ABSTRACT This paper proposes an unsupervised approach to construct a deep learning based stereo matching method using single-view videos (SMV). From videos, a set of corresponding points are computed between images, and image patches that center at the computed points are extracted. Negative and positive samples constitute a dataset to train a similarity network that is then used as a matching cost function. In addition, we propose a local-global matching cost network that exploits the first feature maps (local features) accompanying with last feature maps (global features) as output feature of the proposed network. The concatenated features are connected to full-connected layers and the network outputs a similarity measure of an image patch pair as a matching cost. Computed matching costs are aggregated using semi-global matching and cross-based cost aggregation, followed by sub-pixel interpolation, left-right consistency check, median and bilateral filtering. We evaluate the proposed stereo matching methods using popular stereo matching datasets, including KITTI 2012 and 2015, and Middlebury. We submit the disparity maps to their benchmark servers to evaluate the performance of SMV. We also compared the generalization of SMV and baseline methods using the training sets of the three datasets. The benchmark results show that SMV is the most accurate method among unsupervised approach, and it even outperforms several deep learning based stereo matching using supervised manner. The evaluation results of generalization show that SMV is comparative with the baseline method, MC-CNN, which is trained with supervision.

INDEX TERMS Stereo matching, unsupervised learning, video extraction.

I. INTRODUCTION
Stereo matching aims to reconstruct 3D information from stereo images. Given the left and right images, a stereo matching method estimates a disparity map, in which pixel intensities indicate the depth information from cameras to objects (that contains considered pixels). Figure 1 shows the illustration of stereo matching.

Stereo matching has been intensively researched for several decades because of its important applications for self-driving cars, 3-D reconstruction, view interpolation, and robot navigation [1], [2]. Scharstein and Szeliski [3] did an excellent survey of stereo matching methods and divided them into local and global methods. Local stereo matching methods normally includes matching cost computation, cost aggregation, and disparity computation steps, whereas global correspondence methods typically consist of matching cost computation and disparity optimization steps. Disparity refinement, such as sub-pixel interpolation via parabolic fitting, a left-right consistency check [4], and image filtering, can be used to improve the quality of the disparity map.

Zbontar and LeCun [5] proposed the first deep learning-based stereo matching cost that exploits a convolutional...
neural network. The matching costs are processed by cross-based cost aggregation (CBCA) [6] and semi-global matching (SGM) [7], followed by post-processing techniques including sub-pixel interpolation, a left-right consistency check, and median and bilateral filtering.

Since the dawn of deep neural networks, many deep learning-based methods are proposed for matching cost computation [8]–[11], cost aggregation [12], and post-processing [13]. Other work [14]–[19] proposed stereo matching methods that unified deep learning-based components and trained in an end-to-end fashion. Recently, disparity confidence methods [20]–[23] are introduced to improve the performance of stereo matching methods.

However, current deep learning methods require domain data for training. Supervised methods require left and right stereo images and ground truth, although unsupervised training methods require just left and right stereo images. This paper proposes training a matching cost network without requiring domain data. Corresponding image patches are extracted from single-view videos and subsequently employed as the training data. Collecting stereo matching dataset for different situations is not an easy task. Therefore, our approach helps to construct a stereo matching method easily.

In this paper we propose an approach to learn a matching cost network from videos. From single-view videos, feature matching points between frames are computed and then image patches for matching points are extracted to build a dataset of corresponding patches. After that, the dataset is used as a training data. In addition, we propose a local-global matching cost network that takes advantages of local features from the first layer.

The contributions of this paper are as follows:

- This paper proposes an approach to train a matching cost network by using single-view videos. This approach does not need stereo images as well as ground truth.
- A local-global matching cost network are proposed to exploit the benefit of using the first layer that can extract features similar to those of local binary patterns.
- Benchmark results on KITTI 2012, KITTI 2015, and Middlebury show that SMV even outperforms several learning-based stereo matching methods that use domain data for training. In addition, experimental results for testing generalization show that SMV performs robustly for different data sets and comparative with MC-CNN for different stereo pairs.

II. RELATED WORK

Traditional matching cost functions consist of the sampling-insensitive (SI) [24], absolute difference (AD), and squared difference (SD). These traditional functions suppose corresponding pixels between stereo images have the same intensity values. Therefore, they perform poorly when the stereo images are radiometrically distorted.

In many cases, intensity changes between stereo images are monotonically nonlinear wherein the orders of the intensity values are preserved. Matching cost functions that exploit ordinal values rather can tolerate this kind of intensity transformation. These matching cost functions include the rank and census transforms [25], the support local binary pattern (SLBP) [26], the fuzzy encoding pattern [27], and the soft rank transform [28].

