Pattern Recognition 106 (2020) 107378

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A survey on deep learning methods for scene flow estimation

Jiajie Liu[a], Han Li[a], Ruihong Wu[a], Qingyun Zhao[a], Yiyou Guo[b], Long Chen[a,*]

[a] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, PR China
[b] Department of Automation, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, PR China

ARTICLE INFO

Article history:
Received 30 September 2019
Revised 30 March 2020
Accepted 12 April 2020
Available online 16 May 2020

Keywords:
Scene flow
Optical flow
Depth estimation
Deep learning

ABSTRACT

Recently, computer vision has achieved remarkable accomplishments in many domains under the thriving of deep learning. Scene flow estimation turns from the classical manual feature construction to the deep convolutional neural network (DCNN) approaches. In this paper, we review recent works about scene flow, mainly focusing on DCNN methods. We present some milestones of scene flow in recent years, and categorize these methods into supervised and unsupervised based methods. Meanwhile, we also review some multi-task methods related to scene flow. At last, we present a performance comparison among different methods.

1. Introduction

Scene flow [1,2] estimation is one of the most fundamental problems in computer vision. It indicates a 3-dimensional motion field which consists of optical flow and depth estimation in a 3D scene. Scene flow was first introduced by Vedula et al. in 1999 [1], since then, many progresses have been achieved. Early researches on scene flow estimation mainly focused on variational based methods [3] which usually construct energy minimization function with many constrainted terms. Deep learning methods have achieved state-of-the-art results in many computer vision tasks, such as image classification [4,5], objects detection [6,7], segmentation [8–10], video object tracking [11,12], super resolution [13,14] etc. Thus, scene flow turns handcraft features [15,16] to deep learning features and yields promising results.

1.1. Problem definitions and basic notations

To make this survey easy to read, we first define some basic notations which remain consistent with the rest of this paper. Scene flow indicates a 3-dimensional motion field where each pixel is assigned with a 3-dimensional vector consisting of depth and motion information in horizontal and vertical directions. Generally, scene flow estimation can be decomposed into two subtasks, optical flow and depth estimation, which are essentially regression problem. As shown in Fig. 1, given consecutive stereo images in both left view and right view in time frames between $(t, t+1)$, we denote these four inputs as $(I_l^t, I_l^{t+1}, I_r^t, I_r^{t+1})$, respectively. The optical flow between (I_l^t, I_l^{t+1}) is further denoted as forward optical flow $f_l^{(t \rightarrow t+1)}$, and the inverse (I_l^{t+1}, I_l^t) is denoted as backward optical flow $f_l^{(t+1 \rightarrow t)}$. The disparity between (I_l^t, I_r^t) is denoted as d^t. Similarly, we have d^{t+1} regarded to (I_l^{t+1}, I_r^{t+1}). Finally, we use uppercase D to denote the depth (e.g. D^t, D^{t+1}). It's worth noting that not all the works utilize all the notations above.

1.1.1. Depth estimation

Depth estimation can be categorized into two different kinds of methods. Monocular depth estimation predicts depth using single view input. Binocular depth estimation (also called disparity estimation or stereo matching) takes stereo images as input to predict disparity.

Binocular Depth Estimation. Given two stereo images as input at time t, left view I_l^t and right view I_r^t, we can calculate the disparity between them. Disparity map indicates displacement between two corresponding patches in stereo images. This method simulates the way that our human perceives depth with our own eyes. We refer this disparity map as d^t. With lens focus length f and baseline b of camera, we can formulate the depth as follow.

$$D^t = \frac{f * b}{d^t}$$

Monocular Depth Estimation. Dating back to the variational methods time, it was very difficult and even impossible to estimate depth in the scene with single view input. Most of these methods are based on hand-crafted features by exploiting geometric priors. Thanks to the deep learning and big data, monocular depth estima-

[*] Corresponding author.
E-mail addresses: liujj73@mail2.sysu.edu.cn (J. Liu), lihan59@mail2.sysu.edu.cn (H. Li), wurh5@mail2.sysu.edu.cn (R. Wu), zhaoqy27@mail2.sysu.edu.cn (Q. Zhao), wuhee110@163.com (Y. Guo), chenl46@mail.sysu.edu.cn (L. Chen).

https://doi.org/10.1016/j.patcog.2020.107378
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

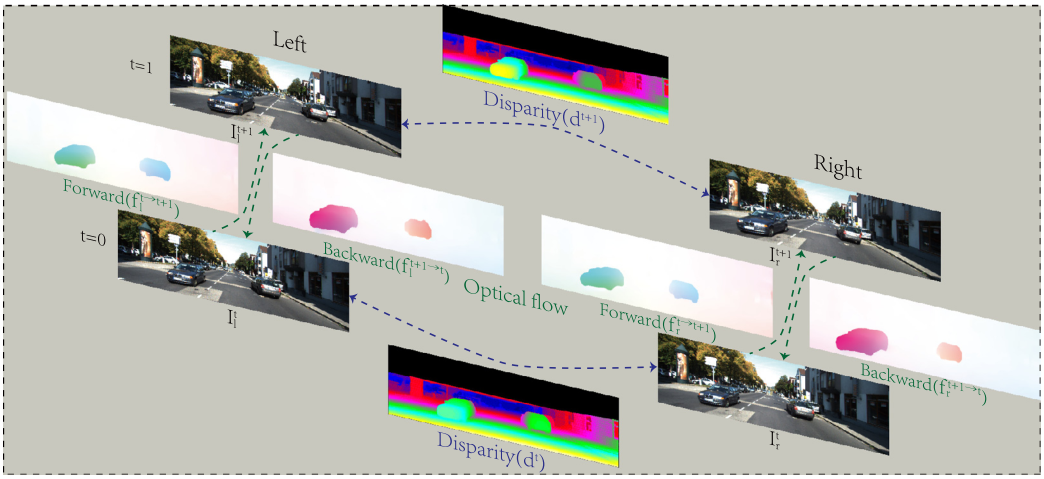

Left
t=1
Disparity(d^{t+1})
I$_l^{t+1}$
Forward(f$_l^{t\to t+1}$)
t=0
Backward(f$_l^{t+1\to t}$)
Optical flow
Right
I$_r^{t+1}$
Forward(f$_r^{t\to t+1}$)
Backward(f$_r^{t+1\to t}$)
I$_l^t$
I$_r^t$
Disparity(d^t)

Fig. 1. The scene projection between frames and views.

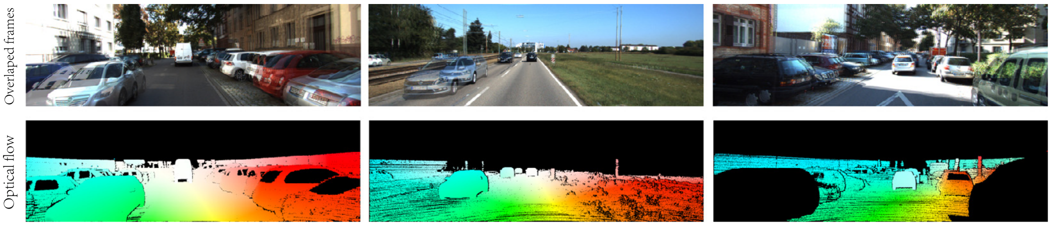


Fig. 2. Visulization of optical flow.

tion becomes possible. It can estimate depth D^t only using a single view image I_l^t, which is very beneficial to low budget devices.

1.1.2. Optical flow

Optical flow indicates a two-dimensional motion field including horizontal and vertical directions. Given two consecutive frames, optical flow describes the displacement of two corresponding points between the frames. In a sense, optical flow estimation is very similar to disparity estimation. More concretely, optical flow estimates the displacement of time sequence images which contain two directions while disparity estimates the displacement between stereo images with only horizontal displacement. Since these two tasks share some similarity, many methods [17,18] can work on both on them with minor changes, like FlowNet [19] and DispNet [20].

Optical flow between two frames occurred for two reasons, the dynamic moving objects and the motion of camera. In this paper, we refer the first mentioned of two as object optical flow which is denoted as f^{object} while the other is refered as scene static rigid flow which is denoted as f^{rigid}. Finally, we denote the full optical flow as $f = f^{full} = f^{object} + f^{rigid}$. Scene static rigid flow mainly occurs with the camera moving which is known as ego-motion. Ego-motion is very common in autonomous driving or robot navigation system. In the past, most works [15,19] study optical flow in a holistic approach. Recently, many works [21,22] focus on how to disentangle these two sub optical flow and yield promising results (Fig. 2).

