Stereo-Matching Network for Structured Light

Qiuchen Du, Rongke Liu, Boshen Guan, Yu Pan, and Shuqiao Sun

***Abstract*—Recently, deep learning has been widely applied in binocular stereo matching for depth acquisition, which has led to an immense increase of accuracy. However, little attention has been paid to the structured light field. In this letter, a network for structured light is proposed to extract effective matching features for depth acquisition. The proposed network promotes the Siamese network by considering receptive fields of different scales and assigning proper weights to the corresponding features, which is achieved by combining pyramid-pooling structure with the squeeze-and-excitation network into the Siamese network for feature extraction and weight calculations, respectively. For network training and testing, a structured-light dataset with amended ground truths is generated by projecting a random pattern into the existing binocular stereo dataset. Experiments demonstrate that the proposed network is capable of real-time depth acquisition, and it provides superior depth maps using structured light.**

***Index Terms*—Structured light, stereo matching, siamese network, SLNet.**

I. Introduction

STRUCTURED light has been widely used in depth acquisition due to its robustness to low-texture scenes and multipath interference, and its capability to produce dense and high-precision depth maps. Currently, several depth cameras based on structured light have been developed, such as Kinect, which realizes real-time depth acquisition, but it has low accuracy and resolution that restrict its widespread use in industrial and scientific contexts. Meanwhile, Intel RealSense [1] provides larger depth maps at a high frame rate, but its accuracy is not satisfactory.

Considerable effort has been made towards improving structured-light-based depth-acquisition algorithms. Classic approaches use designed patterns with distinguishable features [2]–[4], and some consider sub-pixels [5] to improve the accuracy. Moreover, while the speed and accuracy are boosted through machine-learning methods such as random forest [6] and unsupervised learning [7], to date, deep features have remain unconsidered.

Manuscript received July 24, 2018; revised November 4, 2018; accepted November 21, 2018. Date of publication November 28, 2018; date of current version December 8, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61231010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kai Liu. *(Corresponding author: Rongke Liu.)*

The authors are with the Department of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: QC-Du@hotmail.com; rongke_liu@buaa.edu.cn; iceguan@msn.com; panyukop@gmail.com; shuqiao_sun@qq.com).

Digital Object Identifier 10.1109/LSP.2018.2883865

Lately, binocular stereo matching performs better in depth acquisition when used with deep-learning techniques. Since structured light is akin to binocular matching, it is natural to assume that similar improvement may arise. In structured-light systems, projected patterns and patterned scenes are equivalent to the two cameras in binocular matching. Although the network in binocular matching can be directly used in structured light, the results are less accurate due to the absence of structured-light features. Both local features [8]–[10] and global features [11]–[14] are accessible through neural networks in feature extraction. Because of their relatively low generalization ability, global features are only suitable for specific scenes. Considering that structured-light patterns have strong local features, the Siamese network is suitable for extracting such features. In this letter, a Network for Structured Light (SLNet) is proposed, which contains an efficient Siamese network [9], pyramid-pooling [10], [15] and an improved Squeeze-and-Excitation Network (SENet) [16]. The efficient Siamese network [9] treats image-patch matching as a multi-classification task, which results in higher accuracy than methods that use binary classification [8] [17]. Now that different receptive fields have different effects on feature matching, adding pyramid pooling [10] to the Siamese network can take receptive fields into consideration. Moreover, the degree-of-effect can be calculated quantitatively. As a result, the improved SENet [16] architecture is added to calculate the weights of each receptive field.

To train the network, a structured-light-based training dataset, which includes patterned scenes and their corresponding ground truths, is created according to an existing binocular stereo dataset. First, the 2D scenes in binocular dataset are reconstructed into 3D scenes using their calibration parameters. Next, a random pattern is projected onto the 3D scenes and captured by a camera. Finally, the ground truth is calculated based on the parameters and the baseline of the projector and the camera. This solves the problem of a lack of objective evaluation indices, such as PSNR, SSIM, and Bad2.0, in the depth maps acquired via the structured light.

