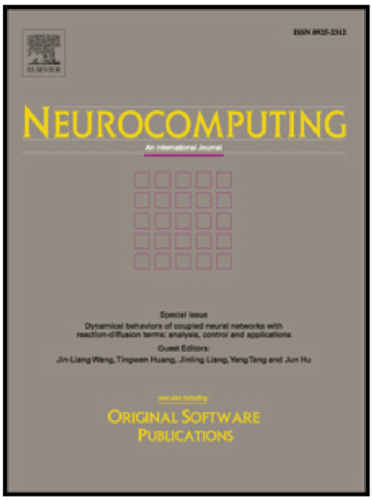


Unsupervised framework for depth estimation and camera motion prediction from video

Delong Yang , Xunyu Zhong , Dongbing Gu , Xiafu Peng , Huosheng Hu

PII: S0925-2312(19)31758-8
DOI: https://doi.org/10.1016/j.neucom.2019.12.049
Reference: NEUCOM 21678

To appear in: *Neurocomputing*

Received date: 27 March 2019
Revised date: 25 October 2019
Accepted date: 11 December 2019

Please cite this article as: Delong Yang , Xunyu Zhong , Dongbing Gu , Xiafu Peng , Huosheng Hu , Unsupervised framework for depth estimation and camera motion prediction from video, *Neurocomputing* (2019), doi: https://doi.org/10.1016/j.neucom.2019.12.049

Highlights

- Unsupervised framework for depth estimation and camera motion prediction
- Depth CNN and pose CNN are trained jointly and can be used respectively
- Only monocular images are required during testing
- Construct the supervision signal based on spatial and temporal geometry constraints
- A novel left-right geometric consistency loss is added to the objective function
- Results outperform previous unsupervised methods and some supervised methods
- A model which is trained on the Euroc dataset is used to test the algorithm's generalization capability.

Unsupervised framework for depth estimation and camera motion prediction from video

Delong Yang[a], Xunyu Zhong[a,*], Dongbing Gu[b], Xiafu Peng[a], Huosheng Hu[b]

[a] Department of Automation, Xiamen University, Xiamen 361005, China

[b] School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK.

Abstract

Depth estimation from monocular video plays a crucial role in scene perception. The significant drawback of supervised learning models is the need for vast amounts of manually labeled data (ground truth) for training. To overcome this limitation, unsupervised learning strategies without the requirement for ground truth have achieved extensive attention from researchers in the past few years. This paper presents a novel unsupervised framework for estimating single-view depth and predicting camera motion jointly. Stereo image sequences are used to train the model while monocular images are required for inference. The presented framework is composed of two CNNs (depth CNN and pose CNN) which are trained concurrently and tested independently. The objective function is constructed on the basis of the epipolar geometry constraints between stereo image sequences. To improve the accuracy of the model, a left-right consistency loss is added to the objective function. The use of stereo image sequences enables us to utilize both spatial information between stereo images and temporal photometric warp error from image sequences. Experimental results on the KITTI and Cityscapes datasets show that our model not only outperforms prior unsupervised approaches but also achieving better results comparable with several supervised methods. Moreover, we also train our model on the Euroc dataset which is captured in an indoor environment. Experiments in indoor and outdoor scenes are conducted to test the generalization capability of the model.

Keywords: Unsupervised deep learning; Depth estimation;
Camera motion prediction; Convolutional neural network.

1. Introduction

Depth estimation based on images has received much attention in recent years due to the properties such as convenience and real-time process which offer important information for simultaneous localization and mapping [1], self-driving platforms and interactive collaborative robotics [2], etc. The purpose of depth estimation is to predict the distance from a scene to the camera based on the image directly. This topic is divided into two technical strategies: traditional methods and deep learning models.

Traditional methods include structured light [3], time-of-flight [4], structure-from-motion [5], photometric stereo method [6], stereo matching [7,8] and symmetric models for 3D object structure estimation [9,10], etc. These methods

* Corresponding author. E-main address: zhongxunyu@xmu.edu.cn

typically formulate depth estimation as multi-views problems. Stages of traditional methods such as feature extraction, feature description, feature matching and bundle adjustment are time-consuming. In addition, some regions such as the motorway and building facade are smooth on the surface so that few matching points can be extracted. In fact, extracting features from these non-texture regions which lack of high-quality features such as feature points or edges. Therefore, this problem has not been resolved in the traditional way.

To overcome these shortcomings, convolutional neural networks (CNNs) [11–14] have been widely used in monocular depth estimation tasks, and they have achieved considerable improvement against traditional methods. One of the main reasons for this improvement is big data which makes CNNs obtain pixel-wise semantic information in all regions of images. The other reason is that the generated CNN models compute the scene depth much faster than traditional methods in practical application.

CNN models attempt to estimate the scene pixel-wise depth map which corresponding to the image directly. This strategy has received much attention over the past several years due to its properties such as real-time processing, therefore predicting the scene depth from a single image without prior information has become a fundamental topic in computer vision. Recently, deep learning methods have been divided into two types: supervised deep learning methods which require ground truth for training and unsupervised deep learning methods without the need for ground truth.

Supervised deep learning methods require vast amounts of labeled training data (ground truth) which is usually obtained by active RGB-D cameras in the indoor setting and 3D laser scanners [15] in the outdoor scenes. However, the supervised strategy bears several shortcomings because of the need for ground truth. Firstly, the network may be influenced by the sensors' own error and noise. Secondly, these sensors' measurements are typically sparser than images so that they cannot capture high-resolution information as well as images. Finally, in some places, ground truth cannot be obtained by those sensors. Therefore, unsupervised methods that rely only on training data have captured more attention from the researchers.

Our method is based on the fact that supervision signals can be generated through image rendering. This paper introduces an end-to-end approach for monocular depth estimation and camera motion prediction. It is a novel scheme that uses stereo image sequences for training. Then we can use the generated model to estimate the depth of monocular images during the testing process. In addition, we can also obtain the camera motion of the monocular image sequences. It is an unsupervised framework which can be trained simply using stereo image sequences without ground truth. Moreover, we construct a left-right consistency loss function as a part of the objective function to improve the accuracy and robustness of the model.

In summary, we propose a novel monocular depth estimation and camera motion prediction scheme in an unsupervised way. The CNN structure and objective function are discussed in this paper, then we use BGD (Batch Gradient Descent) to calculate the network's parameters through iterative computing. After all, the generated model is utilized to obtain the monocular image's depth map and its corresponding camera motion in an end-to-end way. Our main contributions are as follows:

- This paper presents a novel framework that uses stereo image sequences as input data to learn an unsupervised model for depth estimation and camera motion jointly.
- We present a novel composition framework with left-right consistency. The framework utilizes the spatial and temporal geometry constraints to construct the objective function.
- The experiments on the KITTI and Cityscapes datasets demonstrate that our model outperforms previous unsupervised methods and some supervised methods.
- The experiments on the Euroc dataset are completed to test the generalization capability of the presented technique.

The remainder of this paper is organized as follows:

Section 2 gives a review of related works. Section 3 gives the detail of our end-to-end model and implementation details. Section 4 shows experimental results on the KITTI, Cityscapes and Euroc datasets. Finally, we give a conclusion of this paper in Section 5.

2. Related works

There is plenty of published papers that pay close attention to depth estimation from images, either using stereo image pairs, temporal image sequences or multi-view images. It is inconceivable to understand the structure of a scene from single-view images in traditional methods. Fortunately, deep learning has achieved great prosperity in computer vision since the breakthrough work of [16]. The vast majority of depth estimation algorithms based on CNNs are supervised. These approaches need more than one labeled dataset to learn parameters. To address this issue, here we concentrate on an unsupervised method to estimate scene depth and predict camera motion. In the following, we give a brief introduction to the most closely related work.

2.1 Traditional depth and camera pose recovery methods

Recovering scene depth and camera pose have been studied by computer vision researchers for many years. Konrad et al. [17] propose a 2D-to-3D image conversion for depth estimation from examples. In [18], a plausible depth generation technique from videos which used non-parametric depth sampling as auxiliary information was proposed. This technique outperformed all the state-of-the-art traditional depth methods. As another fundamental research topic of the computer vision community, camera pose recovery has been very successful in traditional strategy. The most famous traditional algorithm for camera pose recovery methods from images is ORB-SLAM [19], which is a feature-based simultaneous localization and mapping system from monocular images. Stages of ORB-SLAM include tracking, mapping, re-localization, and loop closing. However, all the stages must be designed carefully.

Based on the fact that the structures of many man-made objects are symmetric, Gao et al. [9] extended this information from 2D images to 3D object reconstruction, and used symmetry to improve non-rigid structure from motion algorithms. In [10], a 3D structure and camera projection estimating method was proposed. The input of this model came from various intra-class object instances and the symmetry was extended to the multiple-image case. Ma et al. [20] proposed a locally linear transforming model to match both rigid and non-rigid features of remote sensing images. All the above methods either reconstruct the underlying 3D geometry or establish the correspondent relationships per-pixel among input views to obtain the scene depth. Nevertheless, these methods use multi-view images as input data.

2.2 Supervised learning from monocular image

The task of estimating scene depth from a monocular image is a challenging topic since we cannot get the geometric structure by only one view image. Recently, some researchers have treated depth estimation as a supervised learning process. Eigen et al. [11] proposed a network which consisted of two components, the first one estimates the global structure of the scene, then the other uses neighborhood information to refine it. To the best of our knowledge, it is the first paper that predicts scene depth from monocular images based on deep CNN. On the basis of previous work, Eigen et al. [12] addressed a framework to process three different computer vision tasks (depth estimation, surface normal prediction, semantic labeling) simultaneously. Laina et al. [21] proposed a fully convolutional architecture to model the ambiguous mapping between monocular images and depth maps. Li et al. [22] presented a fast-to-train multi-streamed CNN architecture for depth estimation. Yan et al. [23] used surface normal as a reference to assist the task of monocular depth estimation.

Until now, some works have tackled monocular depth estimation combined CNNs and Random Forests. Li et al. [24] coped with this problem by regression on deep CNNs features, combines with a post-processing refining step using conditional random fields. Roy et al. [25] presented a novel neural regression forest that combines random forests and CNNs for depth estimation from a single image. Liu et al. [26] formulated depth estimation into a continuous conditional random field learning problem based on the continuous characteristic of the depth values. Even though the above methods have achieved accurate results for monocular depth estimation, these approaches rely on ground truth for training, which restricts the generalization ability of the model.

2.3 Unsupervised learning from monocular image

In order to overcome the limitation of ground truth, some unsupervised learning frameworks for the task of monocular depth estimation were presented recently. Garg et al. [13] used pairs of images with known camera motions as input data, to learn a CNN to model the complex non-linear transformation which converts the images to depth-maps. Based on Garg’s work, Ren et al. [27] and Yu et al. [28] constructed a spatial smoothness loss to add to the total loss function for unsupervised optical flow learning. Their works and results are similar. Godard et al. [29] treated depth estimation as an image reconstruction problem during training. A loss function is constructed to learn the correspondence between the rectified stereo images by using epipolar geometry constraints. Kuznietsov et al. [30] used predicted inverse depth and sparse ground-truth depth as input to estimate scene depth in a semi-supervised way. Their models require an accurate extrinsic calibration between the 3D laser sensor and the camera. Yan Hua and Hu Tian [31] proposed Convolutional Conditional Random Field Network (CCRFN) for feature learning and depth estimation. CCRFN has two advantages, one is it does not need hand-crafted features and the other is it makes use of the relationship between individual features for depth estimation.