1.2. Variational method vs deep learning method

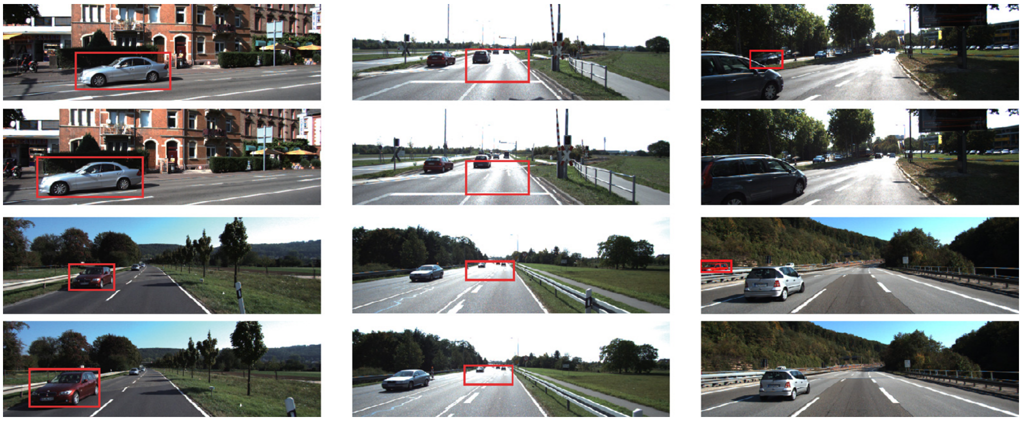
Before the era of deep learning, variational methods [1,15,23] basically dominate scene flow estimation. Variational methods usually turn the scene flow estimation problem to energy minimization problem with the assumptions that consist of multiple energy terms, like smooth term, gradient constancy term, and light constancy term. The gradient or light constancy assume that the brightness value of a pixel is constant over a time frame. And the smooth assumption assumes that the flow field of the same object should be very smooth. However, in the real scene especially outdoors, these constraints are not always necessary. For example, in an outdoor driving scene, various illumination and shadows can undermine the aforesaid assumptions and lead to severe errors. Meanwhile, most of the variational methods are based on the iterative method. It often comsumes a lot of time to calculate optical flow or disparity, which is intolerable for the autonomous driving or robot navigation systems.

With regard to deep learning, Dosovitskiy et al. introduced FlowNet [19] in 2015, and then DCNN based methods have been the research direction of scene flow estimation. Unlike variational based methods which heavily rely on explicit handcraft engineering for feature extraction, supervised DCNN based methods can implicitly learn the essential features which satisfy the aforesaid assumptions. DCNN based methods result in both accuracy and efficiency due to its powerful learning ability. In this paper, we mainly focus on the DCNN based methods of scene flow estimation.

1.3. Challenges

Scene flow estimation is a very challenging computer vision task as a well-known ill-pose problem. Here, we illustrate some of the most challenging problems.

Absent of data. The performance of DCNN heavily depends on the amount of data. Unlike other computer vision tasks, such as image classification [4], object detection [6] and autonomous driving [24], that can access large scale and diverse annotated data [4], there are very few datasets related to scene flow due to its particularity. As for depth estimation, ground truth data can be acquired from RGB-D camera or LiDAR. But there are still many limitations. RGB-D camera is limited to indoor scenario due to its short range perception sensor while LiDAR always produces sparse



(a) large displacement (b) reflection (c) occlusion

Fig. 3. Some challenge scenes of scene flow estimation.

depth map, especially on vertical direction. As for optical flow estimation, there is no sensor that can directly obtain ground truth data. At the present stage, the annotation of optical flow in the real scene relies on time-consuming and expensive manual calculation which is also prone to error. Because it requires the accurate calculation of displacement of each correspondent pixel between two frames. For the existing datasets, KITTI [25] provides real scenario ground truth data with the disadvantage of insufficient data and sparse annotations. FlyingChairs [19] and Sintel [26] provide large scale datasets with dense annotations. However, the fact that the data are not obtained from real scene means it lacks complexity and diversity(illumination, occlusion, reflection, etc.), which leads to a huge gap between research and industrial applications. Many researches try to tackle this problem, mainly focusing on unsupervised learning or self-supervised learning methods, and many progresses have been achieved.

Occlusion. In scene flow estimation, occlusion is very common, in which an object can be seen in this frame but disappears or part of it disappears in the next frame which happens both in optical flow estimation and disparity estimation. Since the core of scene flow estimation is to find the corresponding patch between consecutive stereo images, occlusion will lead to severe errors. Unlike humans who have strong prior knowledge for occlusion, many methods tend to perform poorly in this tricky situation. As shown in Fig. 3(c), a car is occluded by another.

Reflection. In a real world scenario, the illumination is constantly changing which may lead to severe reflection. Reflection corrupts surface texture of objects. As we all known that texture information is very crucial for scene flow estimation. As shown in Fig. 3(b), reflection on the road makes the road become textureless.

Large displacement. In general, large displacement means the corresponding patch between two frames in a relatively long range. In optical flow, a large displacement is very prevalent in real scenes. Fast moving objects like cars running on highway will lead to large displacement. Similarly, too long a time interval between two captured frames may result in large displacement too. Large displacement is one of the biggest challenges in scene flow. As shown in Fig. 3(a), high speed car occurs large displacement motion. Many methods tackle this issue by introducing the coarse-to-fine procedure to gradually estimate optical flow and achieve promising results.

Joint Learning. As aforementioned, scene flow consists of two subtasks, depth and optical flow estimation. Currently, the mainstream of researches tackles these two problems in an isolation manner which means they are trained in two different models independently. When it comes to the practical application, it takes two separate models to estimate depth and optical flow, then it combines them as a 3-dimensional motion field. This can be deemed as a two-stage approach. The disadvantages of this approach are obvious. Two individual models mean increasing in both parameters and computation which can be fatal for a device with limited budget capacity.

2. Common network architectures

Dosovitskiy et al. introduce the DCNN based FlowNet [19] to estimate optical flow. Since then, DCNN based methods of optical flow or stereo matching have achieve many improvements. Most of these works follow some basic principles. A typical coarse-to-fine procedure to estimate optical flow or disparity can be summarized as 1) Feature extraction using DCNN. 2) Constructing cost volume based on two feature maps. 3) Initial estimation for optical flow or disparity. 4) Refining optical flow or disparity estimation with residual learning. Next, we elaborate on some basic components of this framework.

2.1. Encoder-decoder network

Encoder-decoder network is prevalent among many computer vision tasks. A classical encoder-decoder network consists of two parts, encoder and decoder. Given images as input, encoder [27] extracts useful features by gradually down-sampling the feature map from high resolution to low resolution. The down-sample operation has two main advantages. First, down-sample operation leads to low resolution feature map which can significantly reduce memory footprint and computation. More importantly, down-sample is an effective way to enlarge the receptive field which provides much more context information with strong semantic information. After encoder, decoder is employed to gradually restore the resolution of feature map with up-sample operations which include deconvolution or bilinear interpolation. Meanwhile, since low resolution feature map means absent of details, skip connection between encoder and decoder is often used to reconstruct the lost details.

2.2. Cost volume

Cost volume is a kind of feature map that indicates the matching cost between the two patches among input images. Low cost means more similarity between these two patches. Cost volume is a powerful tool in variational methods. They usually explicitly

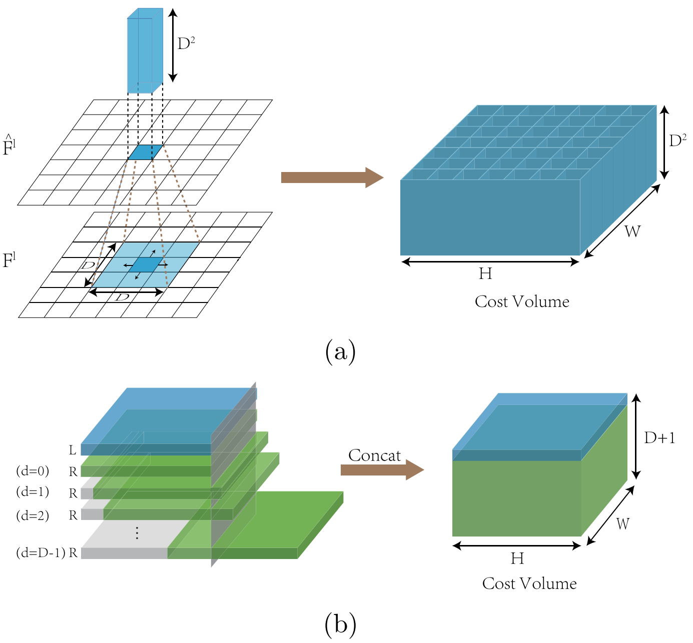


Fig. 4. Two ways to construct cost volume.

model the cost volume based on the similarity of the two input images. In DCNN, constructing cost volume can be in a more implicit manner. Given two images as input, due to the powerful adaptive representations of DCNN, the network can automatically model the cost volume without any prior knowledge. FlowNet [19] is the first DCNN to employ cost volume. Obviously, the simplest way to build cost volume is to concat two images directly along the channel dimension as an input to DCNNs, such as FlowNetS [19]. The concatenated images will form cost volume through convolution neural network.

Correlation layer which is introduced in FlowNetC [19] is a more intricate way to construct cost volume. In FlowNetC, they use the backbone network to extract features from two input images and get two feature maps. Then these feature maps are used to construct cost volume through the correlation layer. As shown in Fig. 4(a), given two feature maps $f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}^c$, and width, height, the number of channels denote as w, h, c, respectively. Then the correlation of two patches center at x_1 and x_2 in feature map f_1, f_2 can be formulated as follow:

$$c(x_1, x_2) = \sum_{o\in[-k,k]\times[-k,k]} \langle f_1(x_1 + o), f_2(x_2 + o)\rangle \qquad (1)$$

The above operation is identical to one step of a convolutional operation but with a patch convolving patch manner which means there are no parameters needed to be trained. Given a square patch of size $K := 2k + 1$, K is identical to kernel size of the convolution operation. Also, it is necessary to set a maximum displacement d for each location x_1 to avoid heavy computation. Thus, correlation $c(x_1, x_2)$ only performs in a neighborhood of size $D := 2d + 1$. Finally, the output feature map with size($w \times h \times D^2$) serves as cost volume.

