Robust Stereo Data Cost With a Learning Strategy

Vinh Dinh Nguyen, Hau Van Nguyen, and Jae Wook Jeon, *Member, IEEE*

***Abstract*—The performance of stereo matching algorithms strongly depends on the quality of the stereo data/matching cost. Most state-of-the-art data costs require expert knowledge for the design of a transformation function, such as census for handling gray-level changes monotonically, adaptive normalized cross correlation for handling Lambertian cases, guided filtering for preserving edge information, and local density encoding for handling illumination differences. However, it is difficult to design a complex transformation function to handle unknown factors that often occur in driving conditions such as snow, rain, and sun. Therefore, this paper has investigated the deep learning strategy to develop a novel stereo matching cost model without using much expert knowledge. Experimental results show that the proposed deep learning model obtains better results than the state-of-the-art stereo matching cost as judged by the standard KITTI benchmark, Middlebury, and HCI datasets.**

***Index Terms*—Stereo matching cost, deep learning, unsupervised training, unlabeled data.**

I. Motivation and Main Contributions

A STEREO matching algorithm consists of four main steps: data cost, aggregation cost, optimization, and post processing; the data cost is the most important step among the four. Many algorithms have been proposed to produce a robust stereo data cost under various conditions. These include the Census and Rank transforms [1], adaptive normalized cross correlation (ANCC) [2], guided image filtering [3], and local density encoding (LDE) [4]. The main objective of those data costs is to extract more accurate and robust information from the local region. However, expert knowledge is required to develop those data costs under specific conditions. For example, Census works well when the gray-level changes monotonically; however, it fails in uniform regions or with larger illumination variations. ANCC can handle neither multiple illumination conditions nor non-Lambertian reflectance objects. Thus, those data costs cannot function well under conditions that vary from those for which they were designed. A combination of unknown factors, such as rain, snow, and sun, is significantly challenging to the development of a robust stereo data cost in the real world, which raises an interesting question. Is it possible to develop a robust data cost model that does not require much expert knowledge under a specific condition? Recently, deep learning performed optimally in various applications, such as natural language processing, texture classification, and object recognition [5]–[7]. The performance of existing applications has been significantly increased, by 30% or more, through the use of the deep learning approach. Therefore, this paper investigates a deep learning strategy to improve the performance of existing stereo matching algorithms. The main contributions of this paper are as follows.

1) This research is the first to apply a deep learning strategy-based unsupervised approach to develop a robust transformation function to compute stereo matching costs (it is different from the work in [8], which used supervised learning to develop a new data cost function).
2) We investigate how a deep learning strategy can be applied to calculate stereo data costs.
3) The proposed transformation function outperforms existing state-of-the-art data costs using the KITTI benchmark [9] and the Middlebury [10] and HCI datasets [11].
4) The proposed approach can improve the performance of existing stereo data costs without requiring their structures to be modified.

The remainder of this paper is organized as follows. Section II provides a brief summary of existing stereo matching algorithms and their limitations. The proposed cost function for stereo matching is presented in Section III. Our experimental results are presented in Section IV. Finally, the conclusion and proposed future work are presented in Section V.

II. Related Work

Stereo vision is an important factor used to detect and estimate the distance to the preceding vehicle in driving assistance systems [12]. Therefore, many stereo matching algorithms have been proposed to produce an accurate disparity map [13]–[19]. Without loss of generality, stereo matching algorithms can be divided into three categories: local, semi-global/non-local, and global algorithms. Local algorithms often provide fast processing time suitable for real-time systems. Global algorithms are usually more accurate than local algorithms, but they require much longer processing time. Semi-global/non-local algorithms are considered an intermediate state between local and global algorithms that can achieve both real-time processing and adequately high accuracy.