The main contributions of this work are: 1) we propose SLNet, which is suitable for structured-light-based depth acquisition; 2) to create a structured-light dataset for training and testing; and 3) to realize real-time depth acquisition with relatively high quality.

The remainder of this letter is structured as follows: In Section II, we describe the architecture of the SLNet, while in Section III we introduce the training and testing of the network. Experimental results are presented in Section IV, and finally, Section V concludes this letter.

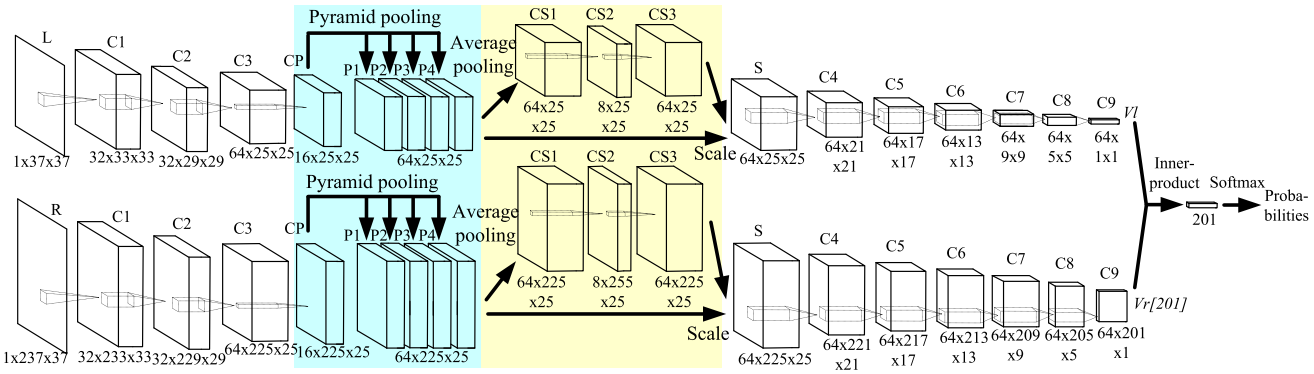


Fig. 1. The SLNet architecture, which takes two input image patches as input and outputs the matching probabilities. Pyramid pooling and SENet are respectively labelled in blue and yellow. The $c \times m \times n$ tags at the bottom of each layer represent the number of channels (c), lines (m), and columns (n) in each layer.

II. SLNet Architecture

The SLNet takes two image patches as input: one is square and extracted from the patterned scene, while the other is rectangular and contains all possible matching positions from the projected pattern. Output from the network includes the matching probabilities of all possible positions from which the ultimate matching position is selected. The architecture of the SLNet is shown in Fig. 1, which contains three parts: an efficient Siamese network, pyramid pooling and the improved SENet. Detailed descriptions of each are provided in the following subsections.

A. Efficient Siamese Network

The basic structure of the SLNet was designed according to the efficient Siamese network of [9]. Input to the SLNet contains two image patches of different sizes. The left patch, extracted from the patterned scene, is a square patch centred at the point to be matched, while the right patch, extracted from the projected pattern at the same row, is a rectangular patch including 201 possible matching points within a maximum disparity of 100. Features are calculated separately for the left and right patches despite them having the same structure and parameters. Each branch cascades nine convolution layers $C1 - C9$ with the same kernel size (5×5), stride (1), and no padding. The size of the convolution layers decreases as the layers increase in depth. At the final convolution layer, a vector, Vl, which represents a feature of the left patch, and a vector array $Vr[201]$, which represents the 201 features of the possible matching patches, are estimated. Similarities between Vl and $Vr[201]$ are measured by the following inner-product layer and are then passed to a Softmax layer to obtain the matching probabilities. If we suppose that the index of the patch with the maximum probability is i, the disparity of the centre point of the left patch is $|i - 101|$.

