists, and the object surface is a Lambertian. To deal with these issues, on the one hand a robust technique for calculating picture similarity is utilized [19], which strikes a balance between describing similarity and outlier redundancy.

$$L_r = a\frac{1-SSIM(I_t, \tilde{I}_s^{rig})}{2} + (1-a)\,||\, I_t - \tilde{I}_s^{rig} \,||_1 . \tag{5}$$

From Equation (5), Photometric loss tends to be computed for each pixel, then a mask is used to subsequently eliminate outliers. The SoftMax function converts the generated mask into a matrix with only 0 and 1.0. This implies that the pixel fails to attain the requirements and that there is no computation of photometric loss. 1 indicates the feasibility of computing the photometric loss. The transformed mask is multiplied using the photometric loss to remove the non-compliant pixel points and averaging is performed to obtain the result of the final photometric loss.

$$L_1 = mean(E_m \cdot L_r). \tag{6}$$

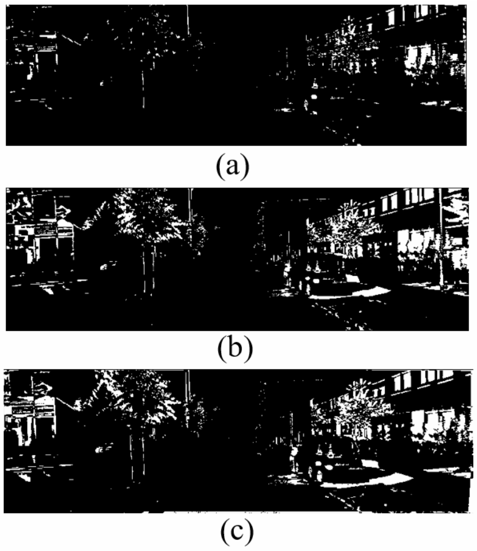
4.2 Minimum Photometric Loss

I_{t+1}, I_{t-1} denote the before and after frames of I_t. The movement of the camera results in an occluded area on the I_{t+1} image that could be visible on the I_{t-1} image. Likewise, the occluded parts of I_{t-1} may be visible on the I_{t+1} image. The warping result error because if object occlusion on I_{t+1} (or I_{t-1}) will then not appear on the warping result of I_{t-1} (or I_{t+1}). A method to compute the minimum photometric loss is proposed in this article. The method is as follows: take the lowest value of the photometric loss per pixel, then determine the average value as the minimum photometric loss, which can maximize the effect of occlusion, motion, as well as other factors. Since half of the monitoring information is lost during this computation, it is applied in combination with photometric loss with a mask.

The formula for computing the minimum photometric loss is

$$L_2 = mean(min(L_2)). \tag{7}$$

The minimum photometric loss for the depth estimation is depicted in Fig. 5(a) and the photometric loss for the I_{t+1} and I_{t-1} reverse warping results are shown in Fig. 5(b) and Fig. 5(c), where the white parts suggest the pixel points where the photometric loss exceeds 10%, apparently showing that (b)(c) has more white parts than (a) and the minimum photometric loss is capable of covering more pixel points.



(a)

(b)

(c)

Fig. 5. Pinhole camera model

4.3 Total Loss

The overall loss function is a weighted sum of photometric loss, minimum photometric loss, mask loss along with smoothness loss:

$$L = L_1 + L_2 + \lambda_3 L_3 + \lambda_4 L_4, \tag{8}$$

where L_1 represents the photometric loss, L_2 denotes the minimum photometric loss, L_3 is the mask loss, λ_3 denotes the weight of the L_3 loss, L_4 is the smoothness loss, and λ_4 serves as the weight of the L_4 loss.

Mask loss prevents the mask value form becoming all zeros through calculation of the average number of the whole elements in E_m that contain zero elements.

$$L_3 = n / N, \tag{9}$$

where: n represents the number of 0 elements in the mask and *N* denotes the number of all elements. Information in areas of low texture or homogeneous areas is not provided by photometric loss, and research has been carried out to deal with this problem by including smoothing to the depth map prior to regularised estimation and to certain level avoid “gaps” in the depth estimate. Using edge-aware smoothing loss [10], the depth estimate is inverted to acquire the parallax and then the smoothing loss is computed:

$$L_4 = \Sigma_p (e^{-\nabla I_a(p)} \cdot \nabla DS_a(p))^2, \tag{10}$$

where ∇ implies the first order derivative of the spatial direction, ensuring that the smoothness is guided through the edges of the image.