2.3. Warping strategy

As we mentioned above, many methods are based on the coarse-to-fine scheme to estimate optical flow or disparity. Warping strategy [3] plays an important role in this scheme. FlowNet2.0 [28] and SPyNet [29] employ warping strategy to refine the optical flow estimation from the previous sub-network. Unlike FlowNet2.0 [28] which warps the second image I_2 via flow to $\tilde{I_2}$, PWCNet [30] warps the second feature map F_2 to $\tilde{F_2}$ and has a better result. Given the second imageI_2 (or feature map) and optical flow $f = (uv)^\top$ or disparity, warping $I_2(x, y)$ to $\tilde{I_2}$ can be formulated as follow:

$$\tilde{I_2}(x, y) = I_2(x + u, y + v) \qquad (2)$$

where x, y denote the location index of each pixel. Theoretically, in ideal conditions, given a precise enough optical flow, $\tilde{I_2}$ should be equal to I. Thus, the rest of the network can focus on minimizing the error $e = \left\| \tilde{I_2} - I_1 \right\|$ which can be treated as an coarse-to-fine training scheme.

3. Supervised methods for optical flow and depth estimation

In this section, we will review some of the most innovative works on supervised methods for optical flow and depth estimation. With the strong fitting ability of DCNN, supervsied methods achieves many astonshing achievements on optical flow and depth estimation. Many of these methods take groundtruth data for training the network.

3.1. Optical flow estimation

FlowNet family [19] is the first attempts to directly estimate global dense optical flow. FlowNet directly models the optical flow based on supervised DCNN learning. The framework of FlowNet is a typical encoder-decoder network which is very similar to U-Net [31]. It follows most of the principles of current common encoder-decoder networks, where an encoder gradually down-samples the feature maps for feature extraction followed by a decoder network to gradually restore the feature map to high resolution. Also, skip connections and multi-scale predictions are also adopted to stabilize and accelerate the training. Multi-scale predictions can provide extra supervised signal for training and the skip connections between low- and high-level features can provide the information which lost in the process of down-sampling operation.

There are two versions of FlowNet, FlowNetC and FlowNetS, where C stands for correlation layer and S stands for simple. The difference between these two versions is the way to construct cost volume. In FlowNetS, a naive way is to construct cost volume by simply concatenating the two images along the channel dimension. And the resulted 6 channels image is treated as input to the network. As for FlowNetC, it separates the shallow part of the encoder into two streams with shared parameters, and each takes one image as input to produce its corresponding feature maps. These two feature maps construct a cost volume through correlation layer. And the cost volume serves as input to the rest of the network which is identical to FlowNetS's. The authors suggest that the correlation layer can make the network better to learn the corresponding part between two frames. The result supports that with correlation layer, FlowNetC produces more robust optical flow than its counterpart. However, there is still a small gap when comparing to variational methods, but it does set a milestone in the research of optical flow especially with its real-time inference time.

FlowNet2.0 [28] is built upon FlowNet [19] and consists of multiple sub-networks. More concretely, there are two streams in FlowNet2.0, a stream predicting the large displacement optical flow while the other predicts the small displacement optical flow. The large displacement stream named FlowNet-CSS consists of three stacked FlowNetS, a FlowNetC followed by 2 FlowNetS. After FlowNetC producing the optical flow, it warps the second frame using the estimated optical flow and computes the brightness loss between the first frame and the warped second frame. The following FlowNetS takes these elements and the original images as inputs. The second FlowNetS takes the exact same process and produces a large displacement optical flow. The structure of the small displacement stream is named FlowNet-SD. The authors create a new dataset with small displacement to train the FlowNet-SD to learn small displacement optical flow. After that, a fusion network produces the final optical flow by fusing the small and large displacement optical flow. FlowNet2.0 stacks multiple sub-networks to learn the residual flow, thus the memory footprint and computation are pretty heavy for real world application.

SPyNet [29] proposes a lightweight framework which combines classical spatial-pyramid formulation with deep learning for optical flow estimation. SPyNet tries to estimate large displacement motion and precise sub-pixel optical flow. In order to achieve these goals, SPyNet constructs image pyramid to estimate optical flow in a coarse-to-fine manner. Given consecutive frames as input, SPyNet resizes the size of images to match the size of each pyramid level. At each pyramid level, it warps the second image using the up-sampled flow which is from a deeper layer. Then a CNN takes the first image and the warped second image as inputs to estimate the residual optical flow. It up-samples the estimated flow and passes it to the next level. SPyNet has five sub-networks corresponding to five pyramid levels. By constructing this image pyramid network, motions with different scales are fused together, which enable the networks to capture both large motion and detail information.

PWC-Net [30] is also a lightweight network for optical flow estimation. As the name suggests, PWC-Net consists of three major parts, feature pyramid extractor, warping layer and cost volume layer. As we well known, a typical encoder network gradually down-samples the input to extract feature. The feature maps at different stages have different resolutions which can be deemed as a pyramid. Practically, in PWC-Net, the two input frames are encoded via the same Siamese network. At the end of each stage of the network, it outputs two feature maps which are considered as a level of the pyramid. PWC-Net predicts optical flow at each level of the pyramid. More concretely, at each level, the second feature map is warped towards the coordinate of the first frame via an up-sampled optical flow which is predicted from the previous scale. Then it constructs a cost volume between the first feature map and the warped second feature map using correlation layer which is identical to the correlation layer in FlowNet [19]. The cost volume, the first feature map and the up-sampled optical flow are served as inputs to an optical flow estimator to output the optical flow at this level. To further improve the quality of optical flow, PWC-Net adopts a sub-network called context network at each level to refine the estimated optical flow. The context network can provide much more context information by adopting dilated convolution [32] which can effectively enlarge the receptive field of the networks without losing detail information. Compared to FlowNet [19] or FlowNet2.0 [28], PWC-Net adopts multi-level warping strategy and cost volume. And PWC-Net performs warping strategy on feature map instead of the image. Compared to SPyNet, PWC-Net adopts feature pyramid rather than image pyramid for multi-scale information. All of these techniques make the prediction of optical flow more accurate in PWC-Net.

LiteFlowNet [33] is a compact and accurate framework for optical flow estimation. It consists of two sub-networks, NetC and NetE. NetC is an encoder network for feature description by constructing feature pyramid. NetE is a decoder network for cascaded flow inference and flow regularization. Given two consecutive frames I^t and I^{t+1}, NetC produces a pair of 6-levels feature pyramid. Similar to PWC-Net, at each pyramid level, optical flow is an inference from feature maps F^t and F^{t+1} of frames I^t and I^{t+1}. NetC also performs feature warping at each level via higher level optical flow. In NetE, it proposes the cascaded flow inference to inference optical flow. It divides the inference into two phases named descriptor matching and sub-pixel refinement. The first flow inference employs correlation layer to construct a cost volume for pixel-level optical flow. Then the second inference produces sub-pixel optical flow using the warped feature map. The authors suggest that this progressive cascade flow inference allows early correction of the estimated optical flow. It can be deemed as data fidelity in energy minimization methods. Thus, it has less error when passing the estimated flow to the next pyramid level. After that, the authors propose a flow regularization layer to sharpen the vague flow boundaries and reduce the undesired artifacts. Further, the feature-driven local convolution(f-lcon) is implemented in flow regularization layer. The f-lcon is inspired by local convolution [34] but with a more generalized application. The f-lcon can smooth the flow field but with sharpening boundaries.

LiteFlowNet [33] is a compact and accurate framework for optical flow estimation. It consists of two sub-networks, NetC and NetE. NetC is an encoder network for feature description by constructing feature pyramid. NetE is a decoder network for cascaded flow inference and flow regularization. Given two consecutive frames I^t and I^{t+1}, NetC produces a pair of 6-levels feature maps as feature pyramid. Similar to PWC-Net, at each pyramid level, optical flow is an inference from feature maps F^t and F^{t+1} of frames I^t and I^{t+1}. NetC also performs feature warping at each level via higher level optical flow to allow the network to learn residual optical flow. In NetE, LiteFlowNet proposes cascaded flow inference to inference optical flow which divides the inference into two phases named descriptor matching and sub-pixel refinement. The first flow inference employs correlation layer to construct a cost volume and uses it to produces pixel-level optical flow. Then the second inference produces sub-pixel optical flow using the warped feature map. The authors suggest that this progressive cascade flow inference allows early correction of the estimated optical flow which can be deemed as data fidelity in energy minimization methods. Thus, it has less error when passing the estimated flow to the next pyramid level. After that, the authors propose a flow regularization layer to sharpen the vague flow boundaries and reduce the undesired artifacts. The feature-driven local convolution(f-lcon) is implemented in flow regularization layer. The f-lcon is inspired by local convolution [34] but with a more generalized application. The f-lcon can smooth the flow field but with sharpening boundaries.

3.2. Binocular depth estimation

DispNet [20]follows the architecture of FlowNet [19] with minor adaptations. It changes the correlation layer to only cover a horizontal direction because disparity only has one direction. However, DispNet does not generalize well enough on unseen data.

CRL [35] is a two-stage method for stereo matching. CRL contains two sub-networks, DispFulNet and DispResNet. The DispFulNet is built upon DispNet [20] but with minor differences that DispFulNet produces full-resolution disparity map by incorporating extra up-sample modules. DispResNet is also built upon DispNet but with residual signals [27]. Specifically, given stereo images as input, DispFulNet produces a full resolution disparity map and the training loss, then CRL warps the right view image via the disparity. The DispResNet takes left view image, warped right view image, the loss, and the disparity map as inputs. In the decoder part of the DispResNet, each level incorporates a down-sampled disparity map as residual learning. During the two-stage training scheme, CRL first trains the DispFulNet until it converged. Then CRL fixes the parameters of the DispFulNet to train DispResNet as residual training. The authors also suggest this two-stage learning method provides more effective refinement comparing to directly learning.