Most stereo algorithms contain four main steps: stereo data cost, aggregation cost, optimization, and post processing. Stereo data cost computation is the most important step because it produces the input for the remaining steps. The performance of aggregation cost and optimization is strongly affected by the quality of the stereo data cost, as shown in Fig. 1. In this test,

Manuscript received June 19, 2015; revised December 11, 2015 and January 27, 2016; accepted April 30, 2016. This work was supported by Samsung Research Funding Center of Samsung Electronics under Project SRFC-IT1402-12. The Associate Editor for this paper was K. Wang.

The authors are with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon 440-746, South Korea (e-mail: vinhnd@skku.edu; haunguyen@skku.edu; jwjeon@yurim.skku.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2563661

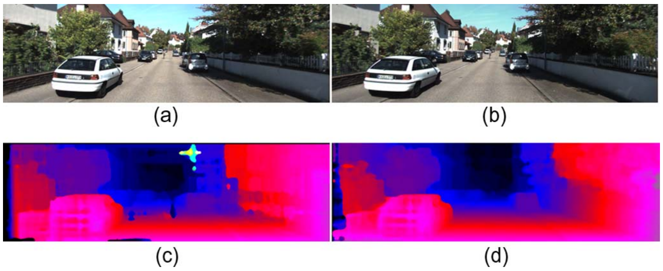


Fig. 1. Experimental results with the KITTI dataset. (a) Left and (b) right images. (c) Disparity results of SGM with SAD data cost (RMS = 21.39) and (d) SGM with the proposed data cost (RMS = 7.15).

the result of semi-global matching (SGM) [19] with the sum of absolute differences (SAD) data cost is worse than SGM with our proposed data cost under real world conditions. Many stereo methods are based on the assumption that the intensities of the left and right images have similar values under ideal conditions. However, that assumption is false under realistic outdoor conditions. Therefore, many stereo data costs have been proposed to extract more stable information from the local region; these include pixel-based and window-based methods, sampling-insensitive absolute differences, normalized cross correlation (NCC), and nonparametric features including Rank, SoftRank, and Census [20]. Recently, a state-of-the-art stereo matching cost, ANCC [2], was proposed to handle the Lambertian condition. Analysis shows that the image color is usually affected by noise coming from illuminant color or image device changes, so [2] proposed a new color mode called the log-chromaticity color mode that is insensitive to radiometric variations. The proposed method works well in radiometric variations. However, ANCC fails in cases of severe illuminant changes, a large brightness difference between the left and right images, or a non-Lambertian reflectance object, and a great deal of time is consumed to establish the color model. Recently, He *et al.* [29] proposed a new edge-aware filter called guided image filter. Guilder filter is different from bilateral filter where its runtime is linear in the number of image pixels. Motivated by the benefit of guided filter, Hosni *et al.* developed a stereo matching cost by filtering the cost volume with guided image filter [3]. Existing local patterns encode the local region based on differences in intensities between two adjacent pixels. Therefore, mis-encoding easily occurs when noise appears in the local region. Motivated by the limitation of existing local patterns, Nguyen *et al.* [4] introduced a new local pattern, called local density encoding (LDE), to encode the local region based on pixel similarity with the help of the $(n-1)$th order derivative. LDE is then applied to develop a robust stereo matching cost under different brightness condition. Recently, Kasun *et al.* [30] proposed extreme learning machine auto-encoder (ELM-AE) to represents input features based on singular value for digit classification application. ELM-AE doesn't use fine-tuning to update the neuron parameters. The neuron parameters of ELM-AE can be calculated analytically. ELM-AE is different from our proposed deep learning approach where ELM-AE learns to represent input feature via single values, while our proposed method learns the actual representation of input data. In addition, neuron parameters of our proposed method are randomly generated, and then updated iteratively, using optimization method, a limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) [23], while neuron parameters of ELM-AE are randomly generated and then fixed. More recently, Zbontar and Lecun proposed a supervised deep learning model-based convolution neural network for stereo matching (CNN)[8]. Zbontar and Lecun use a convolutional neural network (CNN) with eight layers to train how well left image and right image patches match together where disparity ground truth is known. Their supervised approach uses stochastic gradient descent to minimize the cross-entropy loss with given true disparity to compute the training error. The output of their proposed network is then used to compute the stereo data cost between a pair of patches. The data cost is then applied to their modified SGM to find a disparity map. In addition, post-processing methods (left-right check, median filter, bilateral filter, interpolation, and subpixel enhancement) are also introduced to refine their final disparity map. However, their proposed system requires the disparity ground truth in advance for training their network (with eight layers), whereas our proposed solution uses an un-supervised approach that does not require ground truth for training (with three layers).