5 Experiment

This paper’s network model was implemented in the TensorFlow framework, through the application of an NVIDA RTX2080Ti graphics card, as well as an Intel Xeon E5-2678 v3 processor for training testing. The KITTI odometry dataset alongside the Cityscapes dataset [18] were chosen for training the model. Three consecutive images were selected for network training as the data for one training session. An assessment was carried out on optimization results of DepNet and DispResNet depth estimation networks. An image resolution of 832 × 256 was used, and subsequently setting the training batch to 4, and the images were randomly pre-processed using small scaling and rotating measurements. The training was conducted with the application of Adam optimizer with training parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$, the learning rate was set to 0.0001, and the network has trained a total of 200,000 times. The raw data set from the KITTI raw data set were applied in testing the depth estimation results, and the pose estimation results were assessed with sequence 9 and sequence 10 from the KITTI odometry dataset.

5.1 Results of Depth Estimator Evaluation

The detection assessment results of the two optimised DispNet and DispResNet depth estimation networks with other models on the KITTI unprocessed dataset are presented in Table 1. About seven evaluation metrics for the detection results of depth estimation: absolute relative error (AbsRel); squared relative error (SqRel); root mean squared error (RMSE); root mean squared logarithmic error (RMSlg); and accuracy with threshold values of 1.25, 1.25^2, and 1.25^3, accordingly. Smaller values for all four errors and higher values for all three precisions indicate better recognition. As shown in Table 1, According to OursD, it is indicated that the depth estimation model denotes the outcome of the model detection for the optimized DispResNet network. K and KC implies training with KITTI dataset and training with KITTI and Cityscape dataset respectively.

Table 1. Error and accuracy of depth estimation of multiple unsupervised models

Method	Error metric					Accuracy metric		
	D	AbsRel	SqRel	RMSE	RMSlg	1.25	1.252	1.253
Zhou et al. [7]	K	0.208	1.768	6.86	0.283	0.678	0.885	0.957
Zhou et al. [20]	K	0.195	1.754	6.51	0.271	0.717	0.898	0.961
Yang et al. [21]	K	0.182	1.481	6.50	0.267	0.725	0.906	0.963
GeoNet [8]	K	0.155	1.296	5.86	0.233	0.793	0.931	0.973
Ganvo [22]	K	0.150	1.141	5.45	0.216	0.808	0.939	0.975
CC [10]	K	0.140	1.070	5.33	0.217	0.826	0.941	0.975
Bian et al. [15]	K	0.137	1.089	5.44	0.217	0.830	0.942	0.975
OursD	K	0.141	1.030	5.38	0.217	0.815	0.941	0.977
OursR	K	**0.132**	**0.946**	**5.16**	**0.209**	**0.835**	**0.948**	**0.979**
Zhou et al. [7]	KC	0.198	1.836	6.57	0.275	0.718	0.901	0.960
GeoNet [8]	KC	0.149	1.060	5.57	0.226	0.796	0.935	0.975
Ganvo [22]	KC	0.138	1.155	5.41	0.232	0.820	0.939	0.976
Bian et al. [15]	KC	0.128	1.047	5.23	0.208	0.846	0.947	0.976
OursD	KC	0.131	0.989	5.11	0.209	0.822	0.944	0.976
OursR	KC	**0.124**	**0.931**	**4.98**	**0.204**	**0.851**	**0.952**	**0.982**

Fig. 6 depicts the prediction results for the optimised DispResNet network and several unsupervised depth estimation models. The depth of vehicles, trees and the overall scene were estimated successfully using series of models. Nevertheless, compare to the SfMLearner model result, the results of this study (Ours) as the figure depicted are significantly more suitable, and are close to the detection results for the GeoNet model and Bian et al. More apparent contours of trees and vehicles are shown by the SfMLearner and the GeoNet model. "Voids" indicated in the rectangular box in Fig. 6. At the same time, they will not appear in the detection results for the optimized model that this article proposed. The model proposed in this paper utilizes a superficial mask-producing branch rather than the optical flow network, with a smaller training burden than the GeoNet model. Moreover, using a 416×128 image with a batch size of 4 as training data and 200,000 training sessions on similar device, GeoNet took around 26h, and the network using this paper took about 8h, utilizing an image resolution of 832×256 image input, the training time of the network in this paper was approximately 18h.