GC-Net [36] proposes an end-to-end deep learning method to estimate disparity from rectified stereo images. The key component of GC-Net is the cost volume. It takes a different way to constructs a cost volume unlike DispNet [20] using correlation layer. Given max disparity and two feature maps with the size of $(h \times w \times channels)$, it constructs a cost volume with the size of $(h \times w \times (maxdisparity + 1) \times channels)$ via offsets and concatenates the second feature map with displacement from 1 to *maxdisparity* to the first feature map. As shown in Fig. 4(b), the authors suggest that this construction way can allow the network to learn an absolute representation and carry this to the cost volume. After that, an encoder-decoder takes the cost volume as input. GC-Net adopts 3D convolution and 3D deconvolution operation within encoder-decoder network. The 3D convolutional kernels convolve over the cost volume with dimension of $(h \times w \times (maxdisparity + 1))$ to extract the feature which contains spatial and disparity information. Finally, the decoder outputs a feature maps with size of $(h \times w \times disparity)$, the value of the location at(x, y, d) denotes the cost of disparity belong to d. This process is similar to the other classification task. Meanwhile, in order to produce sub-pixel disparity, the authors propose soft argmin:

$$soft\ argmin := \sum_{d=0}^{D_{max}} d \times \sigma(-c_d) \tag{3}$$

where c_d denotes the costs belongs to disparity d and σ denotes softmax operation. The soft argmin is fully differentiable, allowing end-to-end training.

PSMNet [37] proposes a framework for finding correspondence within ill-posed regions by exploiting context information. PSMNet contains three main modules, a basic CNN with dilated convolution [32] in the last two layers, a spatial pyramid pooling (SPP) [38] following the basic CNN and a 3D CNN which takes cost volume as input. Like many other methods, a basic CNN in the shallow part of PSMNet is used to extract low-level features. SPP is a very powerful tool in semantic segmentation task since it can extract much more context information. PSMNet is the first network that employs SPP for disparity estimation. The SPP module enables the PSMNet to learn rather region-level features than pixel-level features by incorporating global context information. After the SPP module, the two feature maps corresponding to two frames are used to construct a cost volume which is identical to GC-Net's [36]. In order to better integrate the information from disparity dimension and spatial dimension, PSMNet introduces a stacked hourglass structure for cost volume regularization. It contains 3 hourglasses which utilize 3D convolution and 3D deconvolution. Each of the three hourglasses is an encoder-decoder sub-network and produces a disparity map and its corresponded loss. The authors also alter the common used L1 loss function to smooth L1 loss function which can be formulated as:

$$L(d, \hat{d}) = \frac{1}{N} \sum_{i=1}^{N} smooth_{L_1}(d_i, \hat{d}_i) \tag{4}$$

where

$$smooth_{L_1} = \begin{cases} 0.5x^2, & if|x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \tag{5}$$

where N denotes the number of pixels, d and $\hat{d}$ are ground truth and estimated disparity. As comparing to L2 loss or L1 loss, smooth L1 loss is more robustness and low sensitivity to outliers. Because in the L1 loss, the gradient is always set to 1 or -1. When in the situation that the x with a relative small value, the smooth L1 loss is much more robust because the value of the gradient will decrease.

GA-Net [39] proposes two matching cost aggregation layers for stereo matching, semi-global aggregation layer (SGA) and local guided aggregation layer (LGA). SGA and LGA are used in GA-Net to capture local and global cost dependencies respectively. It can replace the 3D convolution for 4D cost volume which is widely used in GC-Net [36] and PSMNet [37]. 3D convolution is very useful in stereo matching since it can convolve both spatial and disparity dimension. However, 3D convolution still has disadvantages of heavy time- and memory-consuming. Meanwhile, the proposed SGA and LGA can effectively reduce the inference time and memory footprint without undermining the quality of stereo matching. SGA can be deemed as a differential approximation of semi-global matching (SGM) [40]. It aggregates matching cost in different directions of the whole image. It allows the network to learn accurate prediction within the ill-posed regions. The proposed LGA aims at refine the thin structures and object boundaries and recovers the details which may be lost in the process of down-sample and up-sample operation.

3.3. Monocular depth estimation

MSDN [41]. This work proposes a novelty method to estimate depth map using a single image. It contains two sub-networks, one to predict a coarse depth map named global coarse-scale network and the other to predict a refined depth map named local fine-scale network. Specifically, given a single view image, the global coarse-scale network which is built upon AlexNet [42] estimates global coarse depth map. Then the local fine-scale network which contains three convolution layers takes both the original image and the coarse map as input to estimate refined local depth map. The authors also propose the scale-invariant error to measure the relationships between patches in the scene. The scale-invariant error can be formulated as:

$$D(y, y^*) = \frac{1}{2n} \sum_{i=1}^{n} (logy_i - logy_i^* + \alpha(y, y^*))^2 \tag{6}$$

where $\alpha(y, y^*) = \frac{1}{n}\sum_i (logy_i^* - logy_i)$. The proposed error has the advantage that it is irrespective of the absolute global scale. During training, the authors first train the global coarse-scale network. Then they fix its parameters and train the local fine-scale network.

AVPL [43]. In this work, the authors propose three main modules to improve the monocular depth estimation, affinity layer, vertical pooing and label enhancement. 1) the authors argue that most of the previous work neglect the relationships between neighboring pixels. Thus, the authors propose the affinity layer to directly model the correlation between the absolute features of two image pixels. The affinity layer can be formulated as:

$$\nu(x)_{m,n} = f(x) \cdot f(x+(m,n));\ m,n \in [-k,k] \tag{7}$$

where $\nu(x) \in R_{(2k+1)\times(2k+1)}$ describes the affinities of pixel x within square region of $(2k+1) \times (2k+1)$. $f(x)$ denotes the absolute feature vector. In practical, the authors only employ the affinity layer in the shallow part of the network to reduce heavy computation. 2) The authors recognize a special pattern in real scene that the change of depth of vertical direction is much larger than horizontal's. Thus, the vertical pooling layer is implemented to obtain local context information in vertical direction. Vertical pooling is built upon average pooling but with kernels of size of $H \times 1$. In order to capture multi-scale context information, the authors employ a set of vertical pooing with kernels of size 5×1, 7×1, 11×1 and 11×1.

CSPN [44] introduces convolutional spatial propagation network for monocular depth estimation. Inspired from SPN [45] which proposes the linear spatial propagation network to learning affinity for segmentation tasks, CSPN extends this technique to monocular depth estimation. Different from SPN which propagates the whole images with four directions separately, CSPN only propagates a local area but towards all directions at each step. In practical, CSPN adopts the convolutional operation to implement the propagation process which can significantly reduce the computation complexity. More concretely, in an encoder-decoder network for monocular depth estimation, the CSPN module is inserted into the last part of the decoder to refine the initial depth map. Since the fined grained information is crucial for learning affinity matrix, the authors also propose two up-sample modules to recover fined grained information which is lost in down-sample within the decoder network.

DORN [46] introduces the ordinal regression optimizer to discretize and recast depth network learning as an ordinal regression problem with the tool of spacing-increasing discretization (SID) strategy. In the previous works [36,37], they tend to dicretize the depth value into uniform space. DRON suggests that as the value of depth increase, the feature of depth becomes less informative. It indicates that the raise of estimation error is consistent with the raise of the value of depth. Thus, they adopt the SID strategy to uniformly spread a given depth interval in the log space within the region with large depth value. The SID can be formulated as:

$$SID: t_i = e^{\log(\alpha)+\frac{\log(\beta/\alpha)\cdot i}{K}} \tag{8}$$

Where $t_i \in \{t_0, t_1, \ldots, t_K\}$ are dicretization thresholds and $[\alpha, \beta]$ denotes the depth interval and K denotes the number of sub-intervals. SID allows the network become more sensitive to small depth value which gives more accurate depth estimation of small and medium depth.

4. Unsupervised or self-supervised methods for optical flow and depth estimation

Though supervised methods achieve impressive performance on most of the benchmarks, such as KITTI [25], Sintel [26]. When it comes to the real scene, the methods which only trained on limited data tend to perform poorly. The main reason is that supervised methods heavily rely on the amount of labeled data. Many methods try to tackle the problem of absent labeled data by implementing unsupervised or self-supervised methods. These methods are not bounded by the ground truth data which allow them to access large scale real scene data. In this section, we review some of the most novelty methods including unsupervised and self-supervised methods.

4.1. *Optical flow estimation*

DSTFlow [47] is an early attempt to estimate optical flow using unsupervised deep learning. DSTFlow consists of three key components, localization net, sampling net and the loss function layer. Localization net which is built upon FlowNetS [19] takes the consecutive frames as input to produce pixel-level offsets which are deemed as optical flow. The pixel-level offset is a two-channel feature map, which contains the horizontal and vertical offsets which is corresponding to the first frame. The sampling net warps the second frame via the estimated optical flow. Then the loss layer, which based on the photometric loss with an additional smooth term, can be formulated as:

$$L_1 = \int_\Omega \Psi(|I_2(x+w) - I_1(x)|^2 + \gamma|\Delta I_2(x+w) - \Delta I_1(x)|^2)dx \tag{9}$$

It models the grey and gradient constancy, where $\Psi(s) = \sqrt{(s^2 + 0.001^2)}$. And the smooth term:

$$L_2 = \int_\Omega \Psi(|\Delta u(x)|^2 + |\Delta \nu(x)|^2)dx \tag{10}$$

The key idea in DSTFlow is to minimize the dissimilarity between the first frame and the warped second frame. Many subsequent unsupervised methods also follow this idea.