Han et al. proposed a gradient-based matching cost function [29]. Scharstein et al. [30] introduced a gradient-based measure that can operate under the differences in the camera gain and bias. Wei et al. [31] proposed an intensity- and gradient-based matching method using hierarchical Gaussian basis functions. Zhou and Boulanger [32] introduced a Gaussian weighted sum of absolute difference based on the relative gradients. P. Pinggera et al. [33] proposed dense gradient features for cross-modal stereo.

Mutual information can tolerate any global intensity changes and has been exploited as a matching cost function in stereo matching. Kim et al. [34] proposed a pixel-wise matching cost for stereo matching based on mutual information. Hirschmuller [7] introduced a stereo matching method based on semi-global matching and mutual information. Heo et al. [35] introduced a stereo matching method where the
matching cost function combines mutual information with SIFT descriptor [36] in log-chromaticity color space.

Heo et al. [37] proposed adaptive normalized cross-correlation (ANCC), which is an improved version of normalized cross-correlation (NCC) and invariant to radiometric distortion. RANCC [38] is an improvement of the ANCC for the context that the effect of texture and noises on image regions. Dinh et al. [39] proposed a matching cost measure to address the non-linearity intensity transformation of pixels between the image patches.

A recent approach to compute matching cost is to use convolutional neural network to predict matching value for a patch pair. Reference [5] introduced a convolutional neural network that is trained for measuring the similarity of a patch pair. Reference [8] proposed a deep embedding model to predict matching cost which explicitly maps intensity values into an embedding feature space to estimate pixel dissimilarities. References [5] and [8] need stereo images and ground truth for training.

Reference [9] proposed a fast matching cost network that uses a product layer for a siamese architecture. Reference [10] proposed an unsupervised approach to estimate matching cost by exploiting left-right consistency check to guide the training process. Reference [11] proposed a weakly supervised techniques for training patch similarity which uses properties of the optical sensor and a rough scene knowledge. Li and Yuan [62] introduced a stereo matching method that is an unsupervised learning method and aware of occlusion problem. Joung et al. [63] proposed a stereo matching method that is trained in an unsupervised manner using confidential correspondence consistency. Tonioni et al. [61], [64] introduced stereo matching methods for domain adaptation using stereo images without ground truth.

The output of the matching cost computation step is a matching cost image space $C$ for which $C_d(p)$ is the matching cost value of a pixel $p$ in the reference image, e.g., the left image of a stereo pair, and at a disparity hypothesis $d$. From $C$, a disparity value for $p$ can be obtained by using a winner-takes-all strategy, as follows:

$$D_E(p) = \arg \min_d (C_d(p)),$$

where $D_E$ is an estimated disparity map. Applying a winner-takes-all strategy is the simplest way to obtain a dense disparity map.

III. SMV

A. DATASET CONSTRUCTION FROM VIDEOS

In this subsection, we present an approach to construct a dataset from videos which is then used to train a matching cost network. Given a video, we extract two frames. To reduce the scene correlation between frames, the two selected frames should not be continuous in the video. We use the SIFT to compute corresponding points between the frames, as shown in Fig. 2(a). For each pair of the corresponding points, we extract image patches whose center pixels are the corresponding points, as shown in Fig. 3.

Brightness and contrast adjustment changes the brightness and contrast by setting the image patch $P$ to

$$P \leftarrow P \cdot \text{contrast} + \text{brightness},$$

where addition and multiplication are element-wise operations. Rotation rotates the patch by rotation degrees, whereas translation translate the patch in the vertical direction by translation. Scaling scales the patch by scaling, and shearing shears the patch in the horizontal direction by shearing.

Elastic distortion [40] is commonly used to generate images that are feasible and label preserving in classification. Elastic distortion distorts an image patch by the intensity of transformation $ED_{\text{alpha}}$ and the smoothness for transformation $ED_{\text{sigma}}$. Noise block addition adds a block of random values to an image patch. The position of the block is selected randomly. Foreshortening is inspired from different view-point of stereo cameras. In foreshortening, first we crop left or right side of an image patch by cropping and $p_{l/r}$, and following that the cropped patch is resized to the same size as the original patch. Fig. 4 shows the illustration of the elastic distortion and left and right foreshortening for an input patch.

In order to prepare a training data of positive and negative example, each image patch extracted from an image is undergone through the transformation pipeline two times.
with different random setting of parameters. The two transformed patches forms a synthesized pair of corresponding image patches (positive example). The negative example is created by extracted a new image patch that is far from the considered image patch at a distance, \(d_{\text{data}}\).

**B. LOCAL-GLOBAL MATCHING COST NETWORK**

We propose a local-global matching cost network that exploits the first convolution layer, as shown in Fig. 4. The first convolution layer extracts low-level features of an image patch which are edge-like features. Each convolution kernel in the first convolution layer often extracts different features. The features in the last layers are considered as global features that extract high-level features of the image patch.

In stereo matching, hand-crafted feature extraction, such as census, rank, slbp, have been successfully operated for stereo images in different conditions. Each of these feature extractors are designed to obtain different features that highly discriminative.

