Pyramid Stereo Matching Network

Jia-Ren Chang Yong-Sheng Chen
Department of Computer Science, National Chiao Tung University, Taiwan
{followwar.cs00g, yschen}@nctu.edu.tw

Abstract

Recent work has shown that depth estimation from a stereo pair of images can be formulated as a supervised learning task to be resolved with convolutional neural networks (CNNs). However, current architectures rely on patch-based Siamese networks, lacking the means to exploit context information for finding correspondence in ill-posed regions. To tackle this problem, we propose PSMNet, a pyramid stereo matching network consisting of two main modules: spatial pyramid pooling and 3D CNN. The spatial pyramid pooling module takes advantage of the capacity of global context information by aggregating context in different scales and locations to form a cost volume. The 3D CNN learns to regularize cost volume using stacked multiple hourglass networks in conjunction with intermediate supervision. The proposed approach was evaluated on several benchmark datasets. Our method ranked first in the KITTI 2012 and 2015 leaderboards before March 18, 2018. The codes of PSMNet are available at: https://github.com/JiaRenChang/PSMNet.

1. Introduction

Depth estimation from stereo images is essential to computer vision applications, including autonomous driving for vehicles, 3D model reconstruction, and object detection and recognition [4, 31]. Given a pair of rectified stereo images, the goal of depth estimation is to compute the disparity d for each pixel in the reference image. Disparity refers to the horizontal displacement between a pair of corresponding pixels on the left and right images. For the pixel (x, y) in the left image, if its corresponding point is found at $(x - d, y)$ in the right image, then the depth of this pixel is calculated by $\frac{fB}{d}$, where f is the camera's focal length and B is the distance between two camera centers.

The typical pipeline for stereo matching involves the finding of corresponding points based on matching cost and post-processing. Recently, convolutional neural networks (CNNs) have been applied to learn how to match corresponding points in MC-CNN [30]. Early approaches using CNNs treated the problem of correspondence estimation as similarity computation [27, 30], where CNNs compute the similarity score for a pair of image patches to further determine whether they are matched. Although CNN yields significant gains compared to conventional approaches in terms of both accuracy and speed, it is still difficult to find accurate corresponding points in inherently ill-posed regions such as occlusion areas, repeated patterns, textureless regions, and reflective surfaces. Solely applying the intensity-consistency constraint between different viewpoints is generally insufficient for accurate correspondence estimation in such ill-posed regions, and is useless in textureless regions. Therefore, regional support from global context information must be incorporated into stereo matching.

One major problem with current CNN-based stereo matching methods is how to effectively exploit context information. Some studies attempt to incorporate semantic information to largely refine cost volumes or disparity maps [8, 13, 27]. The Displets [8] method utilizes object information by modeling 3D vehicles to resolve ambiguities in stereo matching. ResMatchNet [27] learns to measure reflective confidence for the disparity maps to improve performance in ill-posed regions. GC-Net [13] employs the encoder-decoder architecture to merge multiscale features for cost volume regularization.

In this work, we propose a novel pyramid stereo matching network (PSMNet) to exploit global context information in stereo matching. Spatial pyramid pooling (SPP) [9, 32] and dilated convolution [2, 29] are used to enlarge the receptive fields. In this way, PSMNet extends pixel-level features to region-level features with different scales of receptive fields; the resultant combined global and local feature clues are used to form the cost volume for reliable disparity estimation. Moreover, we design a stacked hourglass 3D CNN in conjunction with intermediate supervision to regularize the cost volume. The stacked hourglass 3D CNN repeatedly processes the cost volume in a top-down/bottom-up manner to further improve the utilization of global context information.

Our main contributions are listed below:

- We propose an end-to-end learning framework for stereo matching without any post-processing.
- We introduce a pyramid pooling module for incorporating global context information into image features.
- We present a stacked hourglass 3D CNN to extend the regional support of context information in cost volume.
- We achieve state-of-the-art accuracy on the KITTI dataset.

2. Related Work

For depth estimation from stereo images, many methods for matching cost computation and cost volume optimization have been proposed in the literature. According to [25], a typical stereo matching algorithm consists of four steps: matching cost computation, cost aggregation, optimization, and disparity refinement.

Current state-of-the-art studies focus on how to accurately compute the matching cost using CNNs and how to apply semi-global matching (SGM) [11] to refine the disparity map. Zbontar and LeCun [30] introduce a deep Siamese network to compute matching cost. Using a pair of 9×9 image patches, the network is trained to learn to predict the similarity between image patches. Their method also exploits typical stereo matching procedures, including cost aggregation, SGM, and other disparity map refinements to improve matching results. Further studies improve stereo depth estimation. Luo *et al*. [18] propose a notably faster Siamese network in which the computation of matching costs is treated as a multi-label classification. Shaked and Wolf [27] propose a highway network for matching cost computation and a global disparity network for the prediction of disparity confidence scores, which facilitate the further refinement of disparity maps.