After investigating existing matching costs for stereo matching algorithms, we realized that almost matching costs are designed to operate in specific conditions using expert knowledge. Many researchers have sought to solve the current problems surrounding stereo matching costs by developing a new transformation function that can extract more information from the local region. However, their proposed transformations were developed using observations from limited conditions. Developing of a robust transformation function is challenging because many factors exist and must be accounted for in real conditions, such as snow, rain, shadows, and clouds. This limitation motivated us to develop a robust transformation function that automatically learns and extracts information from the local region under realistic conditions. In our proposed method, we first collect a large data set of various driving conditions. Our system then automatically learns and extracts a robust feature transformation function from that dataset.

III. Proposed Stereo Matching Cost

In this research, we proposed a robust stereo matching cost based on deep learning strategy, also called as deep learning pattern (DLP), based on two motivations: (1) existing stereo matching algorithms are designed under specific conditions using expert-knowledge. However, those data costs cannot work well under conditions that vary from those for which they were designed. (2) The desire to produce a deep learning approach with the ability to automatically learn and discover the structure features (or abstract features) from raw input without using much expert knowledge and that less dependent on specific design conditions. This section first briefly reviews an unsupervised algorithm stack auto-encoder [21], and describes how it can learn abstract feature from the input. Second, based on the benefits of an auto-encoder, the proposed deep learning-based data cost is then introduced.

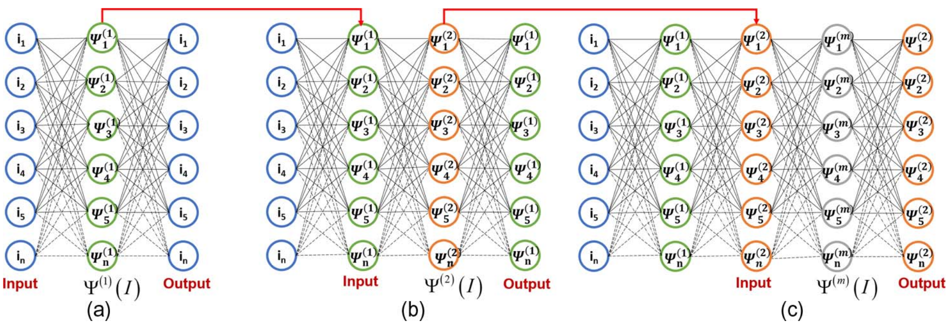


Fig. 2. (a) Autoencoder architecture. (b) Stack autoencoder with two hidden layers. (c) Stack autoencoder with m hidden layers.

A. *Stack Auto-Encoder*

An auto-encoder is an unsupervised algorithm that tries to learn an approximation of a density function by setting the input values equal to the output value [21]. Fig. 2(a) shows the architecture of a simple auto-encoder neural network including a hidden layer, $\Psi^{(1)}(I)$, input layer, and output layer. $I = \{i_1, i_2, \ldots, i_n\}$ is the number of inputs in the training network. The auto-encoder network tries to learn a function, $\Psi^{(1)}(I) \approx I$, which is computed as follows:

$$\Psi^{(1)}(I) = \frac{1}{1 + e^{-\left(W^{(1)} * I + b^{(1)}\right)}} \tag{1}$$

where $W^{(1)}$ and $b^{(1)}$ are the weight matrix and bias vector, respectively, computed by minimizing the cost function Δ_{sparse} with KL divergence [22].