The feature maps of the first convolution layer are somewhat similar to the output of the hand-crafted feature extractors, and even can extract more number of features because the number of feature maps are set, such as 32 or 64, and computed automatically.

As a result, our idea is to combine the local feature (feature maps of the first convolutional layer) and global features (output of the last layer) to increase the discriminative power. The architecture of our proposed network is as follow:

Fig. 4 shows the architecture of the proposed multi-patch matching cost network. The architecture of sub-networks consist of a number of convolution layers followed by rectified linear unit layer (RELU). The resulting four vectors are concatenated and forwardly propagated through a series of fully connected layer followed by RELU. The final output of network is fed to a non linear activation function sigmoid to produce a similarity score between the input patches. The binary cross-entropy loss is used for training. Let \(x\) denote the output of the network for one training example and \(y\) denote the class of that training example; \(y = 1\) if the example belongs to the positive class and \(y = 0\) if the example belongs to the negative class. The binary cross-entropy loss \(L\) for that example is defined as

\[
L = x \log(y) + (1 - x) \log(1 - y).
\]  

The hyperparameters of the proposed network are the number of fully-connected layers (\(num_{fc\_layers}\)), and the number of units in each fully-connected layer (\(num_{fc\_units}\)), the number of feature maps in each layer (\(num_{fmaps}\)),

![FIGURE 4. The architecture of the proposed multi-patch matching cost network.](image)

![FIGURE 5. Model testing for evaluation of the proposed stereo matching method. Refinement steps include sub-pixel enhancement, left-right consistency check, followed by median and bilateral filtering.](image)
the number of convolutional layers (num\_layers), the size of the convolution kernels (kernel\_size), the size of the input patch (input\_patch\_size).

The hyperparameters of aggregation and post-processing methods include cbca\_distance, cbca\_num\_iters\_1, cbca\_num\_iters\_2, which denote for similarity threshold for pixel intensities, number of iteration of cross-based cost aggregation before SGM, and number of iteration of cross-based cost aggregation after SGM, respectively. sgm\_P1, sgm\_P2, sgm\_Q1, and sgm\_Q2 stands for the first smoothness parameter of SGM, the second smoothness parameter of SGM, a factor 1 used for changing sgm\_P1/sgm\_P2, and a factor 2 used for changing sgm\_P1/sgm\_P2, respectively. sgm\_V and sgm\_D denote for reduction of sgm\_P1 by a factor of sgm\_D when considering vertical direction and pixel intensity threshold for changing sgm\_P1/sgm\_P2. Finally, blur\_sigma and blur\_threshold stand for standard deviation for a post-processing filter and threshold for a post-processing filter.

In this paper, we have set 11 × 11 image patches as input to the network. The first convolutional layer is used to extract feature maps from the input patches that are then considered as local image features. The five convolutional layers are with 3 × 3 kernel and 112 feature maps. A 224-length vector is formed by concatenating the two 112-length feature vectors. After that, the 224-length vector is passed through three fully-connected layers with 384 units each. The final fully-connected layer projects the output to a single number that is the similarity score. A matching cost is just a negative value of the similarity score.

C. COST AGGREGATION AND POST-PROCESSING METHODS

The outcome of the local-global matching cost network is a matching cost space that is then aggregated and post-processed to produce the final disparity map. We follow the pipeline introduced in [41] (used later by MC-CNN [46]) as shown in Fig. 5. The pipeline suggests to use CBCA and SGM to aggregate the matching costs. Then, sub-pixel interpolation, left-right consistency check to detect invalid pixels, followed by median and bilateral filtering. Similar to MC-CNN, we use CBCA before and after SGM.
IV. EXPERIMENTAL RESULTS

We evaluated the proposed stereo matching method using KITTI 2012, 2015, and Middlebury datasets. We uploaded the results for the three datasets to their online benchmark servers.

To evaluate the generalization performance of the testing stereo matching methods, we used different datasets for training and testing steps and compared with MC-CNN, AD, and Census. All the testing methods use the same pipeline of cost aggregation and post-processing methods. We followed the parameter setting in [46] for MC-CNN, AD, and Census.

For the proposed matching cost network, we used grid search method to select parameter setting using the mixed dataset, constructed from KITTI and Middlebury training datasets. For each parameter, we first estimated a feasible range and a value step for the grid search method. After that, we chose the parameter setting that had the best performance on the mix dataset. Table 1 shows the parameter setting for the proposed stereo matching method, and the parameters were fixed for all of our experiments.