Some studies focus on the post-processing of the disparity map. The Displets [8] method is proposed based on the fact that objects generally exhibit regular structures, and are not arbitrarily shaped. In the Displets [8] method, 3D models of vehicles are used to resolve matching ambiguities in reflective and textureless regions. Moreover, Gidaris and Komodakis [6] propose a network architecture which improves the labels by detecting incorrect labels, replacing incorrect labels with new ones, and refining the renewed labels (DRR). Gidaris and Komodakis [6] use the DRR network on disparity maps and achieve good performance without other post-processing. The SGM-Net [26] learns to predict SGM penalties instead of manually-tuned penalties for regularization.

Recently, end-to-end networks have been developed to predict whole disparity maps without post-processing. Mayer *et al*. [19] present end-to-end networks for the estimation of disparity (DispNet) and optical flow (FlowNet). They also offer a large synthetic dataset, Scene Flow, for network training. Pang *et al*. [21] extend DispNet [19] and introduce a two-stage network called cascade residual learning (CRL). The first and second stages calculate the disparity map and its multi-scale residuals, respectively. Then the outputs of both stages are summed to form the final disparity map. Also, Kendall *et al*. [13] introduce GC-Net, an end-to-end network for cost volume regularization using 3D convolutions. The above-mentioned end-to-end approaches exploit multiscale features for disparity estimation. Both DispNet [19] and CRL [21] reuse hierarchical information, concatenating features from lower layers with those from higher layers. CRL [21] also uses hierarchical supervision to calculate disparity in multiple resolutions. GC-Net [13] applies the encoder-decoder architecture to regularize the cost volume. The main idea of these methods is to incorporate context information to reduce mismatch in ambiguous regions and thus improve depth estimation.

In the field of semantic segmentation, aggregating context information is also essential for labeling object classes. There are two main approaches to exploiting global context information: the encoder-decoder architecture and pyramid pooling. The main idea of the encoder-decoder architecture is to integrate top-down and bottom-up information via skip connections. The fully convolutional network (FCN) [17] was first proposed to aggregate coarse-to-fine predictions to improve segmentation results. U-Net [24], instead of aggregating coarse-to-fine predictions, aggregates coarse-to-fine features and achieves good segmentation results for biomedical images. Further studies including SharpMask [22], RefineNet [15], and the label refinement network [12] follow this core idea and propose more complex architectures for the merging of multiscale features. Moreover, stacked multiple encoder-decoder networks such as [5] and [20] were introduced to improve feature fusion. In [20], the encoder-decoder architecture is termed the *hourglass* architecture.

Pyramid pooling was proposed based on the fact that the empirical receptive field is much smaller than the theoretical receptive field in deep networks [16]. ParseNet [16] demonstrates that global pooling with FCN enlarges the empirical receptive field to extract information at the whole-image level and thus improves semantic segmentation results. DeepLab v2 [2] proposes atrous spatial pyramid pooling (ASPP) for multiscale feature embedding, containing parallel dilated convolutions with different dilated rates. PSPNet [32] presents a pyramid pooling module to collect the effective multiscale contextual prior. Inspired by PSPNet [32], DeepLab v3 [3] proposes a new ASPP module augmented with global pooling.

Similar ideas of spatial pyramids have been used in context of optical flow. SPyNet [23] introduces image pyramids to estimate optical flow in a coarse-to-fine approach. PWC-

Net [28] improves optical flow estimation by using feature pyramids.

In this work on stereo matching, we embrace the experience of semantic segmentation studies and exploit global context information at the whole-image level. As described below, we propose multiscale context aggregation via a pyramid stereo matching network for depth estimation.

3. Pyramid Stereo Matching Network

We present PSMNet, which consists of an SPP [9, 32] module for effective incorporation of global context and a stacked hourglass module for cost volume regularization. The architecture of PSMNet is illustrated in Figure 1.

3.1. Network Architecture

The parameters of the proposed PSMNet are detailed in Table 1. In contrast to the application of large filters (7×7) for the first convolution layer in other studies [10], three small convolution filters (3×3) are cascaded to construct a deeper network with the same receptive field. The conv1_x, conv2_x, conv3_x, and conv4_x are the basic residual blocks [10] for learning the unary feature extraction. For conv3_x and conv4_x, dilated convolution is applied to further enlarge the receptive field. The output feature map size is $\frac{1}{4} \times \frac{1}{4}$ of the input image size, as shown in Table 1. The SPP module, as shown in Figure 1, is then applied to gather context information. We concatenate the left and right feature maps into a cost volume, which is fed into a 3D CNN for regularization. Finally, regression is applied to calculate the output disparity map. The SPP module, cost volume, 3D CNN, and disparity regression are described in later sections.