A stacked auto-encoder is constructed by combining several auto-encoder layers in which the output of each layer is the input for the next layer, as shown in Fig. 2(b). $\Psi^{(1)}(I)$ becomes an input for computing the next auto-encoder, $\Psi^{(2)}(I)$

$$\Psi^{(2)}(I) = \frac{1}{1 + e^{-\left(W^{(2)} * \Psi^{(1)}(I) + b^{(2)}\right)}}. \tag{2}$$

Without loss of generality, a stack auto-encoder [21] with m layers can be constructed as shown in Fig. 2(c)

$$\Psi^{(m)}(I) = \frac{1}{1 + e^{-\left(W^{(m)} * \Psi^{(m-1)}(I) + b^{(m)}\right)}}. \tag{3}$$

B. *Proposed Stereo Matching Cost Using a Deep Learning Strategy*

Existing local patterns, such as Census/Rank and their variants, are designed to extract the structure of the local region under a specific condition. Therefore, those transformations will fail to handle conditions that differ from their original design. Motivated by the benefits of an auto-encoder and mindful of the limitations of existing hand-design feature-based stereo matching algorithms, we propose a robust stereo matching cost using a deep learning strategy.

The proposed system first design an auto-encoder network to learn the robust transformation function $\Psi^{(m}(I)$ to provide stable features under various conditions as shown in Algorithm 1. The proposed method used only an unlabeled dataset to learn and discover meaningful structure from the raw input image.

Algorithm 1: Learning $\Psi^{(m)}(I)$ using auto-encoder

```
input : unlabelled dataset $\left\{ d_{unlabelled}{}^{(n)} \right\}_{n=1}^{N}$, sparsity parameter $\rho$,
        weight of sparsity penalty term $\beta$, weight decay parameter $\lambda$ number
        of input units $I$, number of hidden units $H$, and number of output
        units $O$
output: $\Psi^{(m)}(I)$: [$W$ and $b$] of hidden units
begin
    Define a number of $P$ patches for batch training and the terminated
    condition $\varepsilon$.
    Initialize weights $W^{(0)}_{hid}(H, I)$, $b^{(0)}_{hid}(H)$, $W^{(0)}_{out}(O, H)$, $b^{(0)}_{out}(O)$
      $W = [W^{(0)}_{hid}, W^{(0)}_{out}]$
      $b = [b^{(0)}_{hid}, b^{(0)}_{out}]$
    for $n \leftarrow 1$ to $N/P$ do
        Select P sub-patches from N
        Minimizes the cost function
        $f = \min_{W,b} \left( \frac{1}{P} \sum_{i=1}^{P} \left( \frac{1}{2} \left\| h_{W,b} \left( x^{(i)} - y^{(i)} \right) \right\|^2 \right) + \frac{\lambda}{2} \sum_{l=1}^{n_l - 1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_l + 1} \left( W^{(l)}_{ji} \right)^2 + \beta \sum_{j=1}^{s_2} \left( \rho \log \frac{\rho}{\widehat{\rho}_j} + (1 - \rho) \log \frac{1-\rho}{1-\widehat{\rho}_j} \right) \right)$
        using L-BFGS optimization.
        Update $W$ and $b$
        if $f \leq \varepsilon$ then
            Compute the cost function $f$ using on N patches
            $f = \min_{W,b} \left( \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{2} \left\| h_{W,b} \left( x^{(i)} - y^{(i)} \right) \right\|^2 \right) + \frac{\lambda}{2} \sum_{l=1}^{n_l - 1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_l + 1} \left( W^{(l)}_{ji} \right)^2 + \beta \sum_{j=1}^{s_2} \left( \rho \log \frac{\rho}{\widehat{\rho}_j} + (1 - \rho) \log \frac{1-\rho}{1-\widehat{\rho}_j} \right) \right)$
            if $f \leq \varepsilon$ then
                Stop training and return $W^{(n)}_{hid}$ and $b^{(n)}_{hid}$
            end if
            ;
        end if
        ;
    end for
end
```