Zhou et al. [32] proposed an unsupervised learning framework for the monocular depth estimation and camera motion prediction synchronously. To our best knowledge, it is the first paper that uses monocular image sequences for training and testing. On the basis of Zhou's work, Yin et al. [33] proposed an unsupervised learning framework named GeoNet [33], which predicts monocular depth, optical flow and detect dynamic objects jointly. Luo et al. [34] come from SenseTime Research presented a method that reformulates the monocular depth estimation problem as two sub-problems followed by stereo matching. Pilzer et al. [35] presented an unsupervised depth estimation framework which is the first paper that uses cycled generative networks. Moreover, it is the first paper that utilizes cycled generative networks to estimate the scene depth. Tulsiani et al. [36] proposed an unsupervised framework which without using ground truth directly for learning single-view shape and pose prediction of indoor instances.

These unsupervised learning models have used pairs of images captured from a stereo camera with accurate calibration or monocular image sequences as supervision. Stereo images cannot take full advantage of temporal information. Monocular images suffer from an inherent problem, depth ambiguity, which means different depths may correspond to the objects with similar appearances in the image. However, although these unsupervised models have achieved the goal that estimated scene depth without ground truth, little attention has been paid to the strategy of jointly uses stereo image pairs and image sequences for depth estimation.

3. Method

In this section, we describe the unsupervised framework for depth estimation and camera motion prediction from monocular videos. During training, we use stereo image sequences captured by a moving binocular camera as input data of the depth and pose CNNs. In spite of being jointly trained, these two CNNs can be used independently in the practical application.

3.1 Overview of our method

The presented an unsupervised learning model can be divided into two parts: depth estimation and camera motion prediction, which can be trained jointly. The photometric warp error between the synthesized and input image is selected to construct the supervision signal.

The overview of our model is shown in Fig. 1. It is composed of two parts, one for depth estimation and the other one for camera motion prediction. The first part estimates scene depth maps through the depth CNN and the second part is conducted by the pose CNN to compute the camera pose between the images of an image sequence. Furthermore, in the second part, we use stereo image sequences and the scene depth obtained from the first part as input for training. Considering the constraints of stereo image pairs and monocular image sequences, we obtain more robust results than other methods. Last but not least, during training, a geometric consistency check which improves the accuracy of the algorithm is added to the objective function.

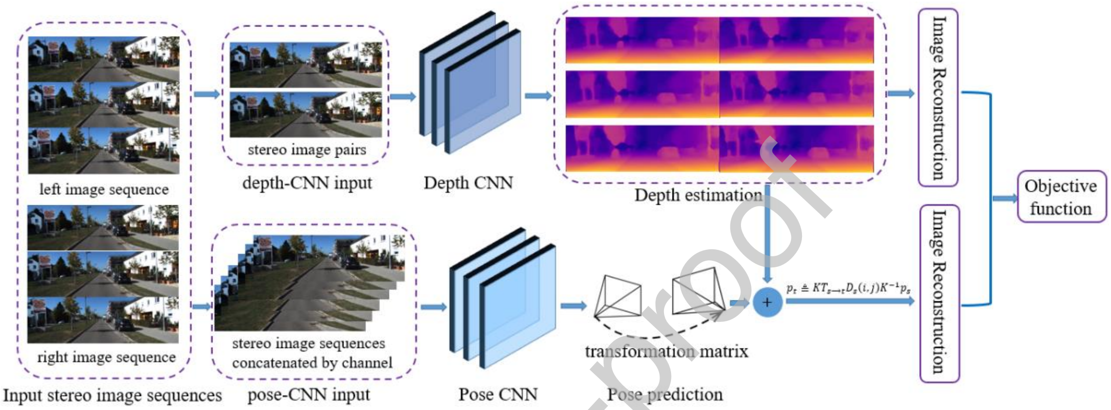


Fig. 1. Overview of our method. Training samples consist of unlabeled stereo image sequences captured from a binocular camera which does not provide the pose of image sequences. This model consists of depth CNN to estimate scene depth and pose CNN to predict camera motion. These two CNNs use image reconstruction instead of ground truth for training. They are training synchronously and operate independently, one for single view depth estimation and the other for camera motion prediction.

3.2 Depth estimation

Given a single image during testing time inference, our purpose is to learn a function $d = f(I)$ which can estimate the per-pixel depth map corresponding to the input image.

This function is actually a CNN which has numerous fixed parameters for depth estimation of a single RGB image. While CNN is used, the depth map can be computed in an end-to-end way through a series of non-linear operations of CNN's layers. All the parameters of these layers have been already calculated during the training process. Therefore, the goal of training is to get all CNN's parameters.

In order to achieve this purpose, we use stereo image pairs as input data for training. It is an iterative process with the use of BGD (Batch Gradient Descent). We construct a loss function for this BGD process and use six stereo image pairs as a batch. Each training sample is a stereo image pair which is composed of I^l and I^r, they are corresponding to the left and right color image which captured from a moving binocular camera synchronously.

We use the disparity map estimated from the depth CNN instead of trying to estimate the scene depth map directly. We assume that all the stereo images are rectified [37] and the surface is Lambertian (make the photo-consistency error is meaningful). For a stereo image pair, we denote the disparity map which corresponding to the left image of this stereo image pair is D^l, the right synthesized image is reconstructed by the formula $\widetilde{I^r} = I^l(D^l)$. The reconstruction function we have just used can be expressed as:

$$\widetilde{I^r}\left(i, j + D^l_{i,j}\right) = I^l(i,j), (i,j) \in \Omega_l \qquad (1)$$

where I_l is the left image of the input stereo image pair, Ω_l is the image pixel space corresponding to I_l, $\widetilde{I_r}$ is the right synthesized image generated from the left image and the right disparity map, (i,j) is the coordinates of a pixel in the image. The left synthesized image $\widetilde{I^l} = I^r(D^r)$ can be reconstructed similarly.

Depth estimation using stereo image pairs obey primary geometric constraint, therefore, this model can be learned without ground truth. Stereo images are captured from binocular cameras that have good synchronization and calibration so that the pixels in the two images of the stereo image pair have a strong correspondence.

These synthesized images are key components of the loss function for our depth CNN. After obtaining the predicted disparity maps, the depth map $d(i,j)$ can be computed by the following linear mapping:

$$d(i,j) = bf/D(i,j) \qquad (2)$$

where b is the binocular camera's baseline, f is the camera's focal length, (i,j) is the pixel coordinate of an image.

The baseline and camera's focal length of the binocular camera are changeless so that the use of stereo image pairs during training allows us to get the absolute scale of monocular depth estimation. Specifically, for a stereo image pair, each pixel in the overlapped area of one image can find its corresponding pixel in the other image of the stereo image pair with horizontal distance (disparity) [13]. The binocular camera's baseline b and the camera's focal length f establish the absolute correlation between the scene depth and the disparity map, and the disparity map determines the image reconstruction effectiveness which has a significant influence on the construction of the loss function. Our model relies on the loss function based on the spatial geometry constraints (formula (1) and (2)) to recover the absolute scale for the monocular depth estimation and camera motion prediction during training. In the testing time, our model can be used to estimate absolute depth for monocular images.

During training, six stereo image pairs are treated as a mini-batch. Our goal is to learn a depth CNN model which generates disparity maps corresponding to the input images. The key insight of this method is that the stereo image pairs are fed through the depth CNN layers, they can produce the left-to-right and right-to-left disparity maps simultaneously. Then we use these disparity maps and original input stereo image pairs to reconstruct the synthesized stereo images. The loss function is constructed based on the difference between the synthesized stereo images and the original stereo images. Therefore, the accuracy of disparity maps generated by our depth CNN has a decisive effect on image reconstructing results. The architecture of the depth CNN is as follows:

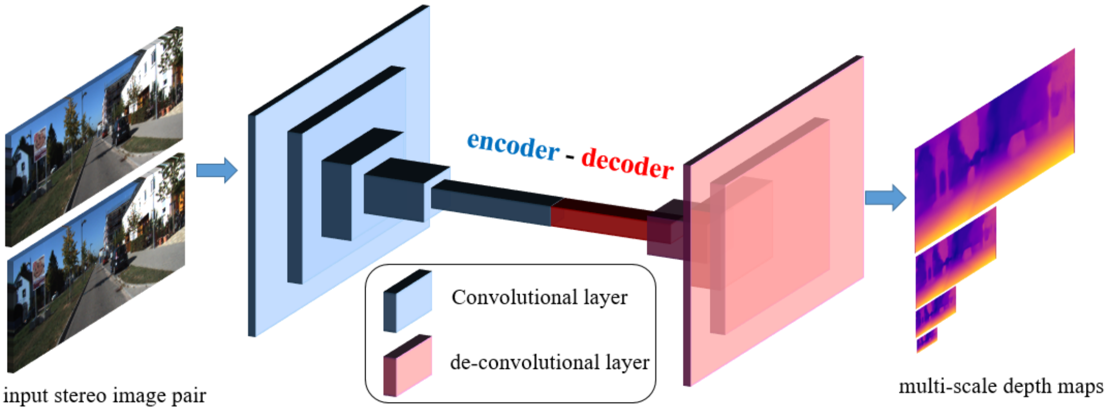


Fig. 2. Architecture of depth CNN. It is an encoder-decoder model, the width and height of each cube indicate the spatial dimensions of the output feature map respectively. The cube channel indicates the channels of the output feature map at the corresponding layer. Each reduction or increase in scale indicates a change by the factor of 2. We adopt the residual net architecture with four scales side predictions. The kernel size is 7 for the first convolutional layer and others are 3. The number of output channels for the first convolutional layer is set to 64 and the last is 2.

The network of the depth CNN is composed of two stages namely encoder and decoder (as shown in Fig. 2). We select ResNet50 as an encoder to extract high-level features and use deconvolution as a decoder to output disparity maps at four different scales. The resolution of the output disparity map is twice that of the previous image. At each output scale s, we define an item of the loss function as L_depth for evaluating the loss of the depth CNN at this specified scale.

Taking account of the fact that images are subordinate to a great diversity of distortions during acquisition and processing, a structural similarity index [38] is selected for measuring photometric loss after image synthesizing. This image similarity measurement maintains an appropriate assessment between appearance similarity and modest resilience for image distortions. In addition, an L1-loss is added to the photometric image reconstruction cost L_ap at each scale. The ultimate purpose of our appearance loss function is to measure the difference between the original input image and its corresponding synthesized image. We suppose the right synthesized image from the left image is $\widetilde{I^r}$, the appearance difference at scale s between $\widetilde{I^r}$ and the original right image I^r is formulated as:

$$L\_ap_s^{right} = \frac{1}{N}\sum_{i,j} \alpha \frac{1 - SSIM(I_{ij}^r, \widetilde{I_{ij}^r})}{2} + (1-\alpha)\left\| I_{ij}^r - \widetilde{I_{ij}^r} \right\|_1 \qquad (3)$$

where N is the number of total pixels in the image, α is a weight parameter, i, j indicate the abscissa and ordinate of each image pixel respectively.