OAULOF [48]. This work proposes two techniques for unsupervised learning optical flow, an end-to-end trainable module to explicitly model the occlusion, a new warping way that facilitates the optical flow to learn the large displacement. The authors suggest that large displacement and occlusion will limit the performance of previous unsupervised learning methods [49]. For the problem of occlusion, the authors explicitly model the occlusion by the forward warping with backward optical flow. More concretely, a FlowNetS [19] is used to produce bidirectional optical flow. Then it warps the backward flow via forward flow to generate occlusion map. The authors also moderate the original warping with a larger search space which aims to tackle the large displacement motion.

Multi-Frame Optical Flow with Occlusions [50]. This work proposes a framework for unsupervised learning to reason occlusion with multi-frames. Unlike usual unsupervised learning methods [47,49] that take two consecutive frames as input, this work takes three frames which aimimg to improve the photometric loss and explicitly reason about the occlusion. Given three consecutive frames (I^{t-1}, I^t, I^{t+1}), which represent the past frame, reference frame and future frame, respectively. The goal is to predict optical flow from I^t to I^{t+1} by leveraging I^{t-1}. Then a network which is built upon PWC-Net [30] takes these frames as input. Unlike the original PWC-Net, at each level of the pyramid, this network constructs two cost volumes, one related to the past frame and the reference frame while the other related to reference frame and future frame. These two cost volumes allow the network to reason the occlusion. Then three separate decoders take the stacked cost volumes as input and produce the past flow, future flow, occlusion map, respectively. The last layer of occlusion decoder employs softmax operation to produce a probability map of occlusion. The occlusion map is integrated into the two optical flow for providing more accurate optical flow. Meanwhile, an additional constant velocity loss is proposed to better regularize the training. The constant velocity loss is under the assumption that the motion between these three frames is linear which means that the past and future optical flow should be equal but with a different direction. The constant velocity loss can be formulated as:

$$L_{cv} = \sum_{p\in\Omega} \rho(F^{past}(p) + F^{future}(p)) \tag{11}$$

when the past flow equal to the future flow with a negative value, the loss L_{cv} should approximate to zero.

DDFlow [51] is an unsupervised method for optical flow estimation using data distillation [52] technique. Comparing to many previous unsupervised methods [52], DDFlow is a data-driven method, and can estimate optical flow in the occluded regions. DDFlow considers that widely used photometric loss would provide misleading information within the occluded regions. DDFlow tries to tackle this problem by explicitly extracting and distilling the occlusion information. The overall framework of DDFlow consists of two main modules with the same architecture, a teacher network and a student network both of which are built on the PWC-Net [30]. The teacher network predicts the optical flow within non-occluded regions. The estimated optical flow is served as ground truth and guides the student network to predict occluded and non-occluded optical flow. More concretely, given consecutive frames as inputs, it produces bidirectional optical flow which consists of forward and backward optical flow. Then the forward-backward consistency check is performed to detect the occluded region and produces an occlusion mask. Then a valid mask is computed that indicates that pixels are occluded in the cropped image while non-occluded in the original image. Then the valid mask is used to guide the student network to learn optical flow of occluded or non-occluded regions. During inference phrase, only the student network is needed. DDFlow provides a new direction for optical flow estimation though the performance still cannot match up with other supervised methods.

SelFlow [53] is a self-supervised method for optical flow estimation. SelFlow extracts reliable optical flow in non-occluded regions. Then it treats the estimated optical flow as ground truth to train the network to predict optical flow in occluded regions. SelFlow is very similar to DDFlow [51], DDFlow works well for regions near the image boundaries but still can't handle well with all the possible occluded regions. Thus, SelFlow proposes a technique named superpixel-based [54] occlusion hallucination which has the ability to reason about the optical flow of all possible occluded regions. SelFlow consists of two identical sub-networks, named NOC-Model and OCC-Model. They are both built upon PWC-Net [30] with a minor alteration, producing bidirectional optical flow at each pyramid level. A forward-backward consistency check is implemented to obtain bidirectional optical flow for occlusion detection. The region is considered occluded if it fails the check. SelFlow adopts occlusion hallucination technique to further improve the quality of the occlusion mask. It randomly generates superpixels and fills them with noise which will be considered as occluded regions. This process hallucinates the occlusion and makes the predicted occlusion maps much more robust. Then, the NOC-model learns the reliable optical flow from non-occluded regions, and the estimated reliable optical flow is used to guide the OCC-Model to predict optical flow within occluded regions. With extra supervised training, SelFlow achieves the state-of-the-art results on KITTI and MPI Sintel datasets.

4.2. Depth estimation

Left-Right Consistency [55]. This work proposes a novelty method to estimate depth using only a single image. Unlike many other supervised methods [20,35,37] rely heavily on the quantities of aligned ground truth depth data, this work only needs the easier-to-obtain binocular stereo footage for training. The key idea is to estimate accurate disparity by warping the left image to match the right one, and enforces consistency between the disparity corresponded to both left and right views. In order to achieve this goal, the authors propose two important tools, appearance matching loss and left-right disparity consistency loss. The appearance matching loss can be deemed as photometric image loss. It consists of two terms, a $L1$ loss and a single scale SSIM [56]. SSIM, *a.k.a.* structural similarity index, is a measure of the similarity of two images. Unlike the widely used $L2$ loss, SSIM is similar to our human visual system and sensitive to local structural changes. This work adopts a simplified version of SSIM which replaces the Gaussian filter with a 3×3 block. In order to achieve more accurate disparity, appearance is clearly not enough. Thus, the authors propose the left-right disparity consistency loss:

$$C^l_{lr} = \frac{1}{N}\sum_{i,j}\left|d^l_{ij} - d^r_{ij+d^l_{ij}}\right| \tag{12}$$

where d denotes the disparity of left and right reviews. It takes both the left and right image disparity as inputs. This loss function enforces the left-view disparity map to be equality to the warped right-view's. However, this method does not work well when ports from KITTI [25] to the Cityscapes [57] dataset. And it performs poorly within the textureless region, such as reflective or transparent surface.

SFMLearner [58] is a prior work of unsupervised learning for monocular depth estimation and camera pose estimation using multi-views unlabeld viedo clips. It explicitly synthesizes the scene structure using the depth and camera pose information. To be specificy, SFMLearner has two sub-networks, one for depth estimation and the other for camera pose estimation(ego-motion). Given a set of images containing target view and multiple nearby views, DispNet takes the target view for depth estimation D_t while a pose net takes all the images as input for 6-DoF relative camera poses estimation $(T_{t\to t-1}, T_{t\to t+1})$. Then it uses the depth and camera pose to warp the nearby views toward the target view. Photometric reconstrction loss is used to measure similarity between the target view and the warped target views and provids the supervised training signal. There are several limitations of the above methods that it cannot handle the regions with dynamic objects or occlusion/non-occlusion situations. Thus, the authors proposes a explainablity network to mask out those regions. It takes the pose net as the encoder part and then branch out for multi-scale explainity masks. SFMLearner is the first attempt to predict the depth and camera pose with unsupervised manner. Though its application is limited to the static scenarios, it did inspire many succssors which try to explicitily model the dynamic scene. We will discuss some of these methods in Section 5.

Depth Prediction Without Sensors [59]. This work is inspired by SFMLearner [58]. Unlike SFMLearner, it models the corresponding motion in the scene as independent transformation, such as rotation and translation which will be used to model 3D geometry and estimate the motion of all objects. By decomposing the scene into 3D geometry and a single object, the network can learn better feature regarded to depth and ego-motion, especially the dynamic objects in the scene. The key idea in this work is that it incorporates structural information into the network. In other words, the estimated depth map is not directly obtained by a DCNN, but is obtained by the reconstruction 3D scene composed of static background and moving objects. Comparing to previous unsupervised methods which learn depth maps from monocular video, this method can accurately restore the depth maps of moving objects. Meanwhile, a seamless online refinement technique is also proposed by the authors to further improve the quality of depth estimation.

5. Towards joint learning of optical flow and depth estimation

This section we mainly focus on the join learning of Optical Flow and Depth Estimation. As we discussed above, many methods focus on one of these two subtasks and achieve many improvements. Their prior works provide a great foundation for the following-up joint learning researches. These methods usually extract the structured information of the scene to better estimate scene flow, unlike many unstructured methods which consider each pixel as an individual. Also, we categorize these methods into supervised and un-supervised based methods.

5.1. Supervised methods for scene flow estimation

ISF [21]. As we all well-known that large displacement is very challenging in scene flow estimation, especially in an autonomous driving scene with high speed cars. ISF, which means instance scene flow, tackles the large displacement problem by treating each of these driving cars as a rigid instance instead of a group of pixels. ISF incorporates DCNN and variational methods to estimate scene flow for the autonomous driving scene. It treats each participant in the driving scene as a rigid instance. It based on the idea that the clustered pixels belonged to the same object with a high chance to act like a whole. Specifically, an instance segmentation mask is employed to select the regions to be treated as rigid objects in frame I^t and I^{t+1}. Thus, the scene flow estimation problem is turned into instance level matching problem instead of pixel level problem. Beyond instance level visual cue for scene flow, ISF also investigates the performance of how other visual cues (2D bounding box, 2D instance segmentation, 3D object coordinate) impact the performance of scene flow. Not surprisingly, 3D object coordinate cues can improve the scene estimation with high margin among other cues. As we mentioned above, ISF is a two-stage method. Given consecutive stereo images, firstly, it predicts the scene 3-dimensional coordinate and instance segmentation with two separate branches, DispNet [20] and MNC [60]. Then it assigns each instance with 3D coordinate. The second stage is an energy minimization problem to infer the 3D geometry motion of each instance.

In summary, ISF provides promising results under certain conditions. However, there is a fly in the ointment, the inference time of ISF is heavy time-consuming due to its iterative solution. Yet, it provides new ideas of scene estimation for many subsequent methods.