In practice, the entire left and right images, rather than the patches, are used as input to the SLNet for efficiency. If we consider that the projected pattern may remain unchanged in structured light, the output of the right branch can be obtained in advance, thus allowing us to remove the right branch. Multiple convolution layers are employed in this basic structure, while the features of the structured light is not involved at this stage.

B. Pyramid Pooling

The size of the receptive field has certain effects on the image matching. If it is too small, the matching results will be accurate but susceptible to noise. On the other hand, if the size is too large, the depth maps will be excessively smooth and lose details. As such, pyramid pooling, labelled with a blue background in Fig. 1, was added into the middle of the basic structure for extracting features with different receptive fields. The receptive field represents the size in the input layer (L or R) sensed by a pixel in the convolution layer. For example, the receptive fields in $C1$, $C2$ and $C3$ are, respectively, 5, 9, and 13. In order to extract any features with different receptive fields simultaneously, four max-pooling layers with increased kernel sizes were inserted, which are shown as $P1 - P4$. The stride of each pooling layer is 1 in the case of pixel information losses. The kernels of the four pooling layers, $P1 - P4$, are, respectively, 1, 3, 5, and 7, so the receptive fields of each layer are 13, 17, 21 and 25.

A convolution layer, CP, with the kernel size (1×1) was added to the front of the pyramid pooling to reduce the number of channels to 1/4. As such, the number of channels remained unchanged and the number of parameters in the SLNet slightly increased.

C. Squeeze-and-Excitation Network

The Squeeze-and-Excitation Network (SENet) [16] gives an explicit quantitative expression of the features' weights in each receptive field. The improved SENet structure is labelled with a yellow background in Fig. 1. The output of the pyramid pooling contains features from the different receptive fields. Each feature has a distinct effect on the results, so the SENet was added after the pyramid pooling to learn the weight of each feature and multiply each feature by the corresponding weight to scale it, which enhances useful features and suppresses fruitless ones.

The SENet of [16] used global pooling, namely the average of all the pixels, to acquire global features of each channel. Then, they employed a fully connected layer in an independent branch to calculate the weights. Since a global feature loses the precise positioning of specific features, when rectangular image patches are applied, the global pooling can be improved by using average pooling (e.g. a kernel size of 25×25, a stride

of 1 and padding of 12). The output of the average pooling is the same size as the input, so the position of features is hence retained. If we consider that the output of the average pooling is a matrix rather than a vector, the fully connected layer in SENet can be replaced by a convolution layer with a kernel size of 1×1. $CS1 - CS3$ are such convolution layers, which are used to calculate the weights, followed by a Sigmoid layer that converts these weights to (0,1).

III. SLNet Training and Testing

A structured-light dataset was created by considering that the SLNet needs a huge amount of labelled data to train. The training and testing procedures are similar to those of [9].

A. Structured-Light Dataset Creation

Due to a lack of off-the-shelf structured-light datasets to train the SLNet, a dataset was created based on existing binocular stereo datasets. The structured-light dataset creation mainly consisted of three steps: 3D scene reconstruction, scene obtainment with random patterns, and new ground truth calculations.

The 3D scene reconstruction was based on an off-the-shelf binocular datasets. KITTI [18] was not used due to its sparse ground truths. Middlebury [19] was used, but only as test dataset because of its insufficient number. Instead, Monkaa [20], which does have sufficient sample quantity and accurate ground truths, was adopted as the training dataset. One pixel in the 2D ground truth of the binocular dataset was mapped into a 3D camera coordinate according to the camera's internal parameters. The mapping formulae are as follows:

$$Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{cases} X = (x - c_x)\, Z/f_x \\ Y = (y - c_y)\, Z/f_y \\ Z = f_x T_x / d \end{cases} \tag{1}$$

where (x, y) are the image coordinates of the 2D ground truth, and (X, Y, Z) is the reconstructed point in the 3D camera coordinates. f_x and f_y are the focal lengths, and c_x and c_y are the principal points of the cameras, in x- and y- directions, respectively. Z is calculated according to stereo triangulation, where d and T_x are the ground truth and baseline between the cameras, respectively. All the reconstructed points constitute a 3D point cloud. By connecting three adjacent points to construct a plane, a high-precision 3D model can be generated.