We used Cityscapes video datasets [42] for training the proposed matching cost network. Specifically, we used three single-view sequences (stuttgart_00, stuttgart_01,}

| Name                | Value      | Name        | Value      |
|---------------------|------------|-------------|------------|
| **Patch Transformation** |            |             |            |
| data_distance       | 5          | scaling     | [0.9, 1]   |
| contrast            | [1.1, 1]   | shearing    | [0.0, 1]   |
| brightness          | [0.0, 4]   | EDP_alphA   | [1.7]      |
| rotation            | [10, 10]   | EDP_sigma   | [1.7]      |
| translation         | [-1, 1]    | cropping    | [0.3]      |
| **Network**         |            |             |            |
| input_patch_size    | 11 x 11    | kernel_size | 3          |
| num_layers          | 5          | num_fc_layers | 3         |
| num_fmaps           | 112        | num_fc_units | 584       |
| **Aggregation and Post-Processing** |            |             |            |
| oboa_distance       | 14         | oboa_intensity | 0.02    |
| oboa_num_sters_1,2  | 2, 16      | sgm_Q0, V   | 9, 2.75    |
| sgm_P1              | 1.3        | sgm_D       | 0.13       |
| sgm_P2              | 18.1, 4.5  | blur_sigma  | 1.7, 2     |
| sgm_Q1              |            | blur_threshold | 2        |
TABLE 2. KITTI 2012 benchmark results in error rate (%) for SMV. Out-Noc is the percentage of erroneous pixels in non-occluded regions, and Out-All is the percentage of erroneous pixels in total. Avg-Noc is the ratio between the average disparity and end-point error in non-occluded regions, and Avg-All is the ratio between the average disparity and end-point error in total.

| Method      | Out-Noc | Out-All | Avg-Noc | Avg-All | Learning Approach | Data Required |
|-------------|---------|---------|---------|---------|-------------------|---------------|
| MC-CNN-act  | 2.43 %  | 3.63 %  | 0.7 px  | 0.9 px  | Yes               | Training data |
| MC-CNN-fast | 2.61 %  | 3.84 %  | 0.8 px  | 1.0 px  | Yes               | Training data |
| MC-CNN-WS   | 3.02 %  | 4.45 %  | 0.8 px  | 1.0 px  | Yes               | Stereo image data |
| DispNetC    | 4.11 %  | 4.65 %  | 0.9 px  | 1.0 px  | Yes               | Training data |
| DLP         | 5.28 %  | 7.21 %  | 1.2 px  | 2.0 px  | Yes               | Training data |
| OASM-Net    | 6.39 %  | 8.60 %  | 1.3 px  | 2.0 px  | Yes               | Stereo image data |
| Deep-Raw    | 8.93 %  | 11.07 % | 3.9 px  | 4.9 px  | Yes               | Training data |
| PR-SM+D     | 4.02 %  | 4.87 %  | 0.9 px  | 1.0 px  | No                | No            |
| PCBP        | 4.04 %  | 5.37 %  | 0.9 px  | 1.1 px  | No                | No            |
| CoR-Conf    | 4.49 %  | 5.26 %  | 1.0 px  | 1.2 px  | No                | No            |
| wSGM        | 4.97 %  | 6.18 %  | 1.3 px  | 1.6 px  | No                | No            |
| SGM         | 5.76 %  | 7.00 %  | 1.2 px  | 1.3 px  | No                | No            |
| S+GF (Cen)  | 9.03 %  | 11.21 % | 2.1 px  | 3.4 px  | No                | No            |
| SMV         | 3.84 %  | 5.04 %  | 0.9 px  | 1.1 px  | Yes               | No            |

stuttgart_02) which include about 2900 images totally with 2048 × 1024 resolution. Let $i$ be the frame index of a video. We use a image pair of $I_i$ and $I_{i+2}$ for compute corresponding points using the SIFT. Totally, about 12.5 millions of point pairs are detected and hence about 25 millions of sample patches (including positive and negative samples) are extracted.

We exploited stochastic gradient descent to optimize the cross-entropy loss of the proposed network training. The network was trained for 22 epochs with the learning rate initially set to 0.003 and decreased by a factor of 10 on the 18th. The training dataset was shuffled prior to learning for each epoch, and the batch size was set to 128.

Disparity maps were evaluated using the average proportion of erroneous pixels in all zones, except occlusions. We used the KITTI error thresholds ($th = 3$) pixel and Middlebury error thresholds ($th = 1$). The error rate (%) was calculated as

$$E_d = \frac{100}{|I_{nocc}|} \sum_{p \in I_{nocc}} \begin{cases} 0, & \text{if } |D_E(p) - D_G(p)| \leq th \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where $I_{nocc}$ is the set of all non-occluded pixels, $|I_{nocc}|$ is the number of pixels in $I_{nocc}$, and $D_G(p)$ and $D_E(p)$ are the ground truth and estimated disparity at $p$, respectively.

A. QUANTITATIVE RESULTS USING STEREO MATCHING BENCHMARKS

The KITTI 2012 and 2015 datasets [43], [44] include outdoor stereo images with sparse ground truth (approximately 50% of the pixels). The KITTI 2012 dataset has 194 stereo pairs for training and 195 stereo pairs for testing, and the KITTI 2015 dataset provides 200 stereo images for training. Middlebury provides indoor stereo images with dense ground truth.