DRISF [22] is similar to ISF, which is also an instance scene flow method. It employs 3 sub-networks, PWCNet [30], Mask-RCNN [61] and PSMNet [37] to estimate initial optical flow, instance segmentation and disparity respectively. Then, it uses these visual cues to form an energy function which is solved by a Gaussian-Newton (GN) solver to fit the best 3D geometry motion for each instance. In practical, DRISF unrolls the inference step of GN as a recurrent neural network which means the whole scheme can be treated as an end-to-end training method. In fact, all the three sub-networks can be learned through backpropagation. The results show that DRSIF has robust scene flow estimation on vary illumination and occlusion due to its ability to separate the cars from the scene. However, there are still two limitations. First, the performance of DRISF heavily depends on the quality of the instance segmentation. If the segmentation module does not recognize the car, the car is likely to be misclassified as a background scene. Meanwhile, the weights in the energy function prefer to the optical flow term. If the optical flow is not robust enough, the convergence of GN solver could be very slow.

PWOC-3D [62] discusses how to tackle the occlusion problem in scene flow estimation. Instead of using bidirectional flow for forward-backward consistency check to estimates occlusion, PWOC-3D is the first method that estimates occlusion with only forward flow. As the name suggesting, PWOC-3D inherits from PWCNet [30] by modifying the naive feature pyramid module to a FPN [63] liked module. More concretely, given four consecutive stereo images $(I^t_L, I^{t+1}_L, I^t_R, I^{t+1}_R)$ serve as the inputs, the network output the initial feature maps. At each pyramid level, the pyramid module produces four corresponding feature maps. Then it employs the warping strategy to warp each of these feature maps toward I^t_L via the predicted scene flow. With the obtained warped feature maps which are served as input to a sub-network, it predicts the occlusion probability mask map with sigmoid operation. The mask map ${}_l o^t_R : \Omega \rightarrow [0, 1]$ indicates the probability of whether the region is occluded. Then it performs pixel-wise multiplication between occlusion and its corresponding warped feature map for a masked feature map. After that, cost volume is constructed based on these masked feature maps. This whole occlusion estimation process is treated as a self-supervised training process. The intuition is that there will be similar characteristic between feature maps ${}_l c^t_L$ and ${}_l c^t_R$ if the region is not occluded, which will enforce the network adaptively to estimate these occluded regions.

FlowNet3.0 [64] is a continuation of FlowNet2.0 [28]. The authors propose a series of variants of FlowNet2.0 for the occlusion problem. Since the nature of optical flow and disparity is almost the same, the proposed generic network is suitable for both optical flow and disparity. There are mainly four proposed networks. 1)FlowNet-CSSR, in general, it follows the principal part of FlowNet2.0 but without the small displacement sub-network. The authors suggest that it can achieve robust optical flow or disparity through stacking multiple sub-networks. Meanwhile, the original fusion module is altered to be a refinement module with additional residual connections for further refinement. 2)FlowNetC-Bi which is based on the FlowNetC [19], employs bidirectional optical flow to estimate occlusion. 3)Then the authors separate the bidirectional optical flow into two independent streams. At each stream, it performs mutual warping to other direction. More concretely, it warps the backward optical flow towards the first image using the forward optical flow. Then, it flips the sign of the warped flow to serve as a forward flow. So the network has the forward optical flow and the corresponding backward optical flow. 4)Finally, the authors combine a FlowNet-CSS and two DispNet-CSS to form a generic network to estimate scene flow. The outputs from the above three networks are fused together and serve as the input of an additional FlowNetS for the final scene flow. Though FlowNet3.0 achieves high accuracy estimation of scene flow, the cumbersome network hinder the application.

5.2. Unsupervised methods for scene flow estimation

As we mention in Section 4.2, SFMLearner tries to predict monocular depth and ego-motion using unlabeled data in static scene. Many successive works extend this method to predict scene flow. Including GeoNet [65], DF-Net [66] and CC [67].

GeoNet [65] extends the SFMLearner [58] to scene flow estimation. It contains a two-stage training process. The first stage is to obtain the rigid structure of the background static scene geometry while the second stage is to learn non-rigid motion based on the results from the first stage.

The first stage is based on SFMLearner for the single view depth maps and the 6DoF ego-motion. Through fusing depth maps and the camera motion together, it can achieve the rigid flow $f^{rigid}_{t\rightarrow s}$. After that, it obtains $I^{\widetilde{rigid}}_s$ by warping I^{rigid}_t toward I_t via rigid flow $f^{rigid}_{t\rightarrow s}$ which will give supervised signal to train the first stage network.

The second stage includes a residual network named ResFlowNet to learn full optical flow. Because the first stage only learns the rigid flow which did not contain any dynamic objects. Since the GeoNet already learned the rigid flow in the first stage, the second stage only needs to learn the residual non-rigid flow. In the training process, the ResFlowNet learns the residual non-rigid flow $f^{res}_{t\rightarrow s} = f^{full}_{t\rightarrow s} - f^{rigid}_{t\rightarrow s}$, where $f^{full}_{t\rightarrow s}$ denotes the full flow. With the high quality rigid flow from the first stage, the second stage can learn residual non-rigid flow with a good initialization. In fact, the authors also note that the ResFlowNet not only rectifies the error within the object motion but also can refine the unperfect estimation.

DF-Net [66] futher extends SFMLearner, it projects the 3D scene flow, which is produced from depth maps and camera motion, to synthesis 2D optical flow within the rigid regions. It also proposes a cross-task consistency loss to minimize the difference between the rigid flow and the full flow. Geometric consistency loss is an extra supervised training signal. The overall framework of DF-Net includes 3 sub-networks for depth maps, ego-motion and bidirectional optical flow, respectively. Through fusing the depth maps and camera motion, it can achieve forward and backward 3D scene flow which is produced by the ego-motion. And the rigid flow can be obtained by projecting the 3D scene from the 3-dimensional space to 2-dimensional space. During training, DF-Net adopts photometric and spatial smoothness costs as regularization terms. The authors consider the rigid flow should be equal to full flow under the condition that most of the scene is static. However, the predictions from these two branches may not be consistency which gives the extra supervised signal to train the whole network. Thus, the authors propose the cross-task consistency loss as an extra loss function term which is formulated as:

$$L_{cross} = \sum_{p \in V_{depth} \cap V_{flow}} \left\| F_{rigid}(p) - F_{flow}(p) \right\|_1 \tag{13}$$

Meanwhile, DF-Net also adopts bidirectional optical flow to identify the occlusion. The occlused regions are masked out during backpropagation for further robust scene flow estimation.

Competitive Collaboration(CC) [67] considers the scene flow estimation as a three players game, 2 players directly compete for the resource and the 3rd player acts as a moderator. The first player $R = (D, C)$ of the two competitors utilizes the depth and camera motion for static scene reconstruction. In CC, the depth D is obtained from DispResNet, a variant of DispNet [20] by replacing the convolutional blocks with residual blocks [27]. An eight layers DCNN produces the ego-motion C. The other competitor F employs the optical flow which obtains from PWCNet [30] or FlowNetC [19] for moving region reconstruction. These two competitors compete for the training data via a static scene and moving regions. The static scene and moving regions are both referenced from unlabeled sequence images. The goal of this competition is to partition the unlabeled data set into two parts to minimize their own loss functions ($loss_R$, $loss_F$). During the competition, a motion segmentation network M acts as moderator M to determine the threshold of the partition. Because the total loss function may not be optimal with regard to the current loss function, it needs R and F to train M collaboratively. In practical, it first fixes the parameters of M in order to train R and F. After it converged under the current conditions, it fixs the parameters of R and F to train M until it converged. The final result is obtained through multiple iterations. The key idea of CC is very similar to expectation-maximization(EM) [68] algorithm with the different that CC unrolls the process of solution into DCNN based method.

6. Datasets and experimental evaluation metrics

In this section, we discuss some most used datasets and evaluation metrics.

6.1. Datasets

Datasets play an important role in the era of deep learning, especially for supervised learning methods.

KITTI [25,69] is a real scene dataset which consists of two major scene flow datasets, KITTI 2012 [69] and KITTI 2015 [25]. KITTI 2012 consists of 194 training scenes and 195 test scenes while KITTI 2015 consists of 200 training scenes and 200 test scenes. Comparing to KITTI 2012, KITTI 2015 comprises dynamic scenes for which the ground truth has been established in a semi-automatic process. In each scene, it contains two consecutive stereo images resulting in 4 images. It provides disparity ground truth for both two consecutive frames with regard to left view and optical flow ground truth for the first frame of left view. All the data are captured from a real scene with LiDAR results in around 50% ground truth density.

MPI Sintel [26] is a synthetic dataset built on an open source graphics movie “Sintel”. It provides sequence scenes with optical flow ground truth. In total, the train set contains 1064 frames while the test set contains 564 frames. The Sintel dataset is very challenging because the scene contains large motion, specular reflections, motion blur, defocus blur and atmospheric effects. There are two versions of the Sintel dataset, the Sintel Final which contains motion blur and atmospheric effects, and the Sintel Clean which does not contain those motion blur and atmospheric effects.

Flying Chairs [19] is a large scale synthetic datasets with optical flow ground truth. It consists of 22,872 image pairs as a train set with optical ground truth. There is no test set in Flying Chairs since it does not provide a benchmark. The data are rendered of synthetic 3D chairs moving in front of random backgrounds from Flickr. The motion in Flying Chairs only contains two-dimension. Flying Chairs is the first large scale dataset for optical flow estimation.