Similarity, the appearance difference at scale s between the left synthesized image $\widetilde{I^l}$ and the original left image I^l is formulated as:

$$L\_ap_s^{left} = \frac{1}{N}\sum_{i,j} \alpha \frac{1 - SSIM(I_{ij}^l, \widetilde{I_{ij}^l})}{2} + (1-\alpha)\left\| I_{ij}^l - \widetilde{I_{ij}^l} \right\|_1 \qquad (4)$$

The final appearance difference at scale s is:

$$L\_ap_s = \frac{1}{2}\left(L\_ap_s^{left} + L\_ap_s^{right}\right) \tag{5}$$

According to the formula (5), it is locally smooth on the disparity gradient. However, depth discontinuity often exists at image gradients in intuition. In order to filter out outliers and preserve sharp details, we use image gradient to construct an edge-aware depth smoothness cost term for the left image below:

$$L\_disp_s^{left} = \frac{1}{N}\sum_{i,j}\left(\left|\frac{\partial d_{ij}^l}{\partial x}\right| e^{-\left\|\partial x I_{ij}^l\right\|_1} + \left|\frac{\partial d_{ij}^l}{\partial y}\right| e^{-\left\|\partial y I_{ij}^l\right\|_1}\right) \tag{6}$$

The edge-aware depth smoothness cost term for the right image can be constructed as same as the left image, the formula is:

$$L\_disp_s^{right} = \frac{1}{N}\sum_{i,j}\left(\left|\frac{\partial d_{ij}^r}{\partial x}\right| e^{-\left\|\partial x I_{ij}^r\right\|_1} + \left|\frac{\partial d_{ij}^r}{\partial y}\right| e^{-\left\|\partial y I_{ij}^r\right\|_1}\right) \tag{7}$$

Therefore, the final edge-aware depth smoothness cost term at scale s is the average of the above loss items. According to the formulas (6) and (7), the last cost term with the consideration of the edge-aware depth smoothness is as follows:

$$L\_disp_s = \frac{1}{2}\left(L\_disp_s^{left} + L\_disp_s^{right}\right) \tag{8}$$

In order to improve the accuracy and robustness of our model, we introduce a left-right consistency part based on the coherence of disparity maps between the left and right images. Considering the fact that disparities of the left and right images on the same pixel locations are not equal, we use the left and right disparity maps which generated from the depth CNN to synthesize each other. We denote D^l is the left estimated disparity map and D^r is the right estimated disparity map. The same image reconstruction function as formula (1) is used to reconstruct the synthesized disparity maps. These reconstruction functions are expressed as:

$$\widetilde{D^r}\left(i, j + D_{i,j}^l\right) = D^l(i,j), (i,j) \in \Omega_l^{disp} \tag{9}$$

where D^l is the left estimated disparity map, Ω_l^{disp} is the image pixel space corresponding to D^l, $\widetilde{D^r}$ is the right synthesized disparity map generated from the left estimated disparity map and the right disparity map, (i,j) is the coordinates of a pixel in the disparity map. The left synthesized image $\widetilde{D^l}$ can be reconstructed similarity.

The calculation procedure at scale s is as follows:

$$L\_lr_s = \frac{1}{N}\sum \sqrt[2]{\left(\widetilde{D^l} - \widetilde{D^r}\right)^2} \tag{10}$$

In the summary, the total loss for stereo image pairs at all scales considers the difference between the synthesized image and the original input image, the edge-aware depth smoothness and the left-right consistency between the disparity maps. The loss function for depth CNN is as follows:

$$L\_depth = \sum_{s=1}^{4} \mu_1 * L\_ap_s + \mu_2 * L\_disp_s + \mu_3 * L\_lr_s \qquad (11)$$

where μ_1, μ_2 and μ_3 are weight parameters.

3.3 Camera motion prediction

The purpose of camera motion prediction is to learn a function $p = g(I)$ which is a CNN for predicting the camera motion of the input image. During training, the depth and pose CNNs are trained simultaneously. The architecture of our pose CNN is shown in Fig 3. The disparity maps and their corresponding original images are used as input data for predicting the camera motion. With a view to the fact that each image of an image sequence is captured in a very short time, and the two cameras of a binocular camera are extremely close to each other, we assume that the scene is static without dynamic objects, such as moving cars and pedestrians.

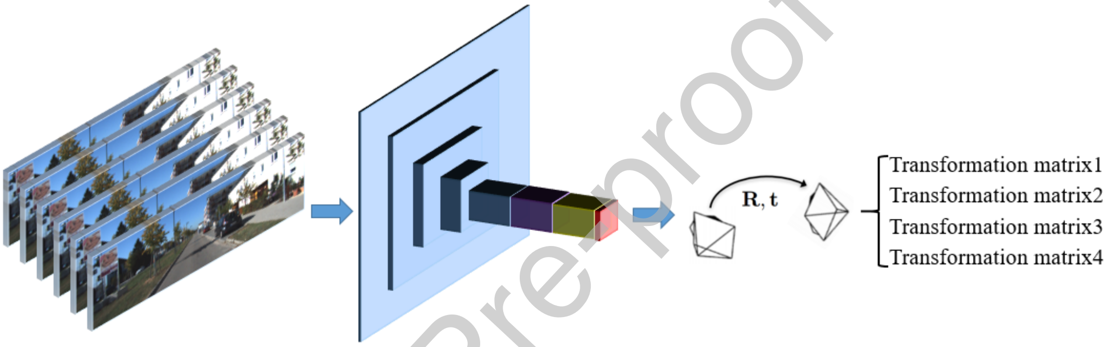


Fig. 3. Network architecture of the pose CNN. It is an encoder model, the output of this network is four matrices corresponding to the transformations from the source images to the target images. Each reduction in scale indicates a change by the factor of 2. The kernel size is 5 for the first convolutional layer and others are 3. The number of output channels for the first convolutional layer is set to 16.

The input stereo image sequence is decomposed into the left and right image sequence. Each of these two image sequences is composed of three frames, we specify that the second image is the target image and the other two images are the source images. The camera motions of the left and right image sequences are computed respectively.

The key supervision signal of the pose CNN comes from image synthesize. As similar to the depth CNN, assume the frames of a sequence are rectified. Let us denote $\{I_1, I_2, I_3\}$ as the consecutive frames of the left image sequence, the middle frame of the sequence is the target image I_t and the rest are the source images $I_n (n = 1,3)$. We define the disparity map corresponding to each frame of an image sequence as $D_i (i = 1,2,3)$, and the relative camera motion estimated by the pose CNN from the source image to the target image is defined as $T_{s\rightarrow t}$. The relative 3D transformation from the source image I_s to the target image I_t can be represented by

$$p_t \triangleq K T_{s\rightarrow t} D_s(p_s) K^{-1} p_s \qquad (12)$$

where K is the binocular camera intrinsic matrix, D_s is the disparity map corresponding to the source image, p_s and p_t denote the pixels of the source image and the target image respectively.

Based on the formula (12), we denote $I_{s1\to t}$ is the synthesized target image reconstructed from the source image I_1, $I_{s2\to t}$ is the synthesized target image reconstructed from the source image I_3.

The formulas of these synthesizing process are:

$$I_{s1\to t}(p_t) = I_t(KT_{s1\to t}D_{s1}(p_{s1})K^{-1}p_{s1}) \quad (13)$$

$$I_{s2\to t}(p_t) = I_t(KT_{s2\to t}D_{s2}(p_{s2})K^{-1}p_{s2}) \quad (14)$$

where $T_{sn\to t}$ is the transform metrics of the camera motion from the source image to the target image, $D_{sn}(p_{sn})$is the depth maps corresponding to the source image. As similar to the depth estimation CNN, the apparent difference between I_t and the synthesized target images $I_{sn\to t}(\mathrm{n} = 1,2)$ at scale s can be formulated as:

$$L_{p_s}^{left} = \frac{1}{2N}\sum_{n=1,2} \beta \frac{1 - SSIM(I_t, I_{sn\to t})}{2} + (1-\beta)\|I_t - I_{sn\to t}\|_1 \quad (15)$$

where N is the number of total pixels of the image, β is a weight parameter, divided by 2 at last because it is the loss of the synthesizes from the two source images to the target image.

In the process of forward-backward propagation of this CNN, gradient descent is the main calculation method. For image, the gradients are mainly computed by the pixel intensity difference between a pixel and its nearby pixels. However, some pixels are located in a low-texture region. In order to overcome this drawback and preserve the sharp details, we prefer a depth smoothness loss part as follows:

$$L\_s_s^{left} = \sum_{p_t}\left|\frac{\partial D(p_t)}{\partial p_t}\right| \cdot \left(e^{-\left|\frac{\partial I(p_t)}{\partial p_t}\right|}\right)^T \quad (16)$$

The final pose loss function at a special scale s of the left image sequence is:

$$L\_pose_s^{left} = \nu_1 * L\_p_s^{left} + \nu_2 * L\_s_s^{left} \quad (17)$$

where $\nu_1$and ν_2 are weight parameters.

As same as the left pose loss function, the pose loss function at scale s of the right image sequence is:

$$L\_pose_s^{right} = \nu_1 * L\_p_s^{right} + \nu_2 * L\_s_s^{right} \quad (18)$$

In summary, the total loss for stereo image sequence at scale s is the average of the above loss items. According to the formulas (17) and (18), the last cost term at all scales is as follows:

$$poss\_loss = \sum_{s=1}^{4}\frac{1}{2}\left(L\_pose_s^{left} + L\_pose_s^{right}\right) \quad (19)$$

3.4 The objective function

The overview of our objective function as shown in Fig. 4.

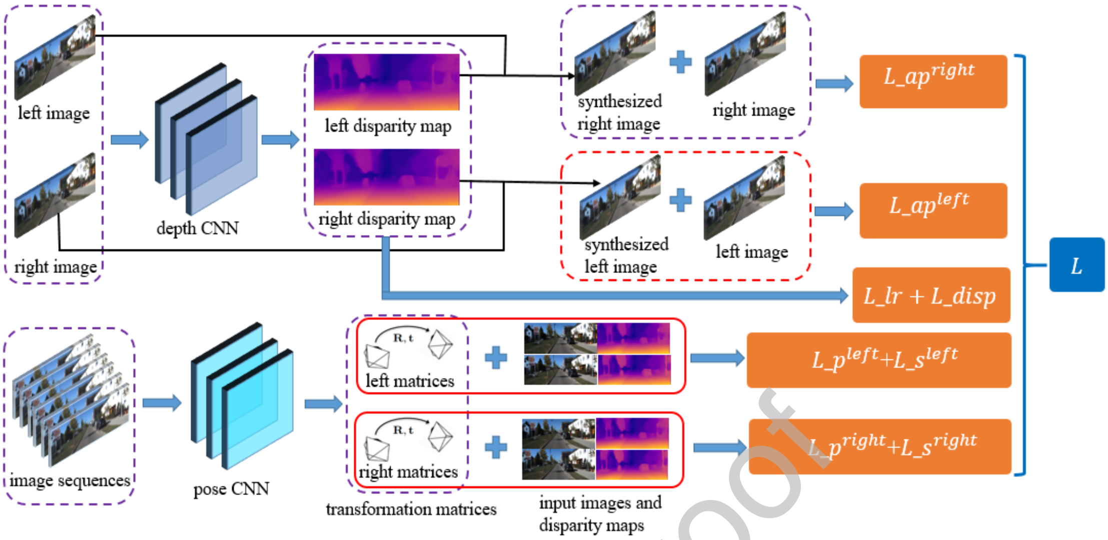


Fig. 4. For depth CNN, we use stereo image pairs as input data to generate corresponding disparity maps. In the process of image reconstruction, for example, we use the left input image and its corresponding depth map to synthesize the right image, then we utilize the generated right image and the right input image to construct the right loss part. The left loss part is similar to the right loss part. Stereo image sequences are fed into the pose CNN to compute transformation matrices from the source images to the target images, then we take advantage of the depth maps, the transformation matrices and the original input images to construct the objective function.

According to the formulas (10) and (18), with the consideration of the constraints of the stereo image pairs and the image sequences, the final objective function at all scales is defined as follows:

$$L = \lambda_1 depth\_loss + \lambda_2 pose\_loss \tag{20}$$

where λ_1 and λ_2 are the weight parameters for the depth estimation and camera pose prediction.

This article uses a stereo image sequence as input data for training, to construct the depth CNN and pose CNN for estimating the scene depth and predicting the camera pose simultaneously. Since the model has been generated, we use monocular images as input for testing.

4. Experiments

In order to evaluate the performance of our framework, comprehensive experiments are conducted on the publicly available KITTI [39] and Cityscapes [40] datasets. The Euroc dataset is also used to train the model, and various datasets are used to test the generalization capability of the presented framework. We compare our approach with a group of state-of-the-art schemes which include supervised and unsupervised frameworks. We also deploy our method on two widely-used CNNs (VGG-16 [41] and ResNet50 [42]) to discuss the effects of these two network structures. In addition, we conduct an ablation study to prove that the use of left-right consistency loss during training can improve the accuracy of depth estimation. To give the qualitative and quantitative analysis of our model, five commonly measures are selected to quantify our results in the task of monocular depth. Moreover, we use images that come from various datasets that include indoor and outdoor environments as input to the models which are trained on the KITTI and Euroc datasets to test the generalization capability. At last, we compare the results of our camera pose prediction with that of ORB-SLAM [19] and an unsupervised method [33].