3.2. Spatial Pyramid Pooling Module

It is difficult to determine the context relationship solely from pixel intensities. Therefore, image features rich with object context information can benefit correspondence estimation, particularly for ill-posed regions. In this work, the relationship between an object (for example, a car) and its sub-regions (windows, tires, hoods, etc.) is learned by the SPP module to incorporate hierarchical context information.

In [9], SPP was designed to remove the fixed-size constraint of CNN. Feature maps at different levels generated by SPP are flattened and fed into the fully connected layer for classification, after which SPP is applied to semantic segmentation problems. ParseNet [16] applies global average pooling to incorporate global context information. PSPNet [32] extends ParseNet [16] to a hierarchical global prior, containing information with different scales and sub-regions. In [32], the SPP module uses adaptive average pooling to compress features into four scales and is followed by a 1×1 convolution to reduce feature dimension,

Table 1. Parameters of the proposed PSMNet architecture. Construction of residual blocks are designated in brackets with the number of stacked blocks. Downsampling is performed by conv0_1 and conv2_1 with stride of 2. The usage of batch normalization and ReLU follows ResNet [10], with exception that PSMNet does not apply ReLU after summation. H and W denote the height and width of the input image, respectively, and D denotes the maximum disparity.

Name	Layer setting	Output dimension
input		$H \times W \times 3$
CNN		
conv0_1	$3 \times 3, 32$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv0_2	$3 \times 3, 32$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv0_3	$3 \times 3, 32$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv1_x	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv2_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 16$	$\frac{1}{4}H \times \frac{1}{4}W \times 64$
conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$, dila = 2	$\frac{1}{4}H \times \frac{1}{4}W \times 128$
conv4_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$, dila= 4	$\frac{1}{4}H \times \frac{1}{4}W \times 128$
SPP module		
branch_1	64×64 avg. pool $3 \times 3, 32$ bilinear interpolation	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
branch_2	32×32 avg. pool $3 \times 3, 32$ bilinear interpolation	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
branch_3	16×16 avg. pool $3 \times 3, 32$ bilinear interpolation	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
branch_4	8×8 avg. pool $3 \times 3, 32$ bilinear interpolation	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
concat[conv2_16, conv4_3, branch_1, branch_2, branch_3, branch_4]		$\frac{1}{4}H \times \frac{1}{4}W \times 320$
fusion	$3 \times 3, 128$ $1 \times 1, 32$	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
Cost volume		
Concat left and shifted right		$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 64$
3D CNN (stacked hourglass)		
3Dconv0	$3 \times 3 \times 3, 32$ $3 \times 3 \times 3, 32$	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
3Dconv1	$\begin{bmatrix} 3 \times 3 \times 3, 32 \\ 3 \times 3 \times 3, 32 \end{bmatrix}$	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
3Dstack1_1	$3 \times 3 \times 3, 64$ $3 \times 3 \times 3, 64$	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dstack1_2	$3 \times 3 \times 3, 64$ $3 \times 3 \times 3, 64$	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 64$
3Dstack1_3	deconv $3 \times 3 \times 3, 64$ add **3Dstack1_1**	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dstack1_4	deconv $3 \times 3 \times 3, 32$ add **3Dconv1**	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
3Dstack2_1	$3 \times 3 \times 3, 64$ $3 \times 3 \times 3, 64$ add **3Dstack1_3**	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dstack2_2	$3 \times 3 \times 3, 64$ $3 \times 3 \times 3, 64$	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 64$
3Dstack2_3	deconv $3 \times 3 \times 3, 64$ add **3Dstack1_1**	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dstack2_4	deconv $3 \times 3 \times 3, 32$ add **3Dconv1**	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
3Dstack3_1	$3 \times 3 \times 3, 64$ $3 \times 3 \times 3, 64$ add **3Dstack2_3**	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dstack3_2	$3 \times 3 \times 3, 64$ $3 \times 3 \times 3, 64$	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 64$
3Dstack3_3	deconv $3 \times 3 \times 3, 64$ add **3Dstack1_1**	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dstack3_4	deconv $3 \times 3 \times 3, 32$ add **3Dconv1**	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
output_1	$3 \times 3 \times 3, 32$ $3 \times 3 \times 3, 1$	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 1$
output_2	$3 \times 3 \times 3, 32$ $3 \times 3 \times 3, 1$ add **output_1**	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 1$
output_3	$3 \times 3 \times 3, 32$ $3 \times 3 \times 3, 1$ add **output_2**	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 1$
3 output [output_1, outpu_t2, output_3]		
upsampling	Bilinear interpolation	$D \times H \times W$
	Disparity Regression	$H \times W$