Scene Flow Datasets [20] is an evolution of Flying Chairs, which consists of three sub-datasets, FlyingThings3D, Driving, and MonKaa. Scene Flow Datasets is a very large scale synthetic dataset with total up to TBs data with the ground truth of optical flow, disparity, segmentation and motion boundaries. Comparing to Flying Chairs, it has much complexity scene with three-dimensional motion.

NYU-Depth-v2 [70] is comprised of video sequences of indoor scenes with depth ground truth for monocular depth estimation. The image and depth ground truth are captured by RGB camera and Microsoft Kinect, respectively. It consists of 1449 densely labeled pairs of aligned RGB with corresponding depth maps (Table 1).

6.2. Evaluation metric

Since the similarity between optical flow and disparity, they both subject to the most commonly used evaluation metric, average end point error (AEPE) which known as the Euclidean distance. Given total pixels N, the corresponding AEPE of disparity is defined as:

$$AEPE_{disp} = \frac{1}{N} \sum |d_i - \hat{d}_i| \tag{14}$$

Where d_i denotes the predicted disparity of pixel i and the corresponding groundtruth is denoted as $\hat{d}_i$. And the AEPE of optical flow is defined as :

$$AEPE_{flow} = \frac{1}{N} \sum \sqrt{(u - \hat{u}_i)^2 + (v - \hat{v}_i)^2} \tag{15}$$

Where u and v denotes the horizontal and vertial displacement of optical flow. Based on AEPE, Fl-All and D1-All indictates the percentage of outliers of optical flow and disparity, respectively. A pixel is considered as outlier if the AEPE exceeds 3 pixels as well as 5% of its true value.

RMSE(root mean sqrt erro) is a metric to evalute the accuracy of monocular depth estimation. RMSE is denoted as:

$$RMSE_{depth} = \frac{1}{N} \sum \left\| d_i - \hat{d}_i \right\|_2 \tag{16}$$

Table 1
Widely used datasets.

Dataset	scene	trains	tests	flow	depth	density	resolutions
KITTI 2012 [69]	real	194	195	yes	yes	~ 50%	(375, 1242)
KITTI 2015 [25]	real	200	200	yes	yes	~ 50%	(375, 1242)
MPI Sintel [26]	sythetic	1064	564	yes	no	100%	(436, 1024)
Flying Chairs [19]	sythetic	22,872	-	yes	no	100%	(384, 512)
Scene Flow Datasets [20]	sythetic	> 39000	-	yes	yes	100%	(540, 960)
NYU-Depth-v2 [70]	real	1449	-	no	yes	~ 100%	(480, 640)

Table 2
Evaluation metric on benchmark datasets, including KITTI and MPI Sintel. We only calculate the weights of convolutional and fully-connected layers. Some methods did not provide enough information for us to calculate the parameters are marked with '-'. Supervised methods are marked with '†' while un- or self-supervised methods are marked with '‡'.

Methods	KITTI 2012		KITTI 2015		Sintel		NYUv2	KITTI RAW	Params
	Fl-All	D1-All	Fl-all	D1-all	Clean	Final	RMSE	RMSE	
EpicFlow [72]	7.88%	-	26.29%	-	4.12	6.29	-	-	-
†FlowNet2 [28]	4.82%	-	10.41%	-	4.16	5.74	-	-	108.90m
†SPyNet [29]	12.31%	-	35.07%	-	6.64	8.36	-	-	1.20m
†PWC-Net [30]	4.22%	-	9.60%	-	4.39	5.04	-	-	6.92m
†LiteFlowNet [33]	3.27%	-	10.24%	-	4.86	6.09	-	-	1.34m
†IRR-PWC [71]	3.21%	-	7.65%	-	3.84	4.58	-	-	4.60m
‡SelFlow [53]	3.32%	-	14.19%	-	3.75	4.26	-	-	6.92m
‡DDFlow [51]	4.57%	-	14.29%	-	6.18	7.4	-	-	6.92m
‡OAULOF [73]	4.20%	-	31.20%	-	7.95	9.15	-	-	14.61m
†DispNetC [20]	-	4.11%	-	4.34%	-	-	-	-	42.32m
†GC-Net [36]	-	1.77%	-	2.87%	-	-	-	-	3.26m
†PSMNet [37]	-	1.49%	-	2.32%	-	-	-	-	4.76m
†SegStereo [74]	-	1.68%	-	2.25%	-	-	-	-	60.93m
†CSPN [44]	-	1.19%	-	1.74%	-	-	-	-	185.68m/266.34m
†CRL [35]	-	-	-	2.67%	-	-	-	-	36.31m
†DORN [46]	-	-	-	-	-	-	0.51	-	78.13m
†MSDN [41]	-	-	-	-	-	-	0.91	6.31	4.92m
†DCNF [75]	-	-	-	-	-	-	0.76	6.52	47.92m
‡LRC [55]	-	-	-	-	-	-	-	5.93	36.31m
‡SfMLearner [58]	-	-	-	-	-	-	-	6.71	36.31m
†ISF [21]	-	-	4.46%	6.22%	-	-	-	-	-
†DRISF [22]	-	-	2.55%	4.73%	-	-	-	-	52.79m
†PWOC-3D [62]	-	-	12.96%	5.13%	-	-	-	-	-
‡DF-Net [66]	-	-	22.82%	-	-	-	-	5.51	-
‡Geo-Net [66]	-	-	-	-	-	-	5.86	-	-

Where d_i denotes the predicted depth of pixel i and the corresponding groundtruth is denoted as $\hat{d}_i$.

6.3. Quantitative and qualitative comparison

We compare the methods above on pulicly available benchmark datasets which include KITTI, Sintel and NYU-Depth-v2. As shown in Table 2, the methods are evaluated on the Fl-All (optical flow on KITTI), D1-all (disparity on KITTI), RMSE (monocular depth on KITTI or NYUv2) and AEPE (optical flow on Sintel). We also compute the parameters of those methods (some methods does not provide enough information for us to compute their parameters). It is difficult to declare which method is the winner compared to the rest as it depends on the specific application scenario, including optical flow or depth estimation methods. IRR-PWC [71] has high optical flow accuracy while LiteFlowNet [33] has the low memories demand. CSPN [44] is the most powerful method for depth estimation but with heavy memories requirement. PSMNet [37] is a descent option for depth estimation as it takes reasonable memories and provides promising disparity result. DRISF [22] can provides both high accuracy optical flow and depth with extra segmentation through multiple sub-models. It also indicates that with extra visual clues, the network can produces better results of each subtask. Also, in the task of optical flow, the performance gap between the KITTI and Sintel dataset indicates that real scene data is much more challenging than synthetic data. It demonstrates that there is still many room for improvement on real scene data.

7. Conclusion

Scene flow estimation is a challenging and important task in many computer vision tasks. Benefiting from the rapid development of deep learning, scene flow estimation has achieved considerable accomplishments. Diverse novelty approaches have been proposed and yielded promising results. As a survey on deep learning methods for scene flow estimation, we highlight some of the most achievements in the past few years.

Although DCNN has shown tremendous improvement in scene flow estimation, there still remain several opportunities. We recognize some of those and summary as follow.

Incorporating other visual cues. Recently, many methods [18,76,77] realize that it can achieve robust scene flow estimation by incorporating other visual cues, like semantic segmentation, instance segmentation, object boundaries. The more important thing is that how to perfectly incorporate these visual cues. Many novelty methods have shown the potential of different incorporation approach. It is very challenging yet meaningful to design appropriate fusion approach.

Adaptive capacity in real scene. Since deep learning is a data-driven method, robust scene flow models rely on large amount real scene data which is still not accessible in current days. Various methods achieve a high rank among many benchmarks with significant low errors, but it does not necessarily mean that they can work well in the real scene. Recently, many methods try to tackle these problems via unsupervised, semi-supervised or self-supervised methods to relieve the burden of demanding for data. Though these methods yet not outperform supervised methods, it does shine some lights on the future direction.

A unified framework for scene flow estimation. Existing scene flow methods usually take separate models for initial optical flow, depth maps, and other cues. Then the separate cues are fused together for final scene flow estimation. As we all know, these approaches have a heavy redundant problem. Current scene flow methods usually take multiple separate models to predict depth and optical flow. Thus, a unified network which can produce robust both depth and optical flow can significantly reduce the computation and memories resource.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1305002 and the National Natural Science Foundation of China under Grant 61773414.