In this section, we firstly give a brief description of the datasets we have used. Then we introduce the five common measurements and our training details. Lastly, the qualitative and quantitative results are displayed.

4.1 The experimental datasets

In order to compare with prior related works on monocular depth estimation, here we mainly use the KITTI dataset for evaluation. We also use the Cityscapes dataset for the benchmarking of cross-dataset generalization ability. In addition, we use the Euroc dataset to retrain our model for indoor environment depth estimation.

The KITTI dataset has been created by Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago in 2012 and it has been updated in 2015. The data was captured by a driving platform around the mid-size city of Karlsruhe, in rural areas and on highways. Up to 15 cars and 30 pedestrians are visible per image. The raw form of this dataset contains 42382 rectified stereo image pairs from 61 scenes with a typical image size being 1242*375 pixels. Considering consistent comparison, we take the split of Eigen et al. that 697 images come from 29 scenes are chosen for testing. We keep 29000 stereo image pairs for training. The Velodyne laser-scanned 3D points are projected onto the image planes in order to generate the ground truth to evaluate the model's performance.

The Cityscapes dataset has been created mainly by Benz. This large-scale dataset contains a diverse set of stereo image sequences recorded in street scenes from 50 different cities of Germany, with high-quality pixel-level annotations of 5000 frames and a larger set of 20000 weakly annotate frames. Because of the unsupervised method, the sub-datasets of Cityscapes dataset namely leftImg8bit_sequence_trainvaltest and rightImg8bit_sequence_trainvaltest are chosen for training. These two sub-datasets contain about 15000 stereo image pairs. At training time, we optionally pre-train the model on the two sub-datasets of the Cityscapes dataset.

The Euroc dataset consists of stereo images, synchronized IMU measurements, accurate motion and structure ground truth. Data of the Euroc dataset are captured in an indoor environment and only stereo images are required for our model. The stereo images are captured by an Aptina MT9V034 global shutter which is equipped to an AscTec Firefly unmanned aerial vehicle (UAV). All the stereo images are monochrome, which are different from those of the KITTI and Cityscapes datasets. The sub-datasets which are ASL dataset format are used to train and test our model.

4.2 Measurements

To evaluate the accuracy of the proposed method in monocular image depth estimation, we use these five scale-invariant metrics as follows to measure the error between our results and ground truth projected from the 3D laser.

- Abs Relative difference(Abs Rel): $\frac{1}{|N|}\sum_{y\in N}|y-y^*|/y^*$
- Squared Relative difference(Sq Rel):$\frac{1}{|N|}\sum_{y\in N}\|y-y^*\|^2/y^*$
- RMSE(linear): $\sqrt{\frac{1}{|N|}\sum_{y\in N}\left\|y_i-y_i^*\right\|^2}$
- RMSE(log10): $\sqrt{\frac{1}{|N|}\sum_{y\in N}\left\|logy_i-logy_i^*\right\|^2}$
- Threshold: % of $y_i\ s.t.max\left(\frac{y_i}{y_i^*},\frac{y_i^*}{y_i}\right)=\delta<thr,thr=1.25,1.25^2,1.25^3.$

where N is the total number of pixels on the ground truth image, y is the value of the predicted depth and $y^*$is the value of ground truth.

The first four metric measures the difference compares with ground truth, and the last metric measures the percentage of the predicted depth value which is within specified thresholds from the correct value. In addition, the maximum depth in the KITTI dataset is about 80 meters, so we set our maximum predictions of this value.

Here we must state that measuring the error in depth space leads to a precision result. Especially, the metrics without threshold measure may be sensitive to the large errors caused by estimated errors at small disparity values.

4.3 Training details

The networks of this article are implemented by TensorFlow. The ResNet50 contains about 65 million trainable parameters, and takes more than 23 hours for training; the VGG16 contains about 32 million trainable parameters and takes more than 16 hours for training. All the models are trained on a single NVIDIA GTX1080Ti GPU, and the number of iteration is 450 thousand. For fair comparisons with other frameworks, we train our model on the same dataset as [32]. In order to prevent overfitting, we perform random resizing, cropping and color augmentations for each image before training. Inference takes less than 25ms per image.

In the process of optimization, we set the weight parameters as follows:

$$\mu_1=1.0,\mu_2=0.1/\gamma,\ \mu_3=1.0$$
$$\nu_1=1.0,\ \nu_2=0.1/\gamma,\ \lambda_1=1.0,\ \lambda_2=0.8$$
$$\alpha=0.85,\ \beta=0.85$$

where γ is the downscaling factor of the layer which corresponds to the resolution of the input image. We use Adam for optimization with $\beta_1=0.9,\beta_2=0.999$, $\epsilon=10^{-8}$. The initial learning rate is 0.0002 for the first 250 thousand iterations and halving it until the end. For the activation function in the network, we find that exponential linear units (ELU) can improve the accuracy compares with rectified linear units (ReLU). The batch size is set to 2 with each training sample is a stereo image sequence which the length is set to 3.

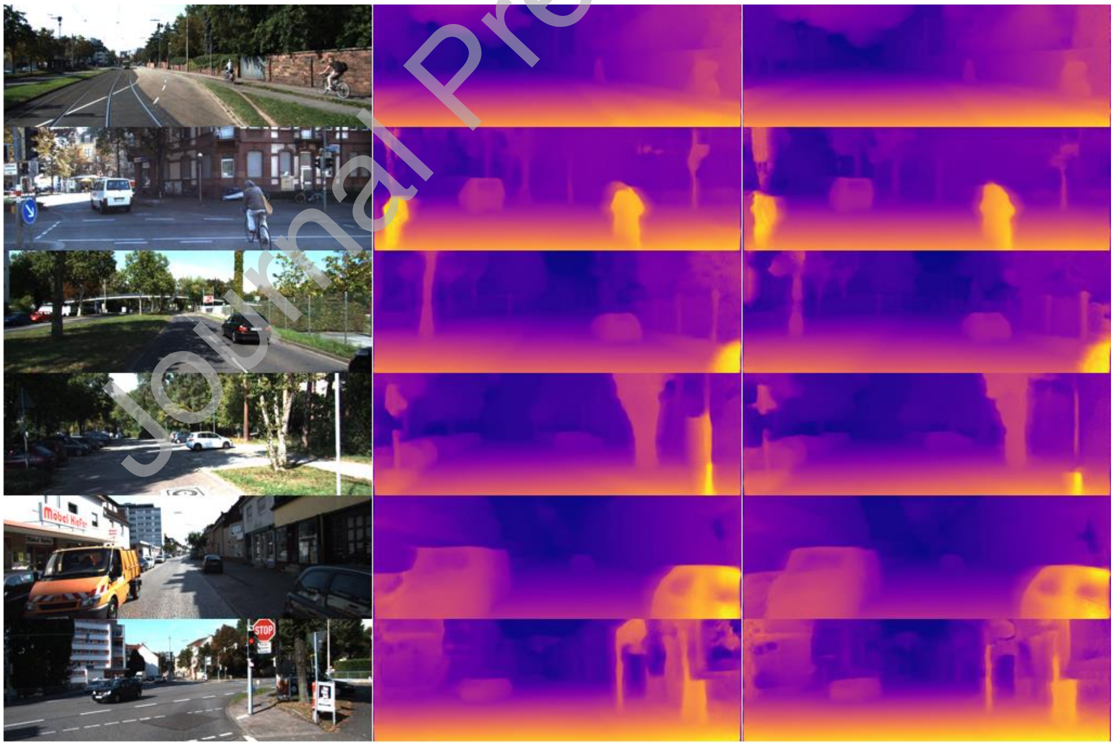
Additional, an identical weighting is used for the loss of each scale but led the network to an unstable convergence. Moreover, we also employ batch normalization in order to improve the performance but find that it is useless. In the final experiment, we exclude identical weighting and batch normalization ultimately.

4.4 Depth estimation

Nowadays, ResNet50 and VGG-16 networks have become the most famous CNN architectures. In order to compare the effects of these two networks, we use them as encoders to generate the disparity maps respectively (as shown in Fig. 5) and give the quantitive results in Table 1.

Table 1.
Results of our monocular depth estimation method with the use of ResNet50 network and VGG network for training.

Network	lower is better				higher is better		
	Abs Rel	Sq Rel	RMSE	REMS lg10	$\delta \leq 1.25$	$\delta \leq 1.25^2$	$\delta \leq 1.25^3$
ResNet50	0.142	1.259	5.768	0.229	0.801	0.933	0.976
VGG-16	0.146	1.304	6.021	0.242	0.785	0.928	0..965



(a) input images (b) results of ResNet50 (c) results of VGG-16

Fig. 5. The performance of monocular depth estimation between Resent50 and VGG-16.

As shown in Table 1, the difference between the ResNet50 and VGG-16 networks reveal that the results of ResNet50 outperform that of the VGG-16. Qualitative comparisons can be visualized in Fig. 5. Therefore, we choose ResNet50 as an encoder of our network architecture.

It is important to select a suitable activation function for the design of a CNN. The most commonly used activation function is the rectified linear unit (ReLU). However, through experiments, we found that the network with the exponential linear unit (ELU) has a more precise prediction compared with the network with ReLU.

Table 2.
Results of our method with the use of ReLU and ELU as the activation function for training.

Activation	lower is better				higher is better		
function	Abs Rel	Sq Rel	RMSE	REMS lg10	$\delta \leq 1.25$	$\delta \leq 1.25^2$	$\delta \leq 1.25^3$
ReLU	0.151	1.325	5.957	0.242	0.793	0.905	0.967
ELU	0.142	1.259	5.768	0.229	0.801	0.933	0.976

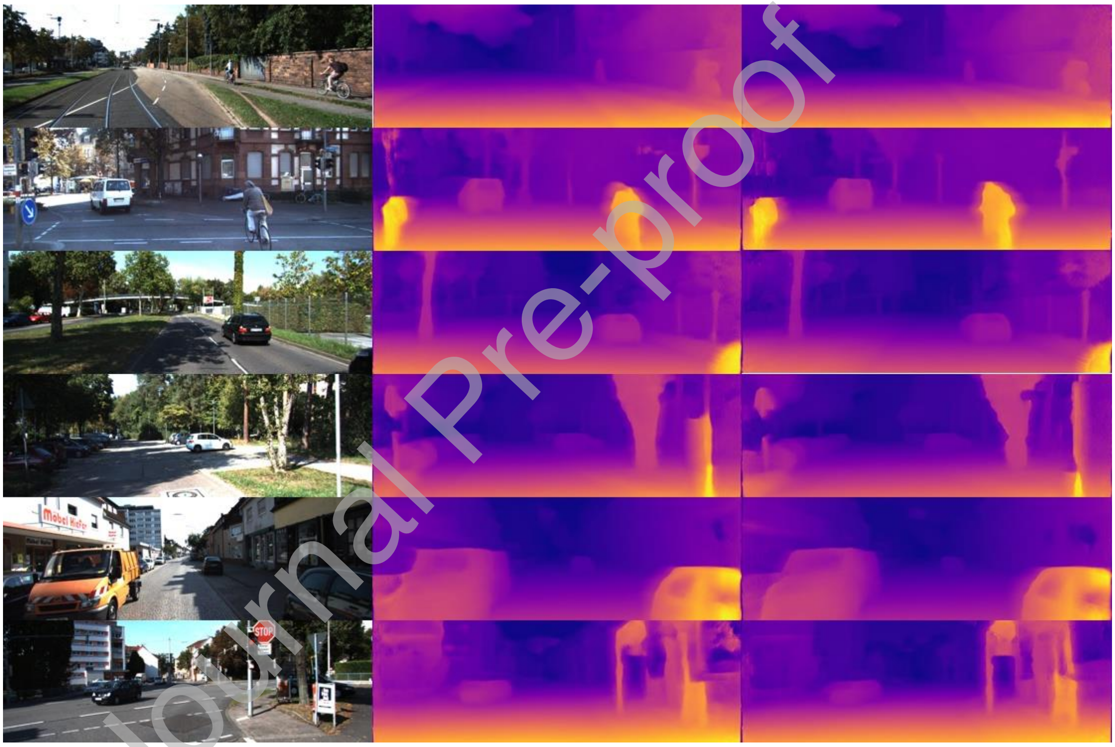

(a) input images (b) ReLU (c) ELU

Fig. 6. Qualitative visual results on the KITTI dataset. Note that the estimation of the model with ELU as its activation function is better than the model with ReLU as its activation function.