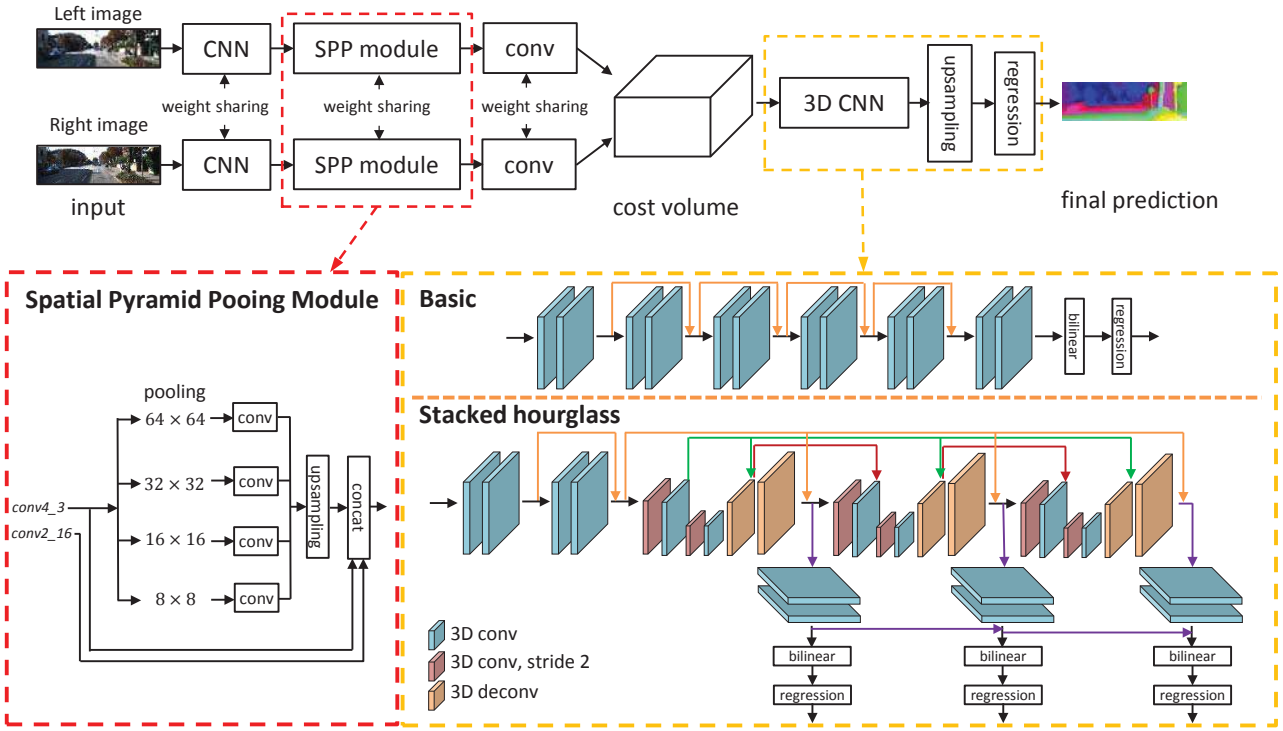


Figure 1. Architecture overview of proposed PSMNet. The left and right input stereo images are fed to two weight-sharing pipelines consisting of a CNN for feature maps calculation, an SPP module for feature harvesting by concatenating representations from sub-regions with different sizes, and a convolution layer for feature fusion. The left and right image features are then used to form a 4D cost volume, which is fed into a 3D CNN for cost volume regularization and disparity regression.

after which the low-dimensional feature maps are upsampled to the same size of the original feature map via bilinear interpolation. The different levels of feature maps are concatenated as the final SPP feature maps.

In the current work, we design four fixed-size average pooling blocks for SPP: 64×64, 32×32, 16×16, and 8×8, as shown in Figure 1 and Table 1. Further operations, including 1×1 convolution and upsampling, are the same as in [32]. In an ablation study, we performed extensive experiments to show the effect of feature maps at different levels, as described in Section 4.2.

3.3. Cost Volume

Rather than using a distance metric, the MC-CNN [30] and GC-Net [13] approaches concatenate the left and right features to learn matching cost estimation using deep network. Following [13], we adopt SPP features to form a cost volume by concatenating left feature maps with their corresponding right feature maps across each disparity level, resulting in a 4D volume (height×width×disparity×feature size).

3.4. 3D CNN

The SPP module facilitates stereo matching by involving different levels of features. To aggregate the feature information along the disparity dimension as well as spatial dimensions, we propose two kinds of 3D CNN architectures for cost volume regularization: the basic and stacked hourglass architectures. In the basic architecture, as shown in Figure1, the network is simply built using residual blocks. The basic architecture contains twelve $3 \times 3 \times 3$ convolutional layers. Then we upsample the cost volume back to size $H \times W \times D$ via bilinear interpolation. Finally, we apply regression to calculate the disparity map with size $H \times W$, which is introduced in Section 3.5.