Since the KITTI and Middlebury servers constrain the limited numbers of submissions, we used the servers to evaluate the results for the complete version of the proposed stereo matching method. Tables 2, 3, and 4 show the results of SMV in the KITTI 2012, 2015, and Middlebury benchmarks, respectively. The proposed stereo matching method significantly outperformed SGM and ELAS methods that are considered as baseline methods for traditional stereo matching approach. In addition, for all the three benchmark results, The proposed stereo matching method performed better several deep learning-based stereo matching methods, even though The proposed stereo matching method is constructed without using a single stereo pair. Figs. 6 and 7 show some disparity maps of SMV downloaded from KITTI server for the KITTI 2012 and 2015 datasets, respectively.

B. GENERALIZATION

In this subsection, we compared the performance of SMV, MC-CNN, AD, and Census methods for data generalization. In other words, MC-CNN, AD, and Census use a training data to train and/or tune parameters of a method, and then are evaluated using different data. For AD and census, the parameters of the post-processing techniques were set the same as in the MC-CNN paper. Let MC-CNN_K15, MC-CNN_K15, and MC-CNN_MB denote MC-CNN with accurate architecture and being trained using the KITTI 2015, KITTI 2015, and Middlebury training sets, respectively. In addition, to evaluate the effective of the multi-patch matching cost network in SMV, we designed a version of SMV that the number of input patch is set to 1, denoted SMV(-). Except the cropped size of $9 \times 9$, proposed method(-) parameters were set the same as those of SMV.

Figs. 8 and 9 show the quantitative results of the testing stereo matching methods for the first 100 stereo pairs for KITTI 2012 and 2015 training sets, respectively. SMV performed better AD and census significantly, and had a comparative performance with the MC-CNN variants that require training data. Because of using multi-patch network, SMV performed much more robustly than SMV(-).
TABLE 3. KITTI 2015 benchmark results in error rate (%) for SMV. D1-bg is the percentage of outliers averaged only over background areas. D1-fg is the percentage of outliers averaged only over foreground areas, and D1-all is the percentage of outliers averaged over all ground truth pixels.

| Method         | D1-bg | D1-fg | D1-all | Learning Approach | Data Required  |
|----------------|-------|-------|--------|-------------------|----------------|
| MC-CNN-acr [46] | 2.89 % | 8.88 % | 3.89 % | Yes               | Training data  |
| DispNet [47]   | 4.32 % | 4.41 % | 4.34 % | Yes               | Training data  |
| MADNet [65]    | 3.75 % | 9.20 % | 4.66 % | Yes               | Stereo image data |
| MC-CNN-WS [11] | 3.78 % | 10.93 % | 4.97 % | Yes               | Stereo image data |
| CBMV [54]      | 4.17 % | 9.53 % | 5.06 % | No                | Training data  |
| DeepCostAggr [55] | 5.34 % | 11.35 % | 6.34 % | Yes               | Training data  |
| OASM Net [63]  | 6.89 % | 19.42 % | 8.98 % | Yes               | Stereo image data |
| OSF [56]       | 4.54 % | 12.03 % | 5.79 % | No                | No             |
| pSGM [57]      | 4.84 % | 11.64 % | 5.97 % | No                | No             |
| SGM [7]        | 5.06 % | 13.00 % | 6.38 % | No                | No             |
| ELAS [58]      | 7.86 % | 19.04 % | 9.72 % | No                | No             |

SMV 4.64 % 10.33 % 5.59 % Yes No

TABLE 4. Middlebury benchmark results in error rate (%) for SMV. Austr, AustrP, Bicyc2, Class, ClassE, Compu, Crusa, CrusaP, Djemb, DjembL, Hoops, Livgrm, Nkuba, Plants, Stairs are testing stereo pairs. The symbol † denotes that a stereo matching method requires domain data for training, whereas ‡ denotes that the method requires left and right image data for training. § denotes that the domain data is not compulsory for constructing the method.