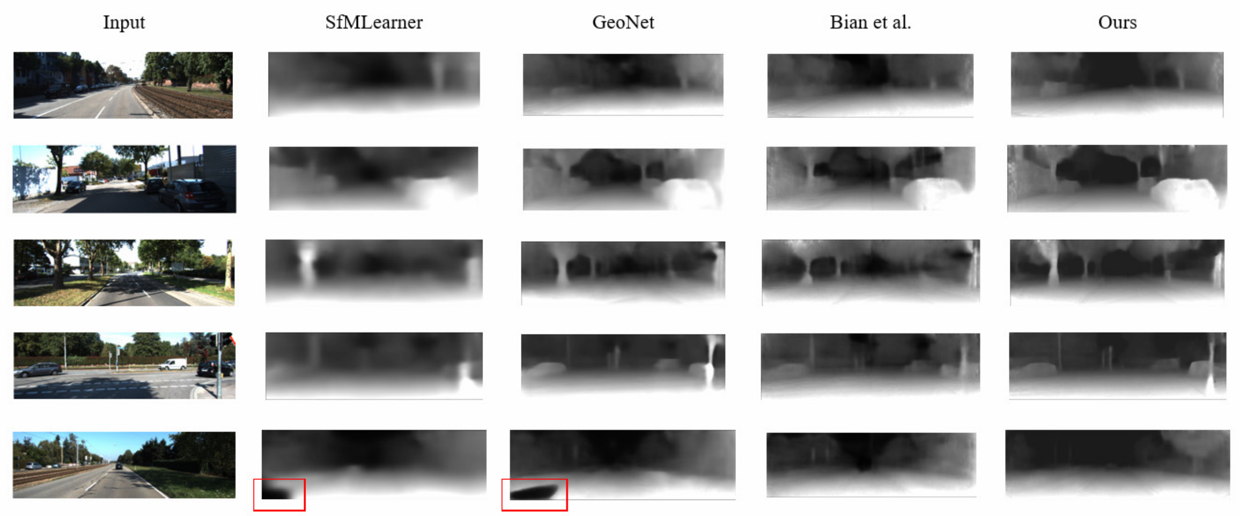


Fig. 6. Depth estimation results of several unsupervised models

As performance improves, the KITTI evaluation metric indicates the network's detection effectiveness as it is no longer intuitive in distinguishing which model detects more effectively by the naked eye. The results reveal that the detection results using the optimised DispResNet are more suitable than those of the optimised DispNet. All seven assessment results are optimal.

5.1 Results of the Camera Pose Estimator Evaluation

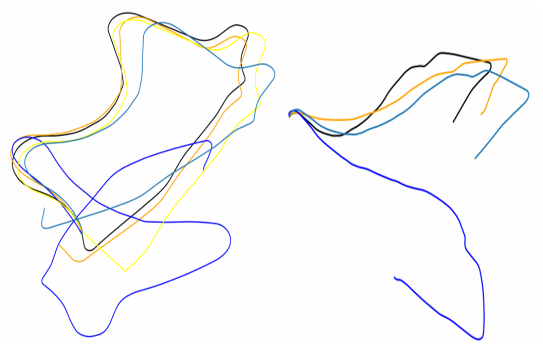
Tests were performed on sequence 9 and sequence 10 in the KITTI odometer dataset. Since dimensional differences exist between the actual images and the input images, each prediction was dimensioned and the assessment criteria was the absolute trajectory error [3]. Computation of absolute trajectory error was conducted over a 5-frame segment and the entire sequence generally displayed the mean and standard deviation. This is represented in the table as $e_1 \pm e_2$, where e_1 denotes the mean of the error and e_2 represents the standard deviation of the error. Table 2 presents the results of assessing the network in this paper against several unsupervised networks, and the detection results for the standard data and ORB-SLAM, which could optimise bit poses by applying the full trajectory information. The detection outcome for the optimised DispNet is Ours (D), and Ours (R) represents the detection result for the optimized DispResNet network. Ours(D) is equivalent to GeoNet, which is the same and more suitable in contrast to the other models, and the optimal detection result is achieved by Ours(R).