In order to learn more context information, we use a stacked hourglass (encoder-decoder) architecture, consisting of repeated top-down/bottom-up processing in conjunction with intermediate supervision, as shown in Figure 1. The stacked hourglass architecture has three main hourglass networks, each of which generates a disparity map. That is, the stacked hourglass architecture has three outputs and losses (Loss_1, Loss_2, and Loss_3). The loss function is described in Section 3.6. During the training phase, the total loss is calculated as the weighted summation of the three

losses. During the testing phase, the final disparity map is the last of three outputs. In our ablation study, the basic architecture was used to evaluate the performance of the SPP module, because the basic architecture does not learn extra context information through the encoding/decoding process as in [13].

3.5. Disparity Regression

We use disparity regression as proposed in [13] to estimate the continuous disparity map. The probability of each disparity d is calculated from the predicted cost c_d via the softmax operation $\sigma(\cdot)$. The predicted disparity $\hat{d}$ is calculated as the sum of each disparity d weighted by its probability, as

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times \sigma(-c_d). \tag{1}$$

As reported in [13], the above disparity regression is more robust than classification-based stereo matching methods. Note that the above equation is similar to that introduced in [1], in which it is referred to as a soft attention mechanism.

3.6. Loss

Because of the disparity regression, we adopt the smooth L_1 loss function to train the proposed PSMNet. Smooth L_1 loss is widely used in bounding box regression for object detection because of its robustness and low sensitivity to outliers [7], as compared to L_2 loss. The loss function of PSMNet is defined as

$$L(d, \hat{d}) = \frac{1}{N} \sum_{i=1}^{N} smooth_{L_1}(d_i - \hat{d}_i), \tag{2}$$

in which

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

where N is the number of labeled pixels, d is the ground-truth disparity, and $\hat{d}$ is the predicted disparity.

4. Experiments

We evaluated our method on three stereo datasets: Scene Flow, KITTI 2012, and KITTI 2015. We also performed ablation studies using KITTI 2015 with our architecture setting to evaluate the influence on performance made by dilated convolution, different sizes of pyramid pooling, and the stacked hourglass 3D CNN. The experimental settings and network implementation are presented in Section 4.1, followed by the evaluation results on each of the three stereo datasets used in this study.

4.1. Experiment Details

We evaluated our method on three stereo datasets:

1. Scene Flow: a large scale synthetic dataset containing 35454 training and 4370 testing images with $H = 540$ and $W = 960$. This dataset provides dense and elaborate disparity maps as ground truth. Some pixels have large disparities and are excluded in the loss computation if the disparity is larger than the limits set in our experiment.

2. KITTI 2015: a real-world dataset with street views from a driving car. It contains 200 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and another 200 testing image pairs without ground-truth disparities. Image size is $H = 376$ and $W = 1240$. We further divided the whole training data into a training set (80%) and a validation set (20%).

3. KITTI 2012: a real-world dataset with street views from a driving car. It contains 194 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and 195 testing image pairs without ground-truth disparities. Image size is $H = 376$ and $W = 1240$. We further divided the whole training data into a training set (160 image pairs) and a validation set (34 image pairs). Color images of KITTI 2012 were adopted in this work.

The full architecture of the proposed PSMNet is shown in Table 1, including the number of convolutional filters. The usage of batch normalization and ReLU is the same as in ResNet [10], with exception that PSMNet does not apply ReLU after summation.

The PSMNet architecture was implemented using PyTorch. All models were end-to-end trained with Adam ($\beta_1 = 0.9, \beta_2 = 0.999$). We performed color normalization on the entire dataset for data preprocessing. During training, images were randomly cropped to size $H = 256$ and $W = 512$. The maximum disparity (D) was set to 192. We trained our models from scratch using the Scene Flow dataset with a constant learning rate of 0.001 for 10 epochs. For Scene Flow, the trained model was directly used for testing. For KITTI, we used the model trained with Scene Flow data after fine-tuning on the KITTI training set for 300 epochs. The learning rate of this fine-tuning began at 0.001 for the first 200 epochs and 0.0001 for the remaining 100 epochs. The batch size was set to 12 for the training on four nNvidia Titan-Xp GPUs (each of 3). The training process took about 13 hours for Scene Flow dataset and 5 hours for KITTI datasets. Moreover, we prolonged the training process to 1000 epochs to obtain the final model and the test results for KITTI submission.

Table 2. Evaluation of PSMNet with different settings. We computed the percentage of three-pixel-error on the KITTI 2015 validation set, and end-point-error on the Scene Flow test set. * denote that we use half the dilated rate of dilated convolution.

Network setting						KITTI 2015	Scene Flow
dilated conv	pyramid pooling size				stacked hourglass	Val Err (%)	End Point Err
	64×64	32×32	16×16	8×8			
						2.43	1.43
√						2.16	1.56
	√	√	√	√		2.47	1.40
√	√					2.17	1.30
√	√	√	√	√		2.09	1.28
√	√	√	√	√	√	1.98	**1.09**
√*	√	√	√	√	√	**1.83**	1.12

Table 3. Influence of weight values for Loss_1, Loss_2, and Loss_3 on validation errors. We empirically found that 0.5/0.7/1.0 yielded the best performance.