| Method         | Avg | Austr | AustrP | Bicyc2 | Class | ClassE | Compu | Crusa | CrusaP | Djemb | DjembL | Hoops | Livgrm | Nkuba | Plants | Stairs |
|----------------|-----|-------|--------|--------|-------|--------|-------|-------|--------|-------|--------|-------|--------|-------|--------|--------|
| MC-CNN-acrmitted [46] | 8.08 | 5.59 | 4.55 | 5.96 | 2.83 | 11.4 | 5.81 | 8.32 | 8.89 | 2.71 | 16.3 | 14.1 | 13.2 | 13.0 | 6.40 | 11.1 |
| MC-CNN-fast [5] | 9.47 | 7.35 | 5.07 | 7.18 | 4.71 | 16.8 | 8.47 | 7.37 | 6.97 | 2.82 | 20.7 | 17.4 | 15.4 | 15.1 | 7.90 | 12.6 |
| CBMV [54]      | 11.1 | 6.07 | 5.22 | 8.09 | 4.05 | 18.7 | 9.31 | 10.7 | 9.61 | 3.11 | 33.7 | 15.6 | 17.5 | 17.1 | 10.1 | 14.4 |
| MC-CNN-WS [11] | 12.1 | 14.8 | 7.20 | 11.1 | 7.62 | 15.9 | 11.8 | 11.5 | 9.01 | 3.89 | 19.7 | 20.5 | 16.3 | 16.3 | 12.1 | 18.3 |
| UCNN [10]      | 20.5 | 44.8 | 9.77 | 13.6 | 18.2 | 36.5 | 12.8 | 23.4 | 12.4 | 9.22 | 39.5 | 30.5 | 24.8 | 21.2 | 19.1 | 32.3 |
| iResNet [59]   | 22.9 | 28.3 | 9.19 | 15.8 | 19.3 | 35.1 | 11.3 | 27.7 | 16.8 | 15.2 | 54.7 | 27.6 | 19.3 | 21.5 | 31.9 | 51.6 |
| DSGCA [60]     | 33.8 | 42.9 | 20.9 | 23.6 | 50.2 | 45.5 | 27.6 | 42.0 | 36.0 | 21.0 | 50.2 | 44.2 | 33.3 | 34.6 | 38.4 | 46.8 |
| PSMNet-ROB [16] | 42.1 | 33.0 | 23.1 | 30.1 | 31.4 | 54.8 | 36.7 | 48.7 | 48.3 | 28.3 | 80.8 | 53.5 | 36.9 | 38.6 | 63.9 | 71.2 |
| SMV [16]       | 15.3 | 8.94 | 11.4 | 22.2 | 3.45 | 5.76 | 17.7 | 31.8 | 7.30 | 22.4 | 21.1 | 19.9 | 10.8 | 37.1 | 6.99 | 37.7 |

FIGURE 8. Error rates of the testing stereo matching methods for the first 100 stereo pairs of the KITTI 2012 training set.
not use the multi-patch network performed poorly, with error rates approximately double those of SMV. In all the cases, even though SMV did not use training data, its error rate is nearly as good as MC-CNN variants, with slightly larger error rates.

C. USING LOCAL BINARY PATTERNS
In this subsection, we evaluate the performance of combining the handcrafted features and the feature maps of convolutional networks. Specifically, instead of combining feature maps from the first and last convolutional layers, we computed census, rank, and SLBP transforms for input images and then concatenate them with the last convolutional feature maps. We denoted this method as SMV_LBP. Figure 10 shows an illustration of census and rank transforms for an image.

We used window size $(3 \times 3)$ for both census and rank transforms. We normalized the transformed images before concatenating with feature maps, computed from the last convolutional layer. Figure 11 shows the quantitative results of SMV_LBP using the KITTI 2012 and 2015 training datasets. SMV_LBP had marginally better performance than SMV(-) and performed worse than SMV. The reason is that census and rank transforms are just two matrix instances of a $(3 \times 3)$ convolution matrix and their weight values are fixed. In contrast, SMV extracted 112 feature maps using 112 convolution matrices, in which weights were selected optimally for a training dataset.

D. SMV EVALUATION
We evaluated the stereo matching methods using their raw matching costs on the KITTI 2012 and 2015 datasets. In addition, we trained SMV in a supervised manner using KITTI 2012, KITTI 2015, and Middlebury training datasets, denoted as SMV_K12, SMV_K15, and SMV_MB, respectively. For a fair comparison, we used the same data augmentation as in MC-CNN.
TABLE 5. Error rates for the raw matching costs of the testing stereo matching methods using KITTI 2012 and 2015 training datasets.

|            | AD   | Census | MC-CNN_K12 | MC-CNN_K15 | MC-CNN_MB | SMV   | SMV_K12 | SMV_K15 | SMV_MB |
|------------|------|--------|------------|------------|-----------|-------|---------|---------|--------|
| K12        | 41.8%| 54.7%  | 14.04%     | 21.6%      | 16.03%    | 22.6% | 11.85%  | 17.47%  | 15.06% |
| K15        | 40.3%| 49.2%  | 18.2%      | 12.3%      | 18.5%     | 20.3% | 14.5%   | 16.8%   | 18.01% |

TABLE 6. Error rates for SMV with different values for input_patch_size using KITTI 2015 training dataset.

|                   | (7 × 7) | (9 × 9) | (11 × 11) | (13 × 13) | (15 × 15) | (17 × 17) | (19 × 19) | (21 × 21) | (23 × 23) | (25 × 25) |
|-------------------|---------|---------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| SMV_woo_pp       | 14.21%  | 11.87%  | 10.83%    | 9.29%     | 8.93%     | 8.53%     | 8.26%     | 8.13%     | 7.85%     | 7.62%     |
| SMV               | 6.20%   | 5.23%   | 4.62%     | 4.83%     | 3.38%     | 6.14%     | 6.36%     | 6.71%     | 6.93%     | 7.17%     |

Table 6 shows the error rates for KITTI 2012 (K12) and KITTI 2015 (K15) training datasets. AD and Census had the worst performance, and AD even outperformed Census. These performance of AD and Census in our experiments are similar to those in [46]. The supervised versions of SMV outperformed MC-CNN for all corresponding datasets. That validates the effectiveness of the use of the local and global CNN features in SMV.