Thus, if we can provide a diverse dataset that includes challenging driving conditions including rain, sun, snow, and so forth, the proposed method can automatically learn an effective transformation function under these challenging conditions. This paper set the input, hidden and output units to be 9 × 9 gray image patches, as shown in Fig. 3. The number of hidden layers was set to 81 $(9 \times 9 = 81)$ units to maintain the original input structure. To help the proposed method quickly obtain the convergence, we used [28] to initialize W and b rather than using a normally random method. The training process was performed off-line to automatically learn the robust transformation $\Psi^{(m}(I)$. It is worth noting that $\Psi^{(m}(I)$ is an independent transformation that can be applied to compute various data costs. This paper introduces a new stereo matching cost based on the output of the proposed off-line network (transformation

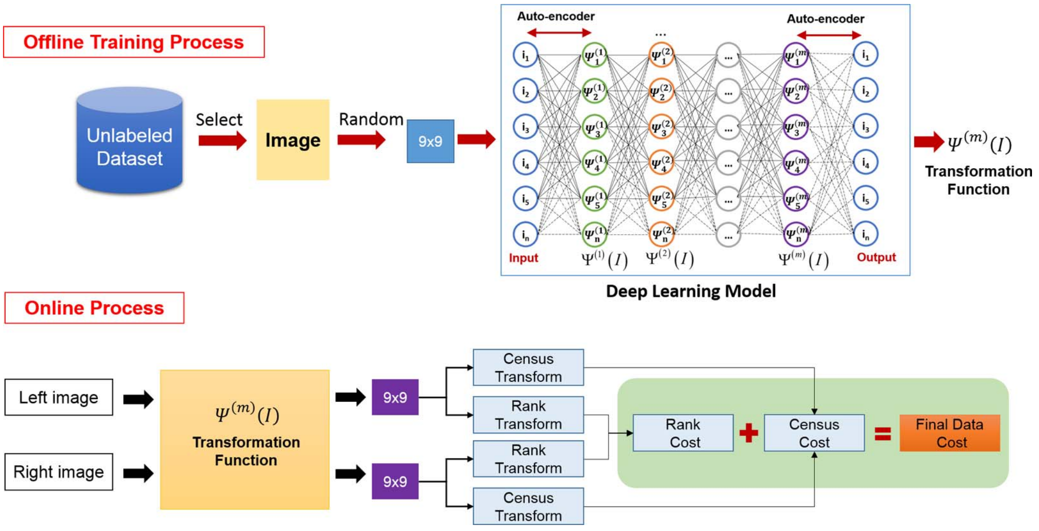


Fig. 3. Workflow of the offline and online processes of the proposed method.

function $\Psi^{(m}(I))$. Thus, the proposed network tries to learn an approximation to the identity function. The identity function seems a particularly trivial function to try to learn. However, by defining constraints on the neural network, such as by maintaining, increasing, or decreasing the number of hidden units, we can discover interesting structure about the input data. For example, if we limit the number of hidden units, the network is forced to learn a compressed representation of the input. However, if the inputs are completely random, where each input comes from an independent and identically distributed (IID) Gaussian distribution of the other features, then the compression task would be very difficult. However, if there is structure in the data, for example, if some of the input features are correlated, then an auto-encoder algorithm will be able to automatically discover some of those correlations.

From the output of off-line training (using the transformation function $\Psi^{(m}(I))$, we then introduced an adaptive Census and Rank to compute stereo data cost. For a pixel $p = (x, y)$, the proposed stereo data cost is computed by fusing the advantages of Rank and Census as follows:

$$
\begin{aligned}
D_{\mathrm{DLP}}(x,y,d) &= \alpha \frac{D_{\mathrm{Rank}}(x,y,d)}{T} + \beta D_{\mathrm{Census}}(x,y,d)\\
D_{\mathrm{Census}}(x,y,d) &= H\begin{bmatrix} f_{\mathrm{census}}\left(\Psi_L^{(m)}(x,y)\right)\\ f_{\mathrm{census}}\left(\Psi_R^{(m)}(x-d,y)\right)\end{bmatrix}\\
D_{\mathrm{Rank}}(x,y,d) &= \left|\begin{matrix} f_{\mathrm{rank}}\left(\Psi_L^{(m)}(x,y)\right)\\ -f_{\mathrm{rank}}\left(\Psi_R^{(m)}(x-d,y)\right)\end{matrix}\right|
\end{aligned} \tag{4}
$$