Table 2. Result of ego-motion estimation network on sequence 9 and sequence 10 of KITTI odometry dataset

Method	Seq.09	Seq.10
ORB-SLAM [3]	0.014±0.008	0.012±0.011
Zhou et al. [9]	0.021±0.017	0.020±0.015
Zhou et al. updated [9]	0.016±0.009	0.013±0.009
Bian [11]	0.016±0.007	0.015±0.009
CC [10]	0.012±0.007	0.012±0.009
GeoNet [11]	0.012±0.007	0.012±0.009
Ours(D)	0.012±0.007	0.012±0.009
Ours(R)	**0.009±0.005**	**0.008±0.006**

In this paper, five frames of pose detection results are linked into a complete trajectory, and consequently the average scale is compared with the real poses for computing the optimisation, and the unsupervised model pose detection results, ORB-SLAM pose detection results and real paths are plotted following several scale optimisations. The actual path is designated with black and orange denotes the results of this article, blue indicates GeoNet results, light blue represents SfMLearner results and yellow reveals ORB-SLAM results ORB-SLAM results in (b) are not displayed since they overlap with the actual trajectory.

Fig. 7 shows that several unsupervised detection results are subject to massive errors as a result of the accumulated errors, mainly the detection error of GeoNet since the rotation part is more considerable. As regards ORB-SLAM optimization, it has a robust back-end optimization in such a way that it has a perfectdetection results for sequence 10. Nevertheless, in Sequence 9, the detection results in this paper following restoration of scaling are more suitable compared to the other detection results, demonstrating that when combined with backend optimization in practical applications, the results of uncontrolled detection of scale ambiguity detection may equally be of certain application.



(a)Sequence 9 result (b)Sequence 10 result

Fig. 7. Pose estimation results

5.3 Ablation Study

To assess the effects of varying resolutions, various loss functions and different network structures on depth estimation, series of comparative experiments are conducted in this paper.

Effect of different network structures on detection results. The second half of the neural network could be modified in two ways: (1) the number of depth estimation layers; as well as (2) the input of the de-convolution layer. A comparison experiment is initially conducted on the number of depth estimation layers. There are changes in the number of depth estimation layers, and the inputs to the de-convolution layers are the outputs of the previous adhesive layer. With an input of 832 × 256, DispNet network structure with 4 and 6 depth estimation layers was selected. The results are presented in Table 3. A 4-layer depth estimation layer is represented as D4, while a 5-layer depth estimation layer is denoted as D5, and a 6-layer depth estimation layer is denoted as D6.

Table 3. Depth estimation of the number of layers on the impact of detection results

Model	Error metric					Accuracy metric		
	D	AbsRel	SqRel	RMSE	RMSlg	1.25	1.25^2	1.25^3
D4	K	0.149	1.112	5.66	0.221	0.802	0.933	0.976
D4	K	0.147	1.106	5.54	0.226	0.800	0.935	0.976
D6	K	**0.145**	**1.048**	**5.40**	**0.219**	**0.809**	**0.940**	**0.977**

The results reveal that the 6-layer depth estimation layer works more appropriately than the 4-layer and 5-layer depth estimation layer.