E. SENSITIVITY ANALYSIS

In this subsection, we present the way to select parameter values and analyze the effect of different parameter configurations to SMV. As shown in Table 1, the SMV network has six parameters, including input_patch_size, num_clayers, num_fmaps, kernel_size, num_fc_layers, and num_fc_units.

For the kernel size kernal_size, using two 3 × 3 kernels have the same receptive field with a 5 × 5 kernel. Therefore, these days, a 3 × 3 kernel size is commonly used for CNN. SMV and MC-CNN share the three common parameters, which are num_fmaps, num_fc_layers, and num_fc_units. In our work, we have selected the values for the three parameters, as recommended by the MC-CNN work. There are two reasons for this. The first reason is that the three parameters were carefully selected by using the grid search in the MC-CNN work. The second reason is that using the same values could show the effectiveness of exploiting the local-global features in SMV.

Here, we evaluated to the effect of using different values for input_patch_size in SMV. The possible values for input_patch_size could be any positive number that is smaller than the width and height of an image. However, We evaluated SMV with commonly used ranges, including (7 × 7), (9 × 9), (11 × 11), (13 × 13), (15 × 15), (17 × 17), (19 × 19), (21 × 21), (23 × 23), and (25 × 25), as shown in Table 6, where SMV_woo_pp is SMV without post-processing. Larger values for input_patch_size are robust to outliers, but sensitive to object boundaries and slanted regions (such as road surfaces) [39]. As a result, SMV_woo_pp performed better with larger values for input_patch_size. Therefore, input_patch_size selection is a tradeoff the the problems. With our experiments, using (11 × 11) kernel for input_patch_size had the smallest error rates for SMV.

V. CONCLUSIONS

This paper proposed an approach for stereo matching method that uses single-view videos in an unsupervised manner. In addition, we proposed a matching cost network that exploits explicitly local and global features. The proposed stereo matching method was evaluated using commonly used
datasets in stereo matching, including KITTI 2012, KITTI 2015, and Middlebury. Experimental results the benchmarks showed that the proposed method had the best performance among unsupervised methods and outperformed several supervised methods. It also performed well across different datasets.

In future work, we plan to investigate deeply image similarity functions in traditional approaches as well as learning based ones. Applications of similarity functions in computer vision and ways to construct them in case of datasets available in different domains.

REFERENCES

[1] E. Trucco and A. Verri, *Introductory Techniques For 3-D Computer Vision*, Upper Saddle River, NJ, USA: Prentice-Hall, 1998.

[2] B. Cygankiewicz and P. Siebert, *Introduction to 3D Computer Vision Techniques and Algorithms*, Hoboken, NJ, USA: Wiley, 2009.

[3] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, 2002.

[4] P. N. Hong, C. W. Ahn: *Unsupervised Learning for Stereo Matching Using Single-View Videos*, IEEE Access, vol. 8, 2020, pp. 73814–73829.

[5] Z. Jie, P. Wang, Y. Ling, B. Zhao, Y. Wei, J. Feng, and W. Liu, “Support local pattern and its application to disparity improvement and texture classification,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 263–276, Feb. 2014.

[6] V. Q. Dinh, V. D. Nguyen, H. Van Nguyen, and J. W. Jeon, “Fuzzy encoding pattern for stereo matching cost,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1215–1228, Jul. 2016.

[7] V. Q. Dinh, J. W. Jeon, and C. C. Pham, “Matching cost function using robust soft rank transformations,” *IET Image Process.*, vol. 10, no. 7, pp. 561–569, Jul. 2016.

[8] H. Han, X. Han, and F. Yang, “An improved gradient-based dense stereo correspondence algorithm using guided filter,” *Optik*, vol. 125, no. 1, pp. 115–120, Jan. 2014.

[9] J. Kim, “Visual correspondence using edge minimization and mutual information,” in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, vol. 2, Oct. 2003, pp. 1033–1040.

[10] Y. S. Heo, K. M. Lee, and S. U. Lee, “Plenoptic information-based stereo matching combined with SIFT descriptor in log-chromaticity color space,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2016, pp. 454–455.