where H is the Hamming distance, and $\Psi_L^{(m)}(x,y)$ and $\Psi_R^{(m)}(x-d,y)$ are the results of the transformation functions of the left and right images, respectively

$$
\begin{aligned}
\Psi_L^{(m)}(x,y) &= \frac{1}{1+e^{-\left(W^{(m)}\times\Psi_L^{(m)}(x,y)+b^{(m)}\right)}}\\
\Psi_R^{(m)}(x-d,y) &= \frac{1}{1+e^{-\left(W^{(m)}\times\Psi_R^{(m)}(x-d,y)+b^{(m)}\right)}}
\end{aligned} \tag{5}
$$

where W and b are parameters of the transformation function $\Psi^{(m}(I))$ that is learned as in Algorithm 1. The Census and Rank transform is then computed as follows:

$$
\begin{aligned}
f_{\mathrm{census}}(x_0) &= \otimes_{p=1}^{P} f_{\mathrm{sig}}(x_0, x_p)\\
f_{\mathrm{sig}}(x_0, x_p) &= \begin{cases} 0, & x_0 > x_p\\ 1, & x_0 \leq x_p\end{cases}\\
f_{\mathrm{rank}}(x_0) &= \sum_{p=1}^{P} f_{\mathrm{sig}}(x_0, x_p)
\end{aligned} \tag{6}
$$

where $\otimes$ represents the bitwise concatenation. However, it is challenging to select good α and β parameter for the proposed method. Fortunately, motivated by the work of Nguyen *et al.* [4], the two parameters α and β are automatically estimated by measuring the intensity difference between the left and right images as follows:

$$
\begin{aligned}
&\begin{cases} \alpha = 0.1 & \text{if } |\rho_L - \rho_R| > \Phi\\ \alpha = 0.9 & \text{if } |\rho_L - \rho_R| < \Gamma\\ \alpha = 0.5 & \text{if } \Gamma \leq |\rho_L - \rho_R| \leq \Phi\end{cases}\\
&\rho_L = \frac{1}{N}\sum_{W_p}\frac{1}{(W_p-1)}\sum_{q\in W_p, q\neq p}|I_L(Z_p) - I_L(Z_q)|\\
&\rho_R = \frac{1}{N}\sum_{W_p}\frac{1}{(W_p-1)}\sum_{q\in W_p, q\neq p}|I_R(Z_p) - I_R(Z_q)|
\end{aligned} \tag{7}
$$

where $\beta = 1 - \alpha$, ρ_L, and ρ_R are used to measure the intensity difference between the left and right image, respectively.

Thus, the proposed method is more robust than existing data costs for two reasons: (1) a robust transformation function $\Psi^{(m}(I)$ is introduced to extract more abstract features from the raw input data. The transformation function was automatically learned under various driving conditions. (2) An adaptive Rank and Census is then introduced to compute stereo data cost based on these abstract features. Census captures a relative ordering and encodes the local spatial structure of the local region, and can distinguish between rotations and reflections, while Rank increases robustness to outliers near depth-discontinuities and