Table 2 shows the qualitative comparisons on the KITTI dataset and the results can be visualized in Fig. 6.

Moreover, a left-right consistency loss part is put forward. We use stereo image pairs as input data for training and each of them produces a disparity map. To achieve more precise disparity maps, the absolute value of the left disparity map should be equal to that of the right disparity map. In order to proof the left-right consistency loss part can improve the accuracy of the proposed model, we use the objective function without the left-right consistency to train the model. Table 3 shows the qualitative comparisons on the KITTI dataset and Fig. 7 gives the visual results.

Table 3.
Qualitative comparisons between the model with and without the left-right consistency.

left-right	lower is better				higher is better		
consistency	Abs Rel	Sq Rel	RMSE	REMS lg10	$\delta \leq 1.25$	$\delta \leq 1.25^2$	$\delta \leq 1.25^3$
with	0.142	1.259	5.768	0.229	0.801	0.933	0.976
without	0.147	1.285	5.902	0.235	0.785	0.912	0.958

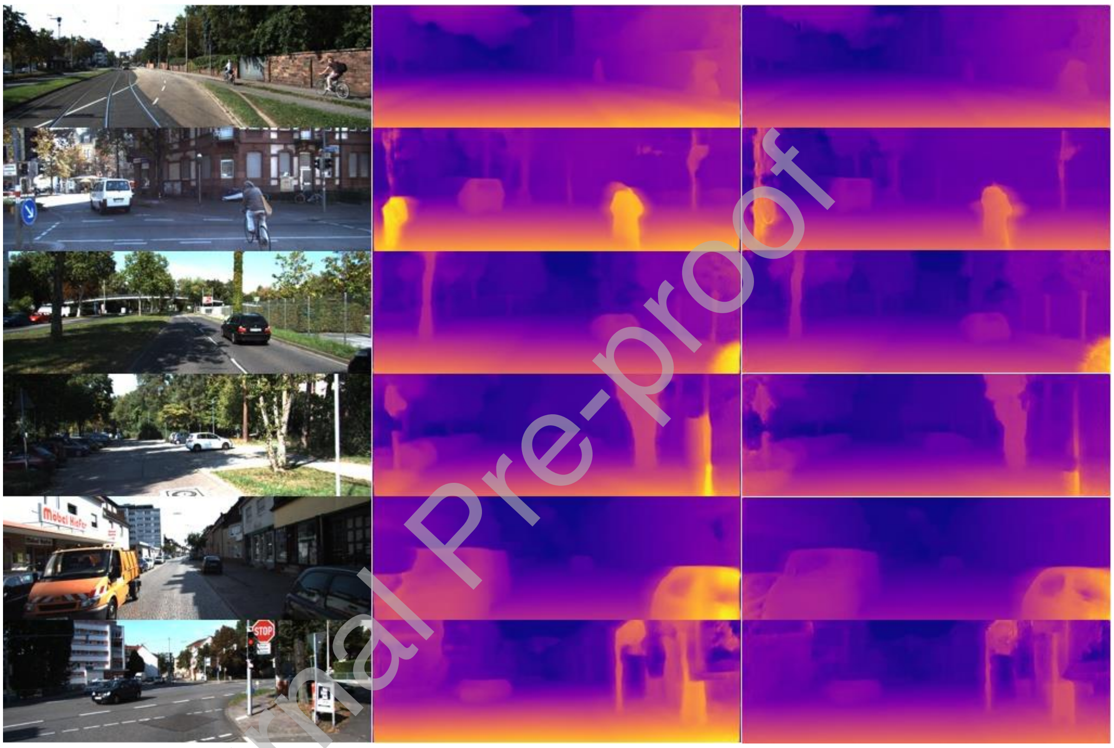

(a) input images (b) with (c) without

Fig. 7. Qualitative visual results between the model with and without the left-right consistency

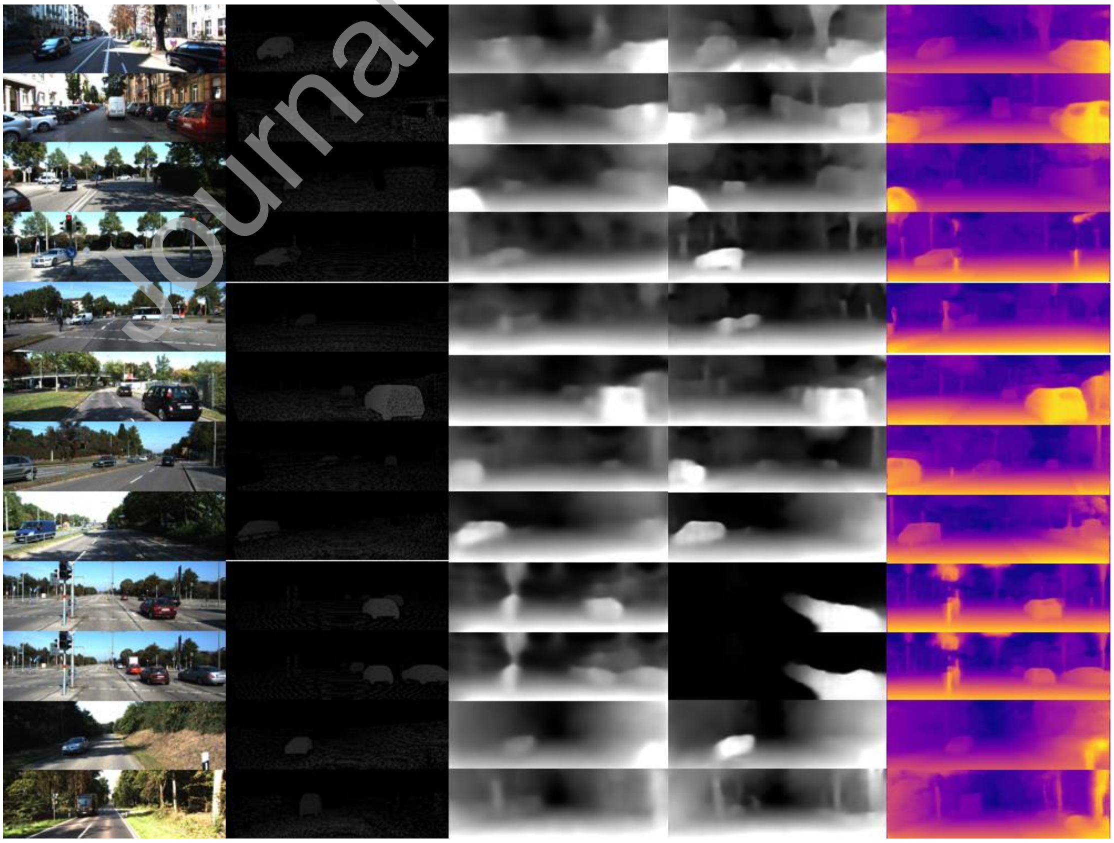
We compare our proposed method with some state-of-the-art depth estimation approaches including (1) Eigen et al. [11]Coarse (Eigen1); (2) Eigen et al. [12] Fine (Eigen2); (3) Liu et al. [26]; (4) Yan Hua et al. [23] ; (5) R. Garg et al. [13]; (6) Zhou et al. [32] (Zhou1); (7) Zhou et al. updated (Zhou2) [32]; (8) Geonet [33] ; (9) UndeepVo [43]; (10) GASDA [44]; (11) ACA (attention-based context aggregation method) [45]; (12) depth-SLAM [46]; (13) Cycle-Gan [35].These methods include several supervised methods and some unsupervised methods. The performance is shown in Table 4.

Table 4
Monocular depth estimation results on KITTI 2015 dataset.

Method	Super-v	lower is better				higher is better		
	ision	Abs Rel	Sq Rel	RMSE	RMSE lg10	$\delta \leq 1.25$	$\delta \leq 1.25^2$	$\delta \leq 1.25^3$

Eigen1	Yes	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen2	Yes	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu	Yes	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Yan Hua	Yes	0.336	-	10.70	-	-	-	-
ACA	Yes	0.083	0.437	3.599	0.127	91.9	98.2	99.5
R. Garg	No	0.177	1.169	5.285	0.282	0.727	0.896	0.958
Zhou1	No	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou2	No	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Geonet	No	0.155	1.296	5.875	0.233	0.793	0.931	0.973
Li's	No	0.183	1.73	6.57	0.268	-	-	-
GASDA	No	0.149	1.003	4.995	0.227	0.824	0.941	0.973
D-SLAM	No	0.180	1.510	6.349	0.256	0.741	0.906	0.966
Cycle Gan	No	0.190	2.556	6.927	0.353	0.751	0.895	0.951
Ours	No	0.142	1.259	5.768	0.229	0.801	0.933	0.976

As shown in Table 4, our unsupervised approach performs comparably with several supervised methods such as Eigen et al. and Yan et al. we also compare our method with some unsupervised methods as baselines. As shown in Table 4, our model outperforms most approaches but inferior to ACA [45] which introduces self-attention to a supervised framework in all measurements. We are also inferior to GASDA [44] which is based on the geometry-aware symmetric domain adaptation in part of the measurements. Moreover, for the visual SLAM approach [46] that added unsupervised learning-based depth estimation, we achieve a better result than it. Fig. 7 provides some comparable visual examples between our result and these baselines.



(a) input images (b) ground truth (c) Zhou’s (d) Geonet (e) ours

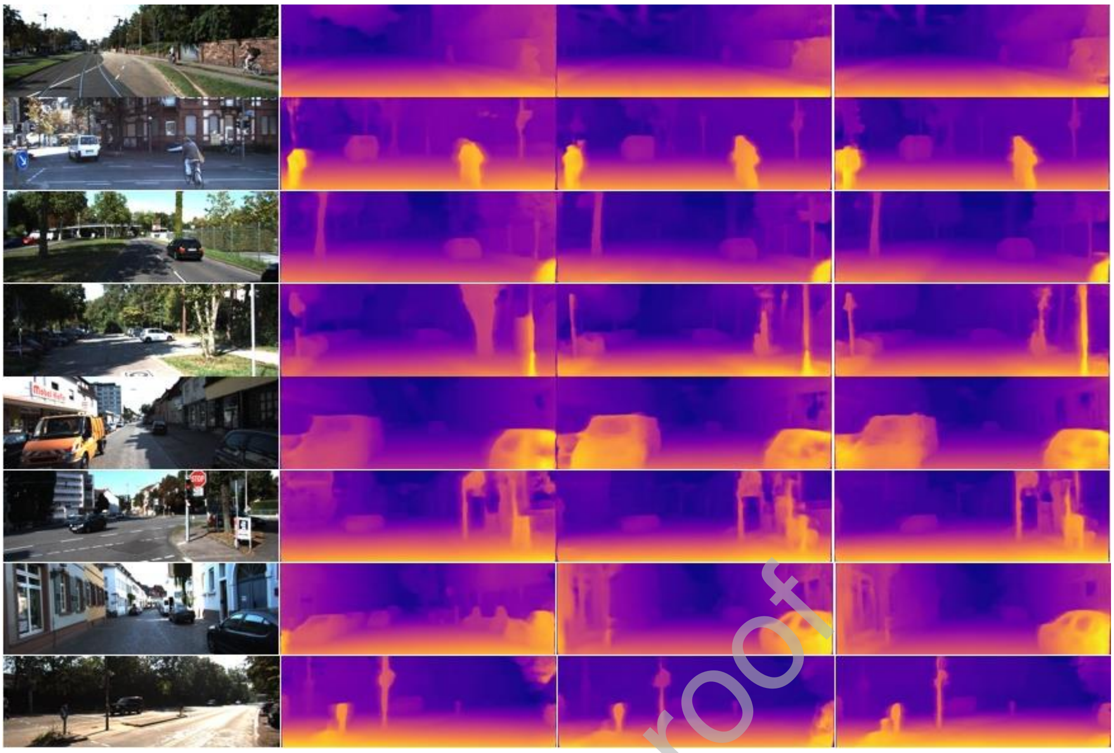
Fig. 7. Comparisons of the monocular depth estimation results between ground truth, Zhou et al. [32], Geonet [33] and ours.