Loss weight			KITTI 2015
Loss_1	Loss_2	Loss_3	val error(%)
0.0	0.0	1.0	2.49
0.1	0.3	1.0	2.07
0.3	0.5	1.0	2.05
0.5	0.7	1.0	**1.98**
0.7	0.9	1.0	2.05
1.0	1.0	1.0	2.01

4.2. KITTI 2015

Ablation study for PSMNet We conducted experiments with several settings to evaluate PSMNet, including the usage of dilated convolution, pooling at different levels, and 3D CNN architectures. The default 3D CNN design was the basic architecture. As listed in Table 2, dilated convolution works better when used in conjunction with the SPP module. For pyramid pooling, pooling with more levels works better. The stacked hourglass 3D CNN significantly outperformed the basic 3D CNN when combined with dilated convolution and the SPP module. The best setting of PSMNet yielded a 1.83% error rate on the KITTI 2015 validation set.

Ablation study for Loss Weight The stacked hourglass 3D CNN has three outputs for training and can facilitate the learning process. As shown in Table 3, we conducted experiments with various combinations of loss weights between 0 and 1. For the baseline, we treated the three losses equally and set all to 1. The results showed that the weight settings of 0.5 for Loss_1, 0.7 for Loss_2, and 1.0 for Loss_3 yielded the best performance, which was a 1.98% error rate on the KITTI 2015 validation set.

Results on Leaderboard Using the best model trained in our experiments, we calculated the disparity maps for the 200 testing images in the KITTI 2015 dataset and submitted the results to the KITTI evaluation server for the performance evaluation. According to the online leaderboard, as shown in Table 4, the overall three-pixel-error for the proposed PSMNet was **2.32**%, which surpassed prior studies by a noteworthy margin. In this table, "All" means that all pixels were considered in error estimation, whereas "Noc" means that only the pixels in non-occluded regions were taken into account. The three columns "D1-bg", "D1-fg" and "D1-all" mean that the pixels in the background, foreground, and all areas, respectively, were considered in the estimation of errors.

Qualitative evaluation Figure 2 illustrates some examples of the disparity maps estimated by the proposed PSMNet, GC-Net [13], and MC-CNN [30] together with the corresponding error maps. These results were reported by the KITTI evaluation server. PSMNet yields more robust results, particularly in ill-posed regions, as indicated by the yellow arrows in Figure 2. Among these three methods, PSMNet more correctly predicts the disparities for the fence region, indicated by the yellow arrows in the middle row of Figure 2.

4.3. Scene Flow

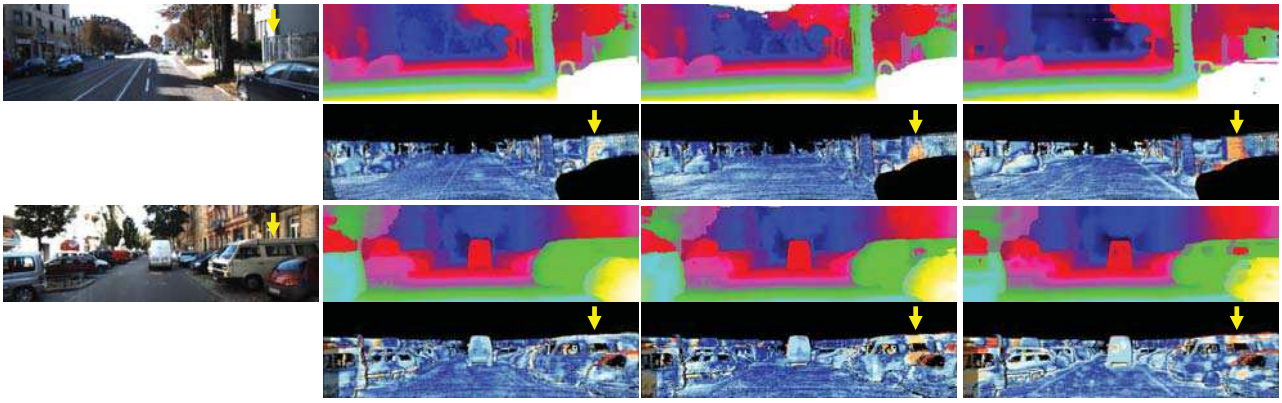
We also compared the performance of PSMNet with other state-of-the-art methods, including CRL [21], DispNetC [19], GC-Net [13], using the Scene Flow test set. As shown in Table 5, PSMNet outperformed other methods in terms of accuracy. Three testing examples are illustrated in Figure 3 to demonstrate that PSMNet obtains accurate disparity maps for delicate and intricately overlapped objects.