References

[1] S. Vedula, S. Baker, P. Rander, R. Collins, T. Kanade, Three-dimensional scene flow, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, IEEE, 1999, pp. 722–729.
[2] L. Chen, M. Cui, F. Zhang, B. Hu, K. Huang, High-speed scene flow on embedded commercial off-the-shelf systems, IEEE Trans. Ind. Inf. 15 (4) (2019) 1843–1852.
[3] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: European Conference on Computer Vision, Springer, 2004, pp. 25–36.
[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. (IJCV) 115 (3) (2015) 211–252, doi:10.1007/s11263-015-0816-y.
[5] D. Lai, W. Tian, L. Chen, Improving classification with semi-supervised and fine-grained learning, Pattern Recognit. 88 (2019) 547–556.
[6] S. Ren, K. He, R.B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[7] L. Chen, L. Fan, G. Xie, K. Huang, A. Nuchter, Moving-object detection from consecutive stereo pairs using slanted plane smoothing, IEEE Trans. Intell. Transp. Syst. 18 (11) (2017) 3093–3102.
[8] W. Wang, J. Shen, R. Yang, F. Porikli, Saliency-aware video object segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (1) (2018) 20–33.
[9] C. Vogel, K. Schindler, S. Roth, Piecewise rigid scene flow, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1377–1384.
[10] W. Wang, J. Shen, L. Shao, Video salient object detection via fully convolutional networks, IEEE Trans. Image Process. 27 (1) (2018) 38–49.
[11] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, M.-H. Yang, Deep regression tracking with shrinkage loss, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 353–369.
[12] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, F. Porikli, Hyperparameter optimization for tracking with continuous deep q-learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 518–527.
[13] L. Zhao, H. Bai, J. Liang, B. Zeng, A. Wang, Y. Zhao, Simultaneous color-depth super-resolution with conditional generative adversarial networks, Pattern Recognit. 88 (2019) 356–369.
[14] K. Nguyen, C. Fookes, S. Sridharan, M. Tistarelli, M. Nixon, Super-resolution for biometrics: a comprehensive survey, Pattern Recognit. 78 (2018) 23–42.
[15] F. Huguet, F. Devernay, A variational method for scene flow estimation from stereo sequences, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–7.
[16] Z. Tu, R. Poppe, R.C. Veltkamp, Weighted local intensity fusion method for variational optical flow estimation, Pattern Recognit. 50 (2016) 223–232.
[17] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, F. Porikli, See more, know more: unsupervised video object segmentation with co-attention siamese networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3623–3632.
[18] W. Wang, J. Shen, Deep visual attention prediction, IEEE Trans. Image Process. 27 (5) (2018) 2368–2378.
[19] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, FlowNet: Learning optical flow with convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2758–2766.
[20] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox, A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4040–4048.
[21] A. Behl, O. Hosseini Jafari, S. Karthik Mustikovela, H. Abu Alhaija, C. Rother, A. Geiger, Bounding boxes, segmentations and object coordinates: how important is recognition for 3d scene flow estimation in autonomous driving scenarios? in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2574–2583.
[22] W.-C. Ma, S. Wang, R. Hu, Y. Xiong, R. Urtasun, Deep rigid instance scene flow, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3614–3622.
[23] Z. Tu, N. Van Der Aa, C. Van Gemeren, R.C. Veltkamp, A combined post-filtering method to improve accuracy of variational optical flow estimation, Pattern Recognit. 47 (5) (2014) 1926–1940.
[24] L. Chen, Q. Wang, X. Lu, D. Cao, F.-Y. Wang, Learning driving models from parallel end-to-end driving data set, Proc. IEEE (2019).
[25] M. Menze, C. Heipke, A. Geiger, Joint 3d estimation of vehicles and scene flow., ISPRS Ann. Photogramm. Remote Sens.Spatial Inf. Sci. 2 (2015).
[26] D.J. Butler, J. Wulff, G.B. Stanley, M.J. Black, A naturalistic open source movie for optical flow evaluation, in: European Conf. on Computer Vision (ECCV), in: Part IV, LNCS 7577, Springer-Verlag, 2012, pp. 611–625.
[27] 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, 2016, pp. 770–778.
[28] 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. 2462–2470.
[29] A. Ranjan, M.J. Black, Optical flow estimation using a spatial pyramid network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4161–4170.
[30] D. Sun, X. Yang, M.-Y. Liu, J. Kautz, PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8934–8943.
[31] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
[32] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, arXiv:1511.07122(2015).
[33] T.-W. Hui, X. Tang, C. Change Loy, LiteFlowNet: a lightweight convolutional neural network for optical flow estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8981–8989.
[34] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[35] J. Pang, W. Sun, J.S. Ren, C. Yang, Q. Yan, Cascade residual learning: a two-stage convolutional neural network for stereo matching, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 887–895.
[36] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, A. Bry, End-to-end learning of geometry and context for deep stereo regression, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 66–75.
[37] J.-R. Chang, Y.-S. Chen, Pyramid stereo matching network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5410–5418.
[38] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[39] F. Zhang, V. Prisacariu, R. Yang, P.H. Torr, GA-Net: guided aggregation net for end-to-end stereo matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 185–194.
[40] H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2007) 328–341.
[41] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, in: Advances in Neural Information Processing Systems, 2014, pp. 2366–2374.
[42] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[43] Y. Gan, X. Xu, W. Sun, L. Lin, Monocular depth estimation with affinity, vertical pooling, and label enhancement, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 224–239.
[44] X. Cheng, P. Wang, R. Yang, Depth estimation via affinity learned with convolutional spatial propagation network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 103–119.
[45] S. Liu, S. De Mello, J. Gu, G. Zhong, M. Yang, J. Kautz, Learning affinity via spatial propagation networks, Neural Inf. Process. Syst. (2017) 1520–1530.
[46] H. Fu, M. Gong, C. Wang, K. Batmanghelich, D. Tao, Deep ordinal regression network for monocular depth estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2002–2011.
[47] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, H. Zha, Unsupervised deep learning for optical flow estimation, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[48] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, W. Xu, Occlusion aware unsupervised learning of optical flow, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4884–4893.
[49] S. Meister, J. Hur, S. Roth, UnFlow: unsupervised learning of optical flow with a bidirectional census loss, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[50] J. Janai, F. Guney, A. Ranjan, M. Black, A. Geiger, Unsupervised learning of multi-frame optical flow with occlusions, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 690–706.
[51] P. Liu, I. King, M.R. Lyu, J. Xu, DDFlow: learning optical flow with unlabeled data distillation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 8770–8777.
[52] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, K. He, Data distillation: towards omni-supervised learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4119–4128.
[53] P. Liu, M. Lyu, I. King, J. Xu, SelFlow: self-supervised learning of optical flow, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4571–4580.
[54] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.
[55] 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. 270–279.
[56] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, et al., Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[57] 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.
[58] T. Zhou, M. Brown, N. Snavely, D.G. Lowe, Unsupervised learning of depth and ego-motion from video (2017) 1851–1858.
[59] V. Casser, S. Pirk, R. Mahjourian, A. Angelova, Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 8001–8008.
[60] J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3150–3158.
[61] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[62] R. Saxena, R. Schuster, O. Wasenmuller, D. Stricker, PWOC-3D: deep occlusion-aware end-to-end scene flow estimation, in: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2019, pp. 324–331.
[63] T. Lin, P. Dollar, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature pyramid networks for object detection, Comput. Vis. Pattern Recognit. (2017) 936–944.
[64] E. Ilg, T. Saikia, M. Keuper, T. Brox, Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 614–630.
[65] 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.
[66] Y. Zou, Z. Luo, J.-B. Huang, DF-Net: unsupervised learning of depth and flow using cross-task consistency, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 36–53.
[67] 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12240–12249.
[68] K. Greff, S. van Steenkiste, J. Schmidhuber, Neural expectation maximization, in: Advances in Neural Information Processing Systems, 2017, pp. 6691–6701.
[69] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
[70] P.K. Nathan Silberman, Derek Hoiem, R. Fergus, Indoor segmentation and support inference from RGBD images, ECCV, 2012.
[71] J. Hur, S. Roth, Iterative residual refinement for joint optical flow and occlusion estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5754–5763.
[72] J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. Schmid, EpicFlow: edge-preserving interpolation of correspondences for optical flow, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1164–1172.
[73] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, W. Xu, Occlusion aware unsupervised learning of optical flow, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4884–4893.
[74] G. Yang, H. Zhao, J. Shi, Z. Deng, J. Jia, SegStereo: exploiting semantic information for disparity estimation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 636–651.
[75] F. Liu, C. Shen, G. Lin, I. Reid, Learning depth from single monocular images using deep convolutional neural fields, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2015) 2024–2039.
[76] J. Zhang, K.A. Skinner, R. Vasudevan, M. Johnson-Roberson, DispSegNet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery, IEEE Rob. Autom. Lett. 4 (2) (2019) 1162–1169.
[77] W. Wang, J. Shen, F. Porikli, R. Yang, Semi-supervised video object segmentation with super-trajectories, IEEE Trans. Pattern Anal. Mach. Intell. 41 (4) (2019) 985–998, doi:10.1109/TPAMI.2018.2819173.

Jiajie Liu received B.S. in Civil Engineering from Guangzhou University in 2017. He is a postgraduate student and studies machine learning and deep learning in applications of both pattern recognition and computer vision currently. His areas of interest include Scene Flow Estimation, Semantic Segmentation.

Han Li received a B.S. in Software Engineering from Shenzhen University in 2018. He is currently pursuing his M.S. degree in software engineering at the School of Data Science and Computer Science, Sun Yat-Sen University. He is interested in autonomous driving and image processing.

Ruihong Wu received a B.S in Software Engineering from Sun Yat-Sen University in 2019. She is currently pursuing her M.S degree in software engineering at the School of Data Science and Computer Science,Sun Yat-Sen University. She is interested in robotics and maching learning.

Qingyun Zhao received B.S in Information Management and Information Systems from Northeast Forestry University in 2019. She is pursuing her M.S degree in Sun Yat-Sen University. She is interested in Semantic Segmentation.

Yiyou Guo received B.S. in Geomatics Engineering and M.S. degrees in GIS from Wuhan University, China, in 2005 and 2010, respectively. He is currently pursuing his Ph.D. degree in the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His research interests include feature selection/learning, deep learning, image processing, and object tracking.

Long Chen received the B.Sc. degree in communication engineering and the Ph.D. degree in signal and information processing from Wuhan University, Wuhan, China, in 2007 and in 2013, respectively. From October 2010 to November 2012, he was co-trained PhD Student at National Univer-sity of Singapore. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He received the IEEE Vehicular Technology Society 2018 Best Land Transporta-tion Paper Award, IEEE Intelligent Vehicle Symposium 2018 Best Student Paper Award. His areas of interest include autonomous driving, robotics, articial intelligence where he has contributed more than 50 publications.