To evaluate the generalization ability of our monocular depth estimation method, we apply our initial model which trained on KITTI dataset to estimate the disparity maps of the images selected from the Cityscapes dataset. The Cityscapes dataset we have used consists of stereo RGB image pairs, thus our method can train on this data directly. Here we train the model on the Cityscapes dataset solely and show the sample predictions by this initial Cityscapes model, the test images come from the KITTI dataset. Then we use the KITTI dataset and the Cityscapes dataset to train a new model. Moreover, we also give the depth estimation results of Zhou’s [32] and Geonet [33] that trained on these two datasets. Quantitative results on the test set of the KITTI dataset are shown in Table 5. In the table, ours (K) denotes the model trained on the KITTI dataset, ours (CS) denotes the model trained on the Cityscapes dataset, ours (K+CS) denotes the model trained on the KITTI dataset and the Cityscapes dataset.

Table 5.

Quantitative results on the test set of the KITTI dataset for the models trained on the KITTI dataset, the Cityscapes dataset and the KITTI + Cityscapes datasets.

Training dataset	lower is better				higher is better		
	Abs Rel	Sq Rel	RMSE	REMS log10	$\delta \leq 1.25$	$\delta \leq 1.25^2$	$\delta \leq 1.25^3$
Ours(K)	0.142	1.259	5.768	0.229	0.801	0.933	0.976
CS	0.209	1.704	6.985	0.285	0.739	0.867	0.923
Ours(K+CS)	0.122	1.079	4.998	0.211	0.854	0.941	0.978
Zhou’s	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Geonet	0.153	1.328	5.737	0.232	0.802	0.934	0.972



(a) input images (b) ours (c) Cityscapes (d) Kitti + Cityscapes

Fig. 8. Comparisons of the monocular depth estimation results between the models trained on the KITTI dataset, the Cityscapes dataset, the KITTI + Cityscapes datasets.

Fig.8 provides the results of the proposed model that trained on the two datasets. Fig. 8 (a) is the raw input images selected from the KITTI dataset, Fig. 8 (b) and (c) are the visual results of our model that trained only on the KITTI dataset and the Cityscapes dataset respectively, Fig. 8 (c) is the visual results of our model that trained on the KITTI dataset and the Cityscapes dataset. These pictures show that the model trained on the two datasets produces superior results on thin structures such as trees and lamppost. The model trained only on the Cityscapes dataset cannot capture the details on the boundaries such as cars. The experimental results show that the generalization ability of the model needs to be strengthened. In addition, we use the images selected from the Cityscapes dataset to test our models which trained on the KITTI dataset and the Cityscapes dataset. The results are shown in Fig. 9.

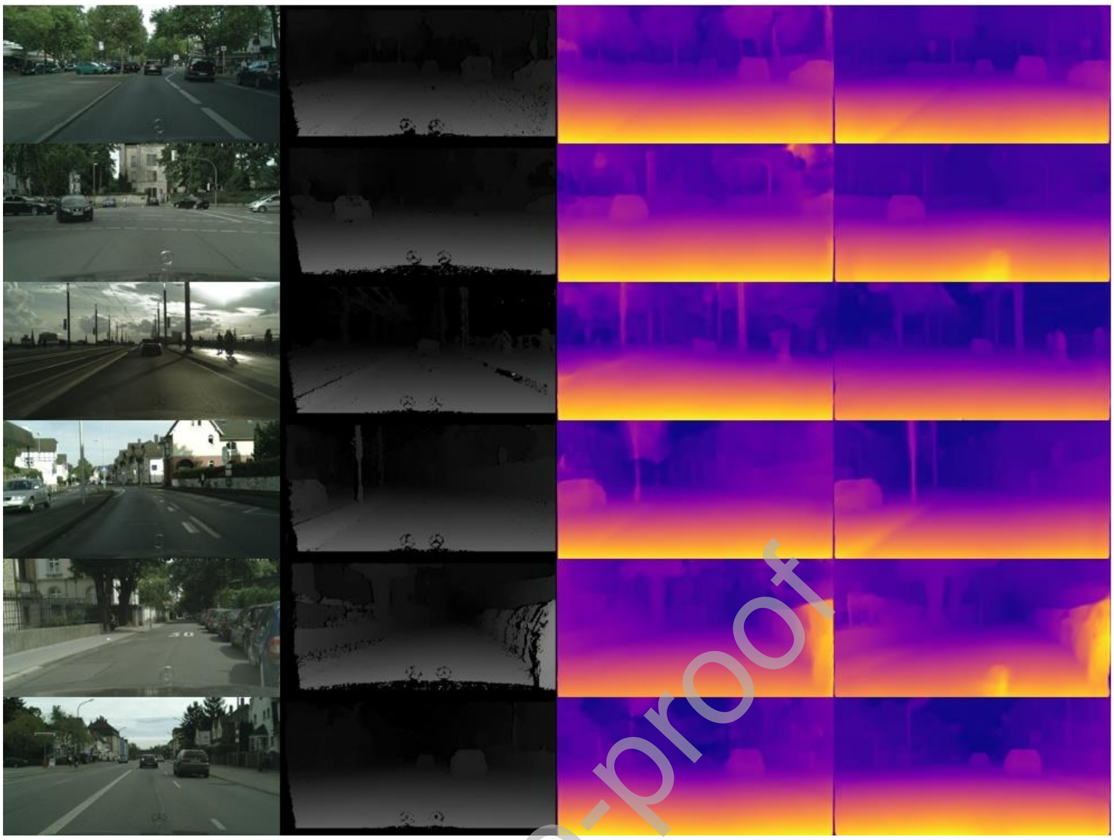


Fig. 9. Comparisons of the monocular depth estimation results between the models trained on the KITTI dataset, the Cityscapes dataset, the KITTI + Cityscapes datasets.

As shown in Fig. 9, (a) is the raw input images selected from the Cityscapes dataset, (b) is the ground truth corresponding to the raw input images, (c) and (d) are the visual results of our model that trained on the KITTI dataset and the Cityscapes dataset respectively.

4.5 Training on the Euroc dataset

Until now, all the models are trained on the KITTI and Cityscapes datasets, images of these two datasets are captured on cars in outdoor environments. To expand the application range of our algorithm, we use the Euroc dataset to train the model. We downloaded all the raw data from the official website of the Euroc dataset. For the sub-dataset named MH_01_easy.zip of the Euroc dataset, the number of left images is 3682, but the number of right images is 2273, hence, we only use images of MH_01_easy subdataset for inference. The rest stereo images of the dataset are chosen to train and test our model.

We use the same architecture to train the model for the indoor environment. 22977 stereo images are used for training, and the total number of iteration steps is about 280 thousand. During training, different image enhancement technologies are utilized to increase the diversity of the training data. Because of all the images of the Euroc dataset are monochrome, single-channel images are employed to inference.

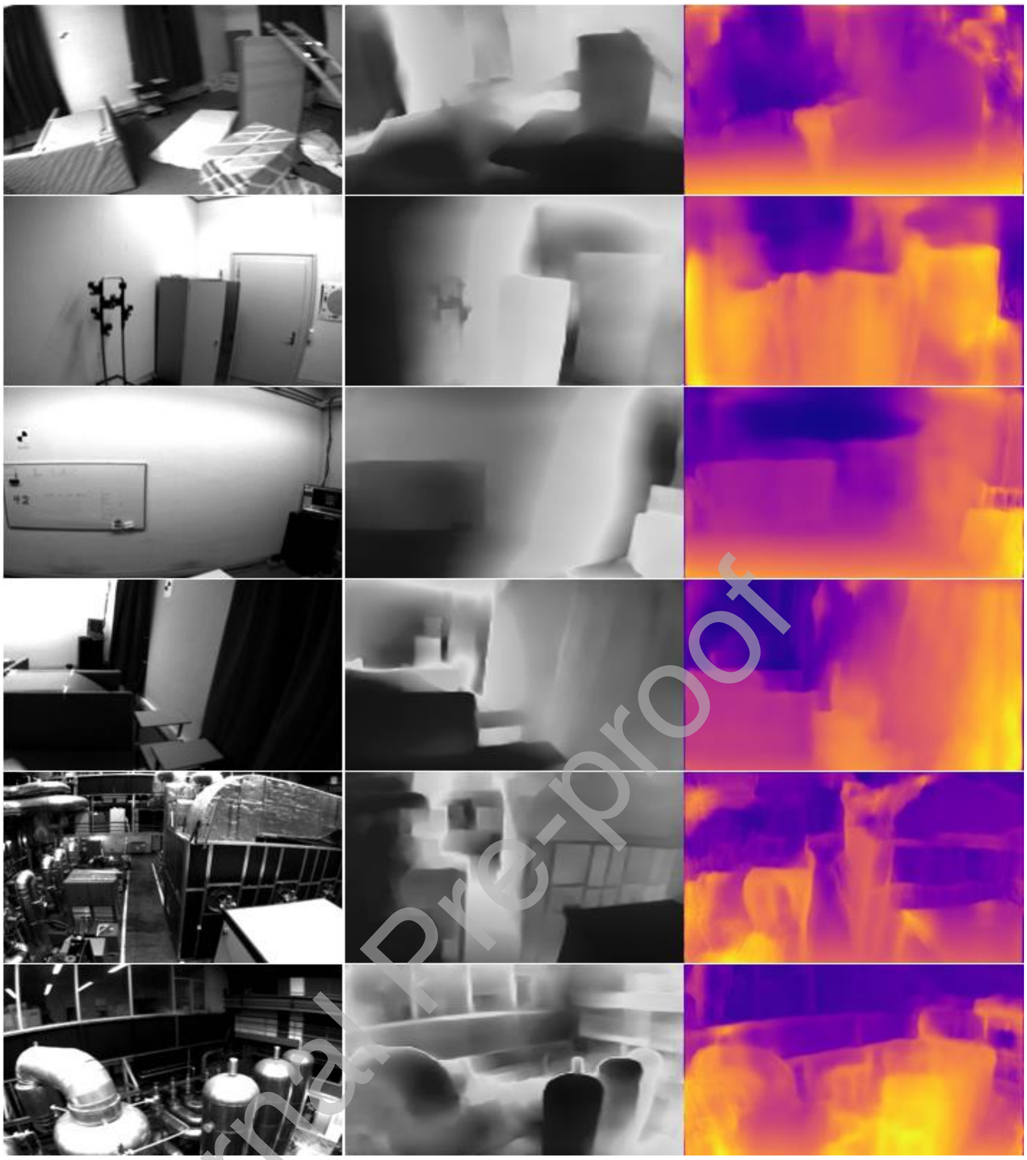
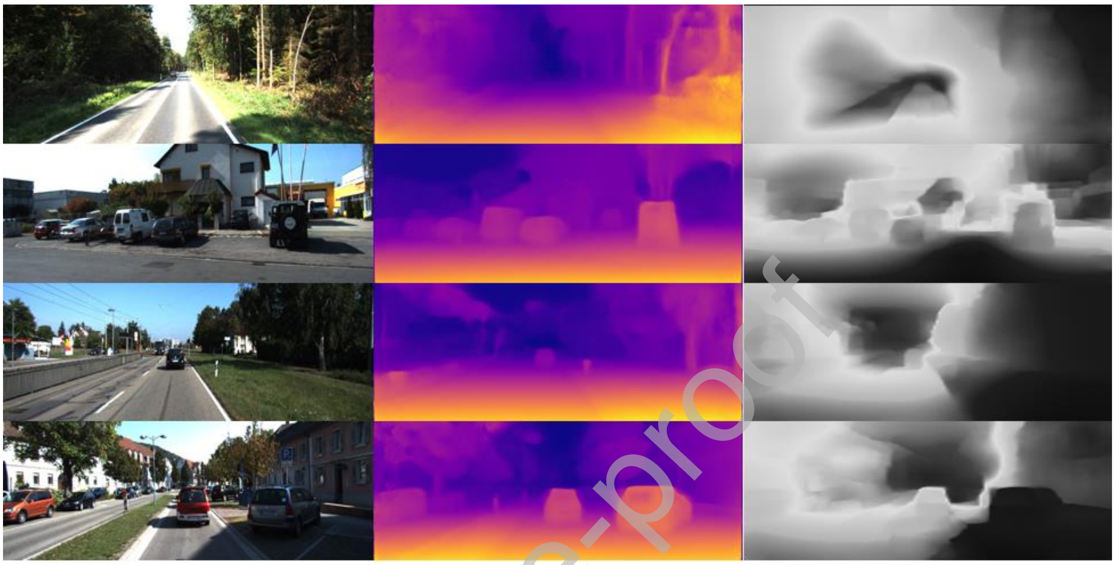