After determining that the depth estimation layer is 6 layers, the input layer of the de-convolution layer is modified. To ensure simplicity is guaranteed, Fig. 3 depicts that the other half of the network is categorized into eight parts. All input layers of the phase 3 through phase 8 de-convolution layer in DispNet are generally outputs of the splicing layer from the previous phase. The input of stage 1 denotes the output of the first half of the network, which is impossible to change. The phase 2 input denotes the output of the de-convolution layer of phase 1. In this paper, we modify the inputs of the de-convolution layers in stages 3 to 8: substitute the splice layer with the previous stage's de-convolution layer, train the modified network and assess it on the KITTI dataset, and the results are presented in Table 4. c6 indicates that the inputs of the de-convolution layers in all six stages are splice layers, c5 indicates that the inputs of the de-convolution layers in stages 4 to 8 are splice layers, c4 implies that the inputs for the de-convolution layers in steps 5 to 8 are a splice layer, C3 indicates that the input for the de-convolution layer in steps 6, 7 and 8 is a splice layer, C2 demonstrates that the input of the de-convolution layer is from Steps 7 and 8 denote a splice layer and C1 indicates that only the input into the de-convolution layer from step 8 is a splice layer.

Table 4. The influence of deconvolution layer input on detection results (DispNet)

Model	Error metric					Accuracy metric		
	D	AbsRel	SqRel	RMSE	RMSlg	1.25	1.25^2	1.25^3
C6	K	0.143	1.142	5.71	0.229	0.814	0.934	0.972
C5	K	0.141	1.063	5.61	0.225	0.814	0.936	0.973
C4	K	0.145	1.048	5.40	0.219	0.809	0.940	**0.977**
C3	K	0.143	1.077	5.54	0.221	0.808	0.940	0.976
C2	K	0.141	**1.030**	**5.38**	**0.217**	0.815	**0.941**	**0.977**
C1	K	**0.139**	1.081	5.55	0.223	**0.821**	0.939	0.975

The data in Table 4 fail to apparently illustrate the correlation between the number of splicing layers and the detection results, nevertheless it could be determined that five of the seven evaluation metrics are optimal when the de-convolutional layers of stages 7 and 8 are splicing layers, and optimal value is attained by two of the seven assessment parameters if only the input to this de-convolution layer in step 8 denotes a splicing layer. Thus, the optimal network structure is denoted as C2. Furthermore, the verification of this conclusion is conducted in DispResNet as in Table 5. Once more, result of the evaluation implied the optimal network structure is C2.

Table 5. The influence of deconvolution layer input on detection results (DispResNet)

Model	Error metric					Accuracy metric		
	D	AbsRel	SqRel	RMSE	RMSlg	1.25	1.25^2	1.25^3
C6	K	0.143	1.142	5.61	0.223	0.829	0.939	0.975
C4	K	0.133	1.083	5.44	0.215	0.833	0.945	**0.979**
C2	K	0.132	**0.946**	**5.16**	**0.209**	0.835	**0.948**	**0.979**
C1	K	**0.131**	0.999	5.32	0.218	**0.839**	0.941	0.977

Effect of minimum photometric loss on detection results. To demonstrate that the proposed minimum luminosity loss can maximise the error caused by object occlusion, the results were contrasted using the input image resolution of 832×256, with the loss function without and with the minimum luminosity loss, as presented in Table 6. The detection results of the optimized DispResNet without corresponding minimum photometric loss is denoted by NR and UR.

Table 6. Whether to use the results of the assessment of the minimum luminosity loss on the KITTI dataset

Model	Error metric					Accuracy metric		
	D	AbsRel	SqRel	RMSE	RMSlg	1.25	1.25^2	1.25^3
ND	K	0.156	1.271	5.72	0.231	0.789	0.923	0.969
UD	K	0.141	1.030	5.38	0.217	0.815	0.941	0.977
NR	K	0.149	1.107	5.64	0.229	0.812	0.934	0.969
UR	K	**0.132**	**0.946**	**5.16**	**0.209**	**0.835**	**0.948**	**0.979**

According to the data presented in Table 6, the use of the minimum photometric loss leads to a significant enhancement in detection, with a considerable reduction in error, as well as an increase in accuracy.