To obtain a patterned scene, a random pattern was projected onto the surfaces in the 3D model. Random patterns, which have obvious and unique local features and strong fault-tolerance, are frequently used for mature depth cameras, and are hence the most appropriate for the SLNet. The colours of the 3D models have been removed, and some noise was added to simulate infrared scenes taken by Kinect-V1. To ensure that a patterned scene captured by camera was of the same size and position as the scene in the binocular stereo dataset, the horizontal viewing angle of the camera was set to $2 \times \arctan(w/2/f_x)$, and the aspect ratio was w/h, where w and h are the width and height of the scene in the binocular stereo dataset. To avoid rectification, internal parameters of the projector and the camera were set to the same values. The camera and the projector were respectively positioned at $(0, 0, 0)$ and $(20, 0, 0)$. Their image planes were both perpendicular to the z-axis. In practice, a camera and a projector with arbitrary parameters and positions are available. However, they must be rectified according to [21].

The ground truth of the structured-light dataset, ds, was deduced from the ground truth of the binocular dataset, d. According to $Z = f_x T_x / d$ in Eq. 1, since Z and f_x in the structured light are the same as those in the binocular dataset, the ground truth, d, is proportional to the baseline, T_x. Supposing that the baseline of the camera and the projector is T_s, the ground truth of the structured light is $ds = dT_s\ /T_x$.

B. Training

Training samples were generated from structured-light dataset. Left patches were extracted from the patterned scenes. Invalid patches, i.e., those with no patterns and shadowed areas, were excluded. Right patches were extracted from the projected patterns according to the ground truths. Supposing that the left patch is centred at (xl, yl) and the ground truth at (xl, yl) is d, the right patch is then centred at $(xl - d, yl)$. The number of training samples was about 20 million, which were extracted from 120 randomly selected scenes.

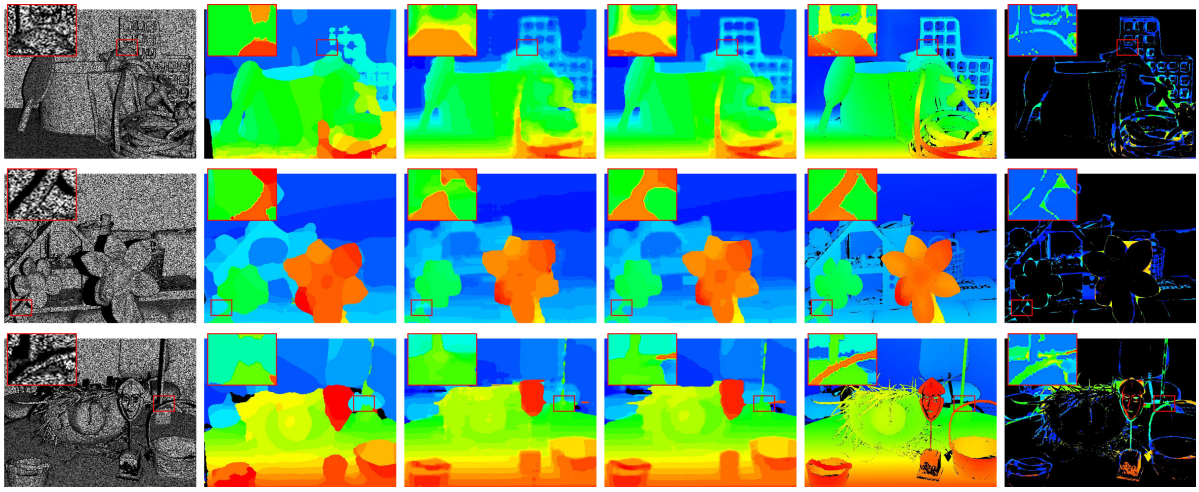
The SLNet was trained using stochastic gradient descent back propagation with AdaGrad [22]. The loss function of the SLNet is the same as that of [9]. A smooth target distribution centred at the ground truth was applied in the loss function. In the structured light, a discontinuous region can seriously affect the matching results; however, such regions can be identified and removed by considering nearby points. The SLNet was trained with the structured-light dataset based on Monkaa. The training times were 40 000, which were divided into 200 epochs. It took about 4 hours for the process to complete on a Geforce TITAN Xp.