Fig. 10. Monocular depth estimation results. (a) Input images: input images come from the Euroc dataset randomly; (b) depth maps (E): the generated depth maps by the model which is trained on the Euroc dataset, (c) depth maps (K): the generated depth maps by the model which is trained on the KITTI dataset.

As shown in fig. 10, only single images of the Euroc dataset are required to generate the depth maps. We use two models that are trained on the Euroc and KITTI datasets respectively to infer the depth maps. The visual results of the model which is trained on the Euroc dataset (fig. 10(b)) are superior to that of the model which is trained on the KITTI dataset (fig. 10(c)). From fig. 10(c), we can hardly get depth information from the depth maps but the objects such as desk, door, whiteboard are clear in fig. 10(b). The results demonstrate that the model which are trained on a relatively fixed scene can be only tested on the very similar scene.

4.6 Generalization capability tests

Dataset plays an important role in the performance of the trained model. To test the generalization capability of the proposed algorithm, we first use several images selected from the KITTI dataset as input to the models, the visual results are shown in fig. 11. Then we use some images come from various datasets as input to the different models respectively. The used datasets include the ICLNUIM dataset [46], SUN3D dataset [47] and TUM RGBD dataset [48] for indoor environments, and the nuScenes dataset [49] for outdoor environments. Experimental results for generalization capability tests are shown in fig. 12.



(a) Input images (b) Depth maps (K) (c) Depth maps (E)

Fig. 11. Monocular depth estimation results. (a) Input images: input images come from the Euroc dataset randomly; (b) depth maps (K): the generated depth maps by the model which is trained on the KITTI dataset, (c) depth maps (E): the generated depth maps by the model which is trained on the Euroc dataset.

As shown in fig. 11, we use the model which is trained on the KITTI dataset to estimate the outdoor scene depth, the visual results are superior to the depth maps which are generated by the model trained on the Euroc dataset. Combined with the results of fig. 10, it is further proved that the testing scene should be similar to the training scene.

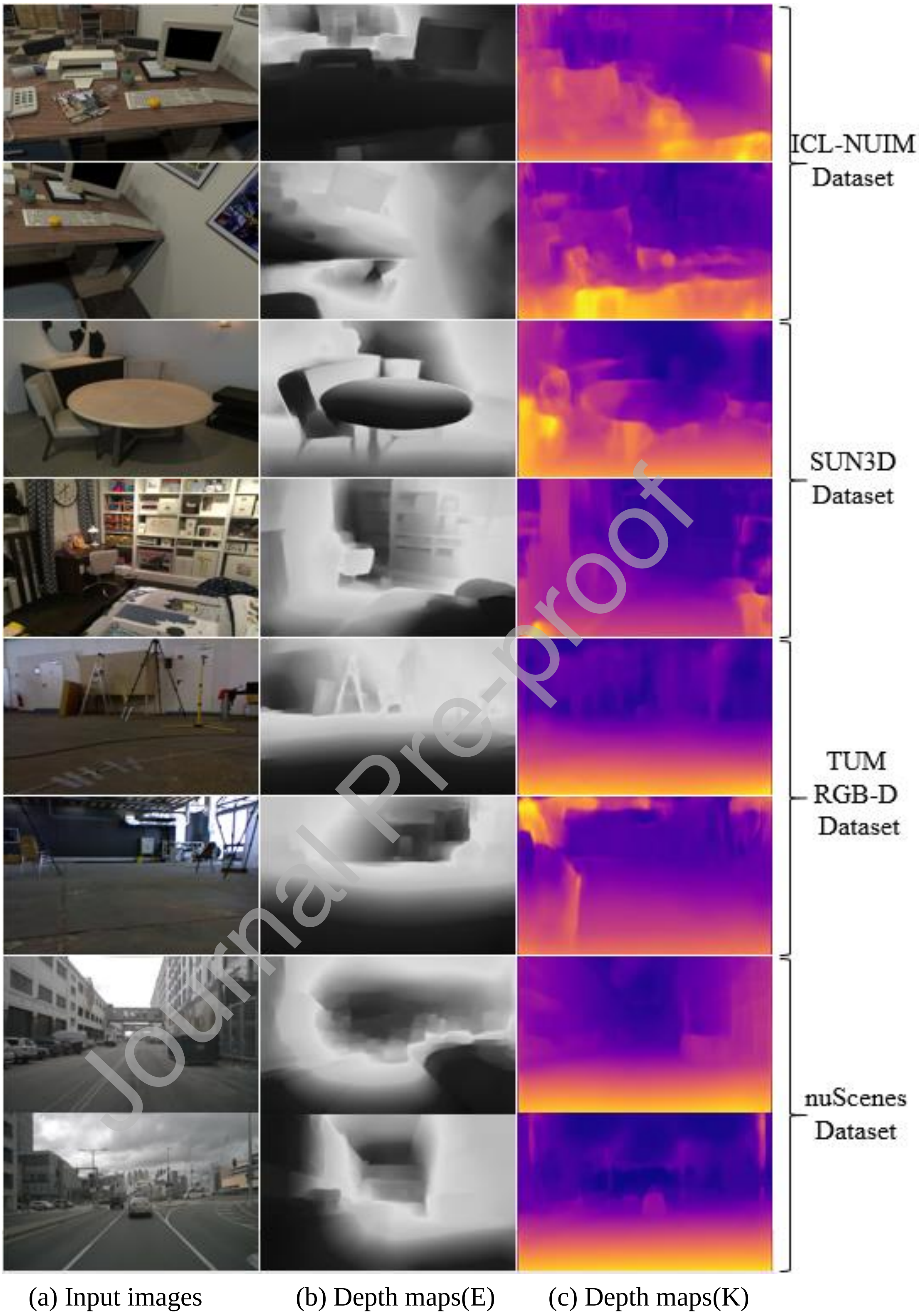


Fig. 12. Generalization capability tests. (a) input images denote the input images come from the different dataset, (b) depth maps(E) denote the generated depth maps by the model which is trained on the Euroc dataset, (c) depth maps(K) denote the generated depth maps by the model which two-loss on the KITTI dataset.

As shown in fig. 12(a), the input images are come from the ICL-NUIM, SUN3D, TUM RGB-D and nuScenes datasets, from top to bottom. Two images of each dataset are chosen for display. The generated depth maps by the two models which are trained on the Euroc and KITTI datasets can be obtained in fig. 11(b) and fig. 11(c), respectively. For images that come from the datasets that are collected in indoor environments, depth maps(E) are obviously superior to the depth maps(K) and vice versa.

The experiments of fig. 12 reveal that the performance of the model displays strong correlations with the trained dataset. Even though the presented technique has some advantages such as real-time process, pixel-wise generated depth images, only a single image is required for inference, its generalization capability cannot compare with the traditional methods.

4.7 Camera pose estimation

In order to evaluate the performance of our pose CNN, we apply our network to the official KITTI odometry split that containing 11 driving sequences with ground truth odometry. The ground truth odometry is obtained through the IMU and GPS. We divide these 11 sequences into two parts: the 00-08 sequences for training and the 09-10 sequences for testing. We compare our camera pose estimation with two monocular ORB-SLAM namely full ORB-SLAM(using all frames of the driving sequence) and short ORM-SLAM(using 5 frames snippets). Moreover, we also compare our method with a state-of-the-art unsupervised framework which has done anything like working for depth estimation and camera prediction. As shown in Table 6, even though we use short sequences for training, our method outperforms these two competing baselines.

Table 6.

Absolute Trajectory Error on KITTI 2015 odometry dataset.

Method	Seq.09	Seq.10
ORB-SLAM(full)	0.014±0.008	0.012±0.011
ORB-SLAM(short)	0.064±0.141	0.064±0.130
GeoNet	0.012±0.007	0.012±0.009
D-SLAM	0.017±0.008	0.015±0.017
Ours	0.012±0.006	0.012±0.007

By comparing with traditional methods such as ORB-SLAM, we establish an end-to-end model to compute all frames of a video while ORB-SLAM creates keyframes to meet the real-time requirement. In addition, the multi-octave structure of CNN makes us extract high-level features of each frame in an automatic way. Therefore we can obtain dense image information while ORB-SLAM only used the sparse map. However, there are a few problems we are unable to solve now. ORB-SLAM can process monocular image sequences for indoor and outdoor scenes but we can only deal with scenarios similar to our training set. Because the strategy of analyzing big data to construct the model, the generalization ability of ORB-SLAM outperforms our model. Consequently, in our opinion, our model and ORB-SLAM are two different strategies and each method works better for different application types. Maybe the combination of these two strategies is the future research direction. In fact, there are already some methods [46] take advantage of deep learning technologies to extend the source of the scene depth information and improve the performance of the visual SLAM system.

5. Conclusion

In this work, we propose a jointly unsupervised learning framework for depth estimation and camera motion estimation. Stereo image sequences are used for training and monocular images are used for testing. The utilization of stereo image sequences cannot only overcome scale ambiguity for monocular depth estimation but also improve the accuracy of camera motion prediction for temporal image sequences. Compare with the previous works, the performance of our method is close to supervised learning approaches and better than most unsupervised methods. Moreover, experiments for generalization capability tests of the presented technique are conducted on multiple datasets.

There are still a few challenges to be mentioned. Although the results show that our method has superior accuracy compared to some existing unsupervised methods, but do not achieve state-of-the-art in all metrics. In addition, our unsupervised framework assumes the scene is static and there is no occlusion in the scene so that this method cannot handle dynamic objects. In the future, an extensive study of the objective function for depth estimation and CNN architecture for tackling dynamic objects will be taken into consideration.

Acknowledgements

We thank the professors of University of Essex, Dongbing Gu, and Huosheng Hu for their help. We also thank the anonymous reviewers for their positive comments. This paper is supported by the Fundamental Research Funds for the Central Universities under Grant 20720190129.

Declaration of Comepting Interest

None

References

[1] F. Fraundorfer, C. Engels, D. Nistér, Topological mapping, localization and navigation using image collections, in: Proceedings of the IEEE Conference on Intelligence Robots and Systems, 2007, pp. 3872-3877. doi:10.1109/IROS.2007.4399123.