4.4. KITTI 2012

Using the best model trained in our experiments, we calculated the disparity maps for the 195 testing images in the

Table 4. The KITTI 2015 leaderboard presented on March 18, 2018. The results show the percentage of pixels with errors of more than three pixels or 5% of disparity error from all test images. Only published methods are listed for comparison.

Rank	Method	All (%)			Noc (%)			Runtime (s)
		D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
1	PSMNet (ours)	**1.86**	4.62	**2.32**	**1.71**	4.31	**2.14**	0.41
3	iResNet-i2e2 [14]	2.14	3.45	2.36	1.94	3.20	2.15	0.22
6	iResNet [14]	2.35	**3.23**	2.50	2.15	**2.55**	2.22	**0.12**
8	CRL [21]	2.48	3.59	2.67	2.32	3.12	2.45	0.47
11	GC-Net [13]	2.21	6.16	2.87	2.02	5.58	2.61	0.90



(a) PSMNet (b) GC-Net (c) MC-CNN

Figure 2. Results of disparity estimation for KITTI 2015 test images. The left panel shows the left input image of stereo image pair. For each input image, the disparity maps obtained by (a) PSMNet, (b) GC-Net [13], and (c) MC-CNN [30] are illustrated together above their error maps.

Table 5. Performance comparison with Scene Flow test set. EPE: End-point-error.

	PSMNet	CRL [21]	DispNetC [19]	GC-Net [13]
EPE	**1.09**	1.32	1.68	2.51

KITTI 2012 dataset and submitted the results to the KITTI evaluation server for the performance evaluation. According to the online leaderboard, as shown in Table 6, the overall three-pixel-error for the proposed PSMNet was **1.89**%, which surpassed prior studies by a noteworthy margin.

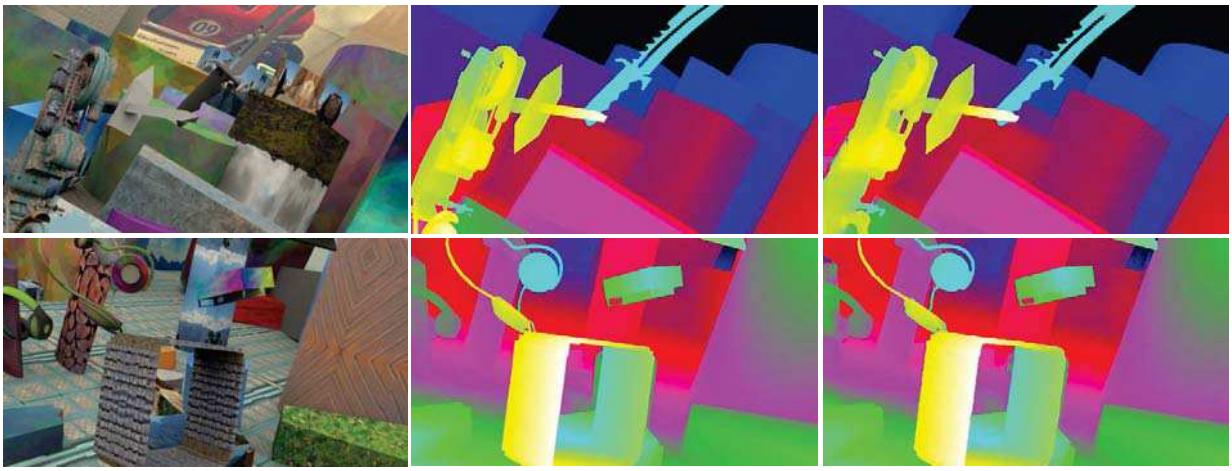
Qualitative evaluation Figure 4 illustrates some examples of the disparity maps estimated by the proposed PSMNet, GC-Net [13], and MC-CNN [30] together with the corresponding error maps. These results were reported by the KITTI evaluation server. PSMNet obtains more robust results, particularly in regions of car windows and walls, as indicated by the yellow arrows in Figure 4.

5. Conclusions

Recent studies using CNNs for stereo matching have achieved prominent performance. Nevertheless, it remains intractable to estimate disparity for inherently ill-posed regions. In this work, we propose PSMNet, a novel end-to-end CNN architecture for stereo vision which consists of two main modules to exploit context information: the SPP module and the 3D CNN. The SPP module incorporates different levels of feature maps to form a cost volume. The 3D CNN further learns to regularize the cost volume via repeated top-down/bottom-up processes. In our experiments, PSMNet outperforms other state-of-the-art methods. PSMNet ranked first in both KITTI 2012 and 2015 leaderboards before March 18, 2018. The estimated disparity maps clearly demonstrate that PSMNet significantly reduces errors in ill-posed regions.