C. Testing

In the testing, entire images rather than image patches were passed as input to the SLNet to improve the efficiency, since features in each pixel would be calculated only once. The output of each branch was a feature map of 64 channels. To ensure that the sizes of the feature maps and the input images are the same, a padding layer was added in front of each branch. To obtain feature maps at all possible matching positions within the maximum disparity, feature maps of the projected patterns were translated to the right pixel-by-pixel. The inner product of the left feature map and the feature maps at all possible matching positions were calculated, which allowed us to determine positions with the highest matching probabilities.

IV. Experimental Results

Experiments were conducted to validate the performance of the SLNet, including simulations and real-world experiments. The simulated experiments used reconstructed test scenes based on Middlebury for three image-matching methods: the classic ZNCC method, the network of [9] and the SLNet. Images from Middlebury were reduced to 1/4 resolution due to memory limitations. The SLNet was not fine-tuned on Middlebury to show



(a) (b) (c) (d) (e) (f)

Fig. 2. Simulated experiments of three Middlebury scenes in the structured-light dataset. The subfigure at the top left corner shows a zoomed-in image patch surrounded by a red rectangle. From the top to bottom: cables, flowers and mask. (a) Patterned scenes, (b) depth maps acquired by the classic ZNCC, (c) depth maps acquired by the binocular network of [9], (d) depth maps acquired by the SLNet (e) ground truths and (f) error maps of the SLNet.

TABLE I
BAD1.0 AND BAD2.0 OF THE DEPTH MAPS

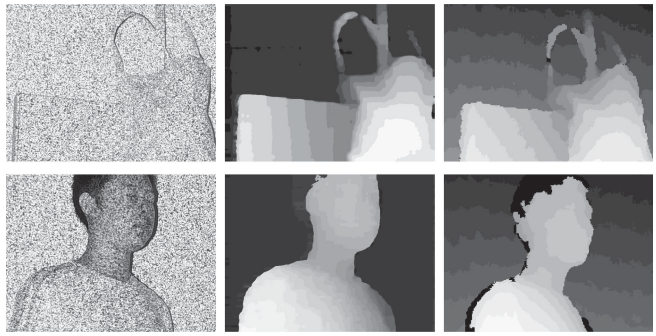
Scenes	ZNCC		Net in [9]		SLNET	
	Bad1.0	Bad2.0	Bad1.0	Bad2.0	Bad1.0	Bad2.0
Cable	14.14%	8.63%	12.01%	5.30%	9.21%	4.43%
Flowers	11.95%	7.59%	11.8%	6.93%	8.89%	5.37%
Mask	15.81%	11.57%	13.97%	9.14%	11.52%	8.06%

its generalization ability. The time consumed to generate a single depth map was about 70 ms on the Geforce TITAN Xp, including post-processing. The results are shown in Fig. 2.

The Bad1.0 and Bad2.0 values for all three cases are listed in Table I. The depth maps estimated with the classic ZNCC method were blurry, which were subsequently improved by the binocular matching network. However, because there are great differences between the binocular scenes and the patterned scenes, the binocular network cannot be used to extract the random pattern features. It is also seen that the accuracies of the SLNet were the highest.

To prove the practical value of the SLNet, real-world experiments were conducted. A projector (LightCrafter) and a camera (PointGrey) were used to construct a depth camera to estimate depth maps with SLNet, which are shown in Fig. 3. The depth maps of the same scenes estimated by Kinect-V1 were used for comparison. The resolution of the pattern used in our depth camera is similar with that in Kinect-V1. The SLNet learned the similarity between pattern patches, which is independent of a pattern's content and resolution. Because ground truths of the real-world scenes are difficult to obtain, only subjective evaluation is presented here. The depth maps obtained by the SLNet are more accurate because the depth layers are denser and the hollows at object boundaries are less.