6 Conclusion

The results indicate that the DispResNet model has more suitable performance than DispNet and that the optimal structure of the network is such that the inputs of the last two de-convolution layers are the outputs of the respective stitching layers of the previous stage and the inputs of the remaining de-convolution. The output of their respective previous de-convolution layers denotes the input for the remaining de-convolution layers. The training of the network is with continuous images as training samples and is completely unmonitored with loss of clarity/luminosity, minimal loss of luminosity, loss of mask, as well as loss of smoothness. The minimal loss of luminosity suggested in this article will maximize the error brought about by occlusion of objects. The depth estimation results are assessed on the KITTI dataset and all assessment metrics are optimal. Evaluating the absolute trajectory error of the pose estimation for five consecutive frames of images, the former optimal results were 0.012 ± 0.007 on KITTI sequence 9 and 0.012 ± 0.009 on sequence 10, the results of the model evaluation in this article were 0.009 ± 0.005 for the KITTI sequence 9 and 0.008 ± 0.006, both recorded improvement from the results of previous studies.

Acknowledgements

This work was supported by the following projects: Beijing Municipal Science and Technology Project (Project No. Z191100001419009).

References

[1] A.-J. Davison, I.-D. Reid, N.-D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE transactions on pattern analysis and machine intelligence 29(6)(2007) 1052-1067.

[2] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: Proc. 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE, 2007.

[3] R. Mur-Artal, J.M. Montiel, J.D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE transactions on robotics 31(5)(2015) 1147-1163.

[4] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: Proc. 2011 International conference on computer vision. IEEE, 2011.

[5] M. Blaha, C. Vogel, A. Richard, T. Pock, Large-scale semantic 3d reconstruction: an adaptive multi-resolution model for multi-class volumetric labelling, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[6] S. Wang, R. Clark, H. Wen, N. Trigoni, Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks, in: Proc. 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.

[7] T. Zhou, M. Brown, N. Snavely, D.-G. Lowe, Unsupervised learning of depth and ego-motion from video, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[8] Z. Yin, J. Shi, Geonet: Unsupervised learning of dense depth, optical flow and camera pose, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[9] N. Mayer, E. Ilg, P. Hausser, D. Cremers, A. Dosovitskiy, T. Brox, A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, in: Proc. of the IEEE conference on computer vision and pattern recognition, 2016.

[10] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, M.-J. Black, Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation, in: Proc. of the IEEE conference on computer vision and pattern recognition, 2019.

[11] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, The International Journal of Robotics Research 32(11)(2013) 1231-1237.

[12] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, in: Proc. Advances in neural information processing systems, 2014.

[13] R. Garg, V.-K. Bg, G. Carneiro, I. Reid, Unsupervised CNN for single view depth estimation: Geometry to the rescue, in: Proc. European conference on computer vision. Springer, Cham, 2016.

[14] R. Li, S. Wang, Z. Long, D. Gu, Undeepvo: Monocular visual odometry through unsupervised deep learning, in: Proc. 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018.

[15] J. Bian, Z. Li, N. Wang, C. Shen, M.-M. Cheng, I. Reid, Unsupervised scale-consistent depth and ego-motion learning from monocular video, in: Proc. Advances in neural information processing systems, 2019.

[16] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, Demon: Depth and motion network for learning monocular stereo, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[17] M. Jaderberg, K. Simonyan, A. Zisserman, Spatial transformer networks, in: Proc. Advances in neural information processing systems, 2015.

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, The cityscapes dataset for semantic urban scene understanding, in: Proc. of the IEEE conference on computer vision and pattern recognition, 2016.

[19] C. Godard, A.-O. Mac, G.-J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[20] L. Zhou, J. Fang, G. Liu, Unsupervised Video Depth Estimation Based on Ego-motion and Disparity Consensus, arXiv preprint arXiv,1909.01028 (2019).

[21] Z. Yang, P. Wang, W. Xu, L. Zhao, R. Nevatia, Unsupervised learning of geometry with edge-aware depth-normal consistency, arXiv preprint arXiv:1711.03665 (2017).

[22] Y. Almalioglu, M R.-U. Saputra, P.-P.-B. de Gusmao, A. Markham, N. Trigoni, Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks, in: Proc. 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.

[23] G. Wang, C. Zhang, H. Wang, J. Wang, X. Wang, Unsupervised Learning of Depth, Optical Flow and Pose with Occlusion form 3D Geometry, arXiv preprint arXiv:2003.00766 (2020).