Acknowledgement

This work was supported in part by the Taiwan Ministry of Science and Technology (Grants MOST-106-2221-E-009-164-MY2 and MOST-107-2634-F-009-009).

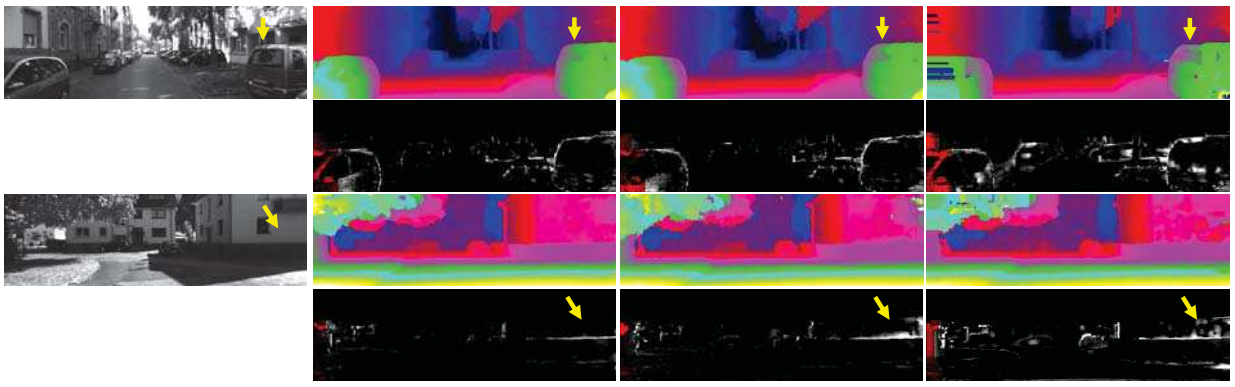


(a) left image (b) ground truth (c) PSMNet output

Figure 3. Performance evaluation using Scene Flow test data. (a) left image of stereo image pair, (b) ground truth disparity, and (c) disparity map estimated using PSMNet.

Table 6. The leaderboard of KITTI 2012 presented on March 18, 2018. PSMNet achieves the best results under all evaluation criteria, except runtime. Only published methods are listed for comparison.

Rank	Method	>2 px		>3 px		>5 px		Mean Error		Runtime (s)
		Noc	All	Noc	All	Noc	All	Noc	All	
1	PSMNet (ours)	**2.44**	**3.01**	**1.49**	**1.89**	**0.90**	**1.15**	**0.5**	**0.6**	0.41
2	iResNet-i2 [14]	2.69	3.34	1.71	2.16	1.06	1.32	0.5	0.6	**0.12**
4	GC-Net [13]	2.71	3.46	1.77	2.30	1.12	1.46	0.6	0.7	0.9
11	L-ResMatch [27]	3.64	5.06	2.27	3.40	1.50	2.26	0.7	1.0	48
14	SGM-Net [26]	3.60	5.15	2.29	3.50	1.60	2.36	0.7	0.9	67



(a) PSMNet (b) GC-Net (c) MC-CNN

Figure 4. Results of disparity estimation for KITTI 2012 test images. The left panel shows the left input image of the stereo image pair. For each input image, the disparity obtained by (a) PSMNet, (b) GC-Net [13], and (c) MC-CNN [30], is illustrated above its error map.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 5

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully con-

nected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 1, 2

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[4] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3D object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015. 1

[5] J. Fu, J. Liu, Y. Wang, and H. Lu. Stacked deconvolutional network for semantic segmentation. *arXiv preprint arXiv:1708.04943*, 2017. 2

[6] S. Gidaris and N. Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[7] R. Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5

[8] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4175, 2015. 1, 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014. 1, 3

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 5

[11] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005. 2

[12] M. A. Islam, S. Naha, M. Rochan, N. Bruce, and Y. Wang. Label refinement network for coarse-to-fine semantic segmentation. *arXiv preprint arXiv:1703.00551*, 2017. 2

[13] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2, 4, 5, 6, 7, 8

[14] Z. Liang, Y. Feng, Y. Guo, H. Liu, L. Qiao, W. Chen, L. Zhou, and J. Zhang. Learning deep correspondence through prior and posterior feature constancy. *arXiv preprint arXiv:1712.01039*, 2017. 7, 8

[15] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[16] W. Liu, A. Rabinovich, and A. C. Berg. ParseNet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 2, 3

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2

[18] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016. 2

[19] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 6, 7

[20] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 2

[21] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 6, 7

[22] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 2

[23] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 2

[24] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 2

[25] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 2

[26] A. Seki and M. Pollefeys. SGM-Nets: Semi-global matching with neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 8

[27] A. Shaked and L. Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 8

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *arXiv preprint arXiv:1709.02371*, 2017. 3

[29] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016. 1

[30] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 1, 2, 4, 6, 7, 8

[31] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui. Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2057–2065, 2015. 1

[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 4