[2] C. Chen, A. Seff, A. Kornhauser, J. Xiao, DeepDriving: Learning affordance for direct perception in autonomous driving, in: Proceedings of the IEEE Conference

on Computer Vision, 2015, pp. 2722-2730. doi:10.1109/ICCV.2015.312.

[3] D. Scharstein, R. Szeliski, High-accuracy stereo depth maps using structured light, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 195-202. doi:10.1109/cvpr.2003.1211354.

[4] J. Zhu, L. Wang, R. Yang, J.E. Davis, Z. Pan, Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps, IEEE Transctions on Pattern Analysis and Machine Intelligence. 33 (2011) 1400–1414. doi:10.1109/TPAMI.2010.172.

[5] S.G. Wysoski, L. Benuskova, N. Kasabov, Fast and adaptive network of spiking neurons for multi-view visual pattern recognition, Neurocomputing. 71 (2008) 2563–2575. doi:10.1016/j.neucom.2007.12.038.

[6] C. Hernandez, G. Vogiatzis, R. Cipolla, Multiview photometric stereo, IEEE Transctions on Pattern Analysis and Machine Intelligence. 30 (2008) 548–554. doi:10.1109/TPAMI.2007.70820.

[7] Y. Bahroun, A. Soltoggio, Online representation learning with single and multi-layer hebbian networks for image classification, in: Proceedings of the International Conference on Artificial Neural Networks, 2017, pp. 354-363. doi:10.1007/978-3-319-68600-4_41.

[8] W. Luo, A.G. Schwing, R. Urtasun, Efficient Deep Learning for Stereo Matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5695–5703. doi:10.1109/cvpr.2016.614.

[9] Y. Gao, A.L. Yuille, Symmetric non-rigid structure from motion for category-specific object structure estimation, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 408-424. doi:10.1007/978-3-319-46475-6_26.

[10] Y. Gao, A.L. Yuille, Exploiting symmetry and/or manhattan properties for 3D object structure estimation from single and multiple images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6718–6727. doi:10.1109/CVPR.2017.711.

[11] D. Eigen, C. Puhrsch, R. Fergus, Depth Map Prediction from a Single Image using a Multi-Scale Deep Network, Advances in Neural Information Processing Systems. 2014: 2366-2374. doi:10.1007/978-3-540-28650-9_5.

[12] D. Eigen, R. Fergus, Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, in: Proceedings of the IEEE Conference on Computer Vision, 2015, pp. 2650–2658. doi:10.1109/ICCV.2015.304.

[13] R. Garg, B.G. Vijay Kumar, G. Carneiro, I. Reid, Unsupervised CNN for single view depth estimation: Geometry to the rescue, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 740–756. doi:10.1007/978-3-319-46484-8_45.

[14] J. Xie, R. Girshick, A. Farhadi, Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 842–857. doi:10.1007/978-3-319-46493-0_51.

[15] W. Göbel, B.M. Kampa, F. Helmchen, Imaging cellular network dynamics in three dimensions using fast 3D laser scanning, Nature Methods. 4 (2007) 73–79. doi:10.1038/nmeth989.

[16] A. Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems. 86 (2012) 2278–2323. doi:10.1109/5.726791.

[17] J. Konrad, M. Wang, P. Ishwar, 2D-to-3D image conversion by learning depth from examples, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 16–22. doi:10.1109/CVPRW.2012.6238903.

[18] K. Karsch, C. Liu, S.B. Kang, Depth extraction from video using non-parametric sampling, IEEE Transactions on Pattern Analysis and Machine Intelligence. 36 (2014) 775–788. doi:10.1007/978-3-642-33715-4_56.

[19] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, ORB-SLAM: A Versatile and Accurate Monocular SLAM System, IEEE Transactions on Robotics. 31 (2015) 1147–1163. doi:10.1109/TRO.2015.2463671.

[20] J. Ma, H. Zhou, J. Zhao, Y. Gao, J. Jiang, J. Tian, Robust Feature Matching for Remote Sensing Image Registration via Locally Linear Transforming, IEEE Transactions on Geoscience and Remote Sensing. 53 (2015) 6469–6481. doi:10.1109/TGRS.2015.2441954.

[21] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, N. Navab, Deeper depth prediction with fully convolutional residual networks, in: Proceedings of the IEEE International Conference on 3D Vision, 2016, 239–248. doi:10.1109/3DV.2016.32.

[22] J. Li, R. Klein, A. Yao, A Two-Streamed Network for Estimating Fine-Scaled Depth Maps from Single RGB Images, in: Proceedings of the IEEE Conference on Computer Vision, 2017, pp. 3392–3400. doi:10.1109/ICCV.2017.365.

[23] H. Yan, S. Zhang, Y. Zhang, L. Zhang, Monocular depth estimation with guidance of surface normal map, Neurocomputing. 280 (2018) 86–100. doi:10.1016/j.neucom.2017.08.074.

[24] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, M. He, Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 07-12-June (2015) 1119–1127. doi:10.1109/CVPR.2015.7298715.

[25] A. Roy, S. Todorovic, Monocular Depth Estimation Using Neural Regression Forest, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5506–5514. doi:10.1109/cvpr.2016.594.

[26] F. Liu, C. Shen, G. Lin, Deep convolutional neural fields for depth estimation from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5162–5170. doi:10.1109/CVPR.2015.7299152.

[27] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, FlowNet 2.0: Evolution of optical flow estimation with deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1647–

1655. doi:10.1109/CVPR.2017.179.

[28] J.J. Yu, A.W. Harley, K.G. Derpanis, Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 3-10. doi:10.1007/978-3-319-49409-8_1.

[29] C. Godard, O. Mac Aodha, G.J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6602–6611. doi:10.1109/CVPR.2017.699.

[30] Y. Kuznietsov, J. Stückler, B. Leibe, Semi-supervised deep learning for monocular depth map prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2215–2223. doi:10.1109/CVPR.2017.238.

[31] Y. Hua, H. Tian, Depth estimation with convolutional conditional random field network, Neurocomputing. 214 (2016) 546–554. doi:10.1016/j.neucom.2016.06.029.

[32] T. Zhou, M. Brown, N. Snavely, D.G. Lowe, Unsupervised learning of depth and ego-motion from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6612–6621. doi:10.1109/CVPR.2017.700.

[33] Z. Yin, J. Shi, GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1983-1992. doi:10.1109/CVPR.2018.00212.

[34] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, L. Lin, Single View Stereo Matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 155-163. doi:10.1109/CVPR.2018.00024.

[35] A. Pilzer, D. Xu, M. Puscas, E. Ricci, N. Sebe, Unsupervised adversarial depth estimation using cycled generative networks, in: Proceedings of the IEEE International Conference on 3D Vision, 2018, pp. 587–595. doi:10.1109/3DV.2018.00073.

[36] S. Tulsiani, A.A. Efros, J. Malik, Multi-view Consistency as Supervisory Signal for Learning Shape and Pose Prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2897–2905. doi:10.1109/CVPR.2018.00306.

[37] M.W. Shields, M.C. Casey, A theoretical framework for multiple neural network systems, Neurocomputing. 71 (2008) 1462–1476. doi:10.1016/j.neucom.2007.05.008.

[38] E.P. Simoncelli, H.R. Sheikh, A.C. Bovik, Z. Wang, Image quality assessment: From error visibility to structural similarity, IEEE Transactions on Image Process. 13 (2004) 600–612.

[39] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the KITTI vision benchmark suite, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361. doi:10.1109/CVPR.2012.6248074.

[40] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes Dataset for Semantic Urban Scene Understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213-3223. doi:10.1109/CVPR.2016.350.

[41] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint arXiv: 1409.1556 (2014).

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 770-778. doi:10.1109/CVPR.2016.90.

[43] R. Li, S. Wang, Z. Long, D. Gu, UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning, in Proceedings of the IEEE International Conference on Robotics and Automation, 2017, pp. 7286-7291. doi:10.1109/ICRA.2018.8461251.

[44] S. Zhao, H. Fu, M. Gong, D. Tao, Geometry-Aware Symmetric Domain Adaptation for Monocular Depth Estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9788-9798. http://arxiv.org/abs/1904.01870.

[45] Y. Chen, H. Zhao, Z. Hu, Attention-based Context Aggregation Network for Monocular Depth Estimation, *arXiv preprint arXiv:* 1901.10137 (2019).

[46] M. Geng, S. Shang, B. Ding, H. Wang, P. Zhang, L. Zhang, Unsupervised Learning-based Depth Estimation aided Visual SLAM Approach. Circuits, Systems, and Signal Processing. 2019, 1–28. http://arxiv.org/abs/1901.07288.

[47] A. Handa, T. Whelan, J. McDonald, A.J. Davison, A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM, in Proceedings of the IEEE International Conference on Robotics and Automation, 2014, pp. 1524–1531. doi:10.1109/ICRA.2014.6907054.

[48] J. Xiao, A. Owens, A. Torralba, SUN3D: A database of big spaces reconstructed using SfM and object labels, in: Proceedings of the IEEE Conference on Computer Vision, 2013, pp. 1625–1632. doi:10.1109/ICCV.2013.458.

[49] J. Sturm, W. Burgard, D. Cremers, Evaluating Egomotion and Structure-from-Motion Approaches Using the TUM RGB-D Benchmark, in Proceedings of the International Conference on Intelligent Robots and Systems, 2012, pp. 299–319. doi:10.1.1.364.5940.

[50] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:* 1903.11027 (2019).

Delong Yang received the B.Sc. in School of Mathematical Science from Huaibei Normal University, Huaibei, China, and the M.Sc. in School of Science from Jimei University, Xiamen, China. He is currently pursuing the Ph.D. degree in the Department of Automation, Xiamen University, Xiamen, China. His research interests include robotics, computer vision, deep learning and image processing.

Xunyu Zhong received the M.E. degree in mechatronics engineering from Harbin Engineering University, Harbin, Heilongjiang, China, in 2007 and the Ph.D. degree in control theory and control engineering from Harbin Engineering University, in 2009. He is currently an Associate Professor with the Department of Automation, Xiamen University, Xiamen, China. His current research interests include robot motion planning, autonomous robotics, deep learning and computer vision.

Dongbing Gu (SM'07) received the B.Sc. and M.Sc. degrees in control engineering from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in robotics from the University of Essex, Colchester, U.K. He was an Academic Visiting Scholar at the Department of Engineering Science, University of Oxford, Oxford, U.K., from 1996 to 1997. In 2000, he joined the University of Essex as a Lecturer, where he is a Professor with the School of Computer Science and Electronic Engineering. His current research interests include robotics, multi-agent systems, cooperative control, model predictive control, visual SLAM, wireless sensor networks, and machine learning.

Xiafu Peng received the M.S. and Ph.D. degrees in control science from the Harbin Engineering University, in 1994 and 2001 respectively. He is currently a Professor with the Department of Automation, Xiamen University, Xiamen, Chian. His current research interests include the navigation and motion control of robots. Prof. Peng is a Fellow of the Fujian Association for the advancement of Automation and Power, and a Senior Member of the Chinese Institute of Electronics. He is the recipient of the provincial/ministerial Scientific and Technological Progress Award.

HUOSHENG HU (M’94–SM’01) received the M.Sc. degree in industrial automation from Central South University, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, U.K., in 1993. He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, U.K., leading the Robotics Research Group. His research interests include behavior-based robotics, human– robot interaction, embedded systems, multi-sensor data fusion, machine learning algorithms, mechatronics, pervasive computing, and service robots. He has authored over 500 papers in journals, books, and conferences in these areas. He is a fellow of the Institute of Engineering and Technology and the Institute of Measurement and Control, U.K., and a Chartered Engineer. He received a number of best paper awards. He has been the Program Chair or a member of the Advisory Committee of many IEEE international conferences, such as the IEEE ICRA, IROS, ICMA, and ROBIO. He currently serves as the Editor-in-Chief of the *International Journal of Automation and Computing* and the *Online Robotics Journal* and the Executive Editor of the *International Journal of Mechatronics and Automation*.