[11] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[12] Y. S. Heo, K. M. Lee, and S. U. Lee, “Robust stereo matching using adaptive normalized cross-correlation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 807–822, Apr. 2011.

[13] V. Q. Dinh, C. C. Pham, and J. W. Jeon, “Robust adaptive normalized cross-correlation for stereo matching cost computation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp. 1421–1434, Jul. 2017.

[14] X. Mei, H. Han, and F. Yang, “Robust matching cost function for stereo correspondence using matching by tone mapping and adaptive orthogonal integral image,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5416–5431, Dec. 2015.

[15] P. Y. Simard, D. Steinkears, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *Proc. 7th Int. Conf. Document Anal. Recognit.*, Edinburg, U.K., 2003, pp. 958–963.

[16] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, “On building an accurate stereo matching system on graphics hardware,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Barcelona, Spain, Nov. 2011, pp. 467–474.

[17] M. Cords, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[43] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Providence, RI, USA, Jun. 2012, pp. 3354–3361.

[44] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4009–4017.

[45] R. Spangenberg, T. Langner, and R. Rojas, “Weighted semi-global matching and center-symmetric census transform for robust driver assistance,” in Proc. Int. Conf. Comput. Anal. Images Patterns, 2013, pp. 34–41.

[46] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. D. Stefano, “Real-time self-adaptive deep stereo,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 195–204.

[47] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, “Continuous Markov random fields for robust stereo estimation,” in Proc. 12th Eur. Conf. Comput. Vis., vol. 7576, pp. 45–58.

[48] A. Kuzmin, D. Mikushin, and V. Lempitsky, “End-to-end learning of cost-volume aggregation for real-time dense stereo,” in Proc. IEEE 27th Int. Workshop Mach. Learn. for Signal Process. (MLSP), Sep. 2017, pp. 1–6.

[49] M. Menze and A. Geiger, “Joint 3D estimation of vehicles and scene flow,” in Proc. ISPRS Workshop Image Sequence Analysis (ISA), 2015, pp. 427–434.

[50] S. Meister, “Outdoor stereo camera system for the generation of real-world benchmark data sets,” Opt. Eng., vol. 51, no. 2, Mar. 2012, Art. no. 021107.

[51] J. Zbontar and Y. LeCun, “Stereo matching by training a convolutional neural network to compare image patches,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 2287–2318, Jan. 2016.

[52] 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 Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4040–4048.

[53] V. D. Nguyen, H. V. Nguyen, and J. W. Jeon, “Robust stereo data cost with a learning strategy,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 2, pp. 248–258, Feb. 2017.

[54] C. Vogel, K. Schindler, and S. Roth, “Piecewise rigid scene flow,” in Proc. IEEE Int. Conf. Comput. Vis., Sydney, NSW, Australia, Dec. 2013, pp. 1377–1384.

[55] Y. Lee, M.-G. Park, Y. Hwang, Y. Shin, and C.-M. Kyung, “Memory-efficient parametric semiglobal matching,” IEEE Signal Process. Lett., vol. 25, no. 2, pp. 194–198, Feb. 2018.

[56] A. Li and Z. Yuan, “Occlusion Aware Stereo Matching via Cooperative Unsupervised Learning,” in Proc. Asian Conf. Comput. Vis. (ACCV), 2018, pp. 197–213.

[57] S. Joung, S. Kim, K. Park, and K. Sohn, “Unsupervised stereo matching using confidential correspondence consistency,” IEEE Trans. Intell. Transp. Syst., early access, May 24, 2019.

[58] I. K. Park, “Deep self-guided cost aggregation for stereo matching,” Pattern Recognit. Lett., vol. 112, pp. 168–175, Sep. 2018.

[59] D. Scharstein, T. Taniai, and S. N. Sinha, “Semi-global stereo matching with surface orientation priors,” in Proc. Int. Conf. 3D Vis. (3DV), Oct. 2017, pp. 215–224.

[60] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, “Unsupervised adaptation for deep stereo,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, Oct. 2017, pp. 1614–1622.

[61] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2012, pp. 3354–3361.

PHUC NGUYEN HONG received the B.S. degree in science service from the Auckland University of Technology (AUT), New Zealand, in 2013, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, in 2017. She currently holds a postdoctoral position at the Department of Information and Communication Engineering, GIST. Her research interests include optimization, machine learning, and computer vision.

CHANG WOOK AHN (Member, IEEE) received the Ph.D. degree from the Department of Information and Communication Engineering, GIST, in 2005. From 2005 to 2007, he worked at the Samsung Advanced Institute of Technology, South Korea. He was an Associate Professor with the Department of Computer Science and Engineering, Sungkyunkwan University, South Korea, from 2008 to 2017. He is currently a Professor at the Artificial Intelligence Graduate School, GIST, South Korea. His research interests include nature-inspired problem solving, evolutionary/quantum machine learning, and creativity-model learning intelligence, such as music and art generation.