There are further improvements that could be applied to the SLNet. Considering that the SLNet applies image patches to extract features, tiny objects that are much smaller than the patch size may not be clearly identified, such as the missing basket handle in the scene $Mask$. This shortcoming can be solved by using a high-density pattern and a high-resolution camera. Besides, although the short baseline between the projector and camera avoids shadowed areas, this may cause a decrease of accuracy. Fortunately, [23] provided a sub-pixel method that can solve this dilemma.



(a) (b) (c)

Fig. 3. Real-world experiments on objects and a student. (a) Patterned scenes, (b) depth maps acquired by the SLNet and (c) depth maps acquired by Kinect-V1.

V. CONCLUSION

Deep learning has become a powerful tool for stereo matching. In this letter, the SLNet was proposed to improve structured-light-based stereo matching, which contains three parts: an efficient Siamese network as the basic structure, pyramid pooling to extract features with different receptive fields and SENet to calculate the weights of features. To train the SLNet and obtain objective evaluation indices, a structured light dataset including patterned scenes and ground truths was created. Finally, simulated and real-world experiments were conducted to validate the superior performance of the SLNet in terms of depth estimation.

REFERENCES

[1] L. Keselman, J. I. Woodfill, A. Grunnetjepsen, and A. Bhowmik, "Intel(r) realsense(tm) stereoscopic depth cameras," in *Proc. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1267–1276.

[2] Z. Yang *et al.*, "Depth acquisition from density modulated binary patterns," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 25–32.

[3] Y. Zhang, Z. Xiong, Z. Yang, and F. Wu, "Real-time scalable depth sensing with hybrid structured light illumination," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 97–109, Jan. 2014.

[4] G. Rosman, D. Rus, and J. W. Fisher, "Information-driven adaptive structured-light scanners," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 874–883.

[5] D. Scharstein *et al.*, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit.*, 2014, vol. 8753, pp. 31–42.

[6] S. R. Fanello *et al.*, "Hyperdepth: Learning depth from structured light without matching," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 5441–5450.

[7] S. R. Fanello *et al.*, "Ultrastereo: Efficient learning-based matching for active stereo systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6535–6544.

[8] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, pp. 1–32, 2016.

[9] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5695–5703.

[10] H. Park and K. M. Lee, "Look wider to match image patches with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1788–1792, Dec. 2017.

[11] Z. Liang, H. Liu, L. Qiao, Y. Feng, and W. Chen, "Improving stereo matching by incorporating geometry prior into convnet," *Electron. Lett.*, vol. 53, no. 17, pp. 1194–1196, 2017.

[12] P. Knobelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock, "End-to-end training of hybrid CNN-CRF models for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1456–1465.

[13] 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 *Proc. Int. Conf. Comput. Vis. Workshop Geometry Meets Deep Learn.*, 2017, pp. 878–886.

[14] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 4040–4048.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," 2017, arXiv:1709.01507.

[17] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *Proc. Int. Conf. Pattern Recognit.*, 2017, pp. 378–383.

[18] A. Geiger, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[19] "Middlebury college stereo vision research page," 2014. [Online]. Available: http://vision.middlebury.edu/stereo/

[20] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4040–4048.

[21] Q. Du, R. Liu, and Y. Pan, "Depth extraction for a structured light system based on mismatched image pair rectification using a virtual camera," *IET Image Process.*, vol. 11, no. 11, pp. 1086–1093, 2017.

[22] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 257–269, 2010.

[23] Y. Pan, R. Liu, B. Guan, Q. Du, and Z. Xiong, "Accurate depth extraction method for multiple light-coding-based depth cameras," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 685–701, Apr. 2017.