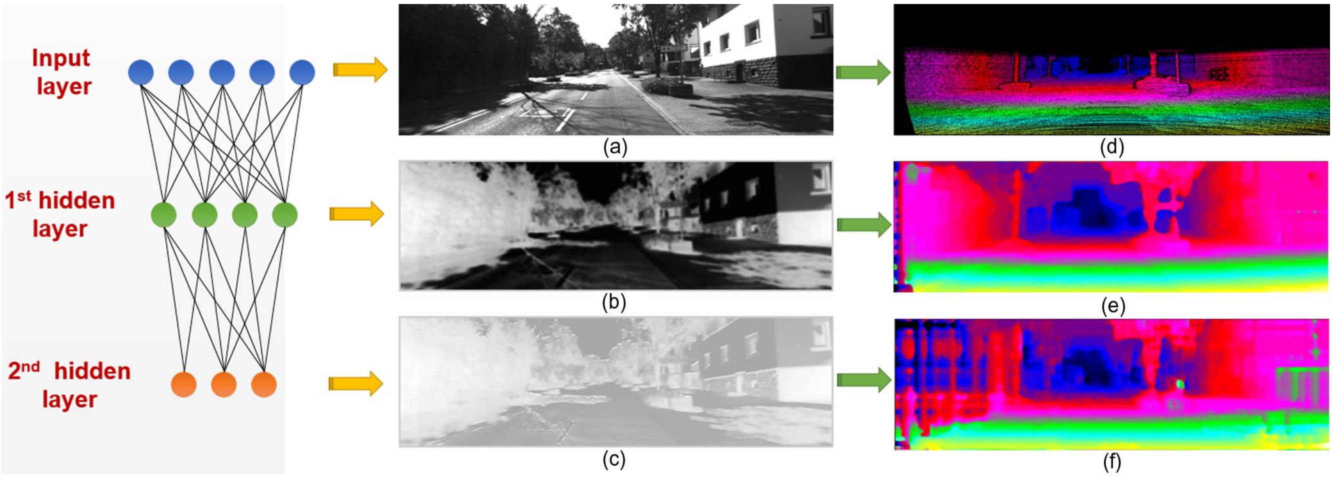


Fig. 4. (a) Left image in the KITTI dataset. (b) Transformation and (e) disparity results from the first hidden layer of the proposed method. (c) Transformation and (f) disparity results from the second hidden layer of the proposed method. (d) Ground truth image.

requires less memory to store a result. Thus, adaptive Rank and Census provide more robust stereo data cost based abstract features. Fig. 4 shows the transformation results and corresponding disparities in each hidden layer of the proposed deep learning method. In the first hidden layer, the proposed deep learning approach has the best performance because the edge information from the local region is effectively extracted. The performance of the proposed method decreases slightly in the second layer because the detailed information present in the first layer is lost in order to produce more abstract information. The results from the first hidden layer are adequate to improve the performance of the existing stereo matching algorithms. Therefore, this paper focuses on investigating the benefits of only the first layer for stereo matching; the benefits of higher layers will be fully considered in the future.

C. Training Parameters

The KITTI dataset provides 194 pairs of stereo images with available ground truth for training and 195 pairs of stereo images with no ground truth for testing. In this research, we used only 388 training images, 194 left images and 194 right images, from the KITTI training dataset. The proposed deep learning model is trained by randomly selecting 2,000 9 × 9 gray image patches for each input image. An auto-encoder is then used to learn a nonlinear function to reconstruct the input data. We used a total of 776,000 9 × 9 patches for training. The number of hidden layers is set to 81 ($9 \times 9 = 81$) units to maintain the original input structure. A Limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) [23] is used to optimize the training cost function with a sparsity parameter of 0.01, lambda of 0.00001, and beta of 3. We designed a deep neural network with three layers including input and output layers. The trained parameter is then used to evaluate the performance of the proposed method with various datasets, such as KITTI, Middlebury, and HCI. It is worth noting that the parameter is trained only on the training images of the KITTI dataset because we want to evaluate how the parameters trained under one dataset work with other datasets.

IV. Experimental Results and Discussion

A. Dataset for Evaluation

Three stereo datasets are used to evaluate the performance of the proposed deep learning-based stereo data cost including: KITTI [9], HCI [11],and Middlebury [10]. KITTI is a standard outdoor dataset that provides a sequence of stereo image pairs and their corresponding spatial ground truth under normal driving conditions. The HCI dataset provides a large number of stereo image pairs under various driving conditions, such as sun flares, shadows on trucks, crossing car, blinking arrow, and so forth. However, HCI does not provide disparity ground truth for its stereo image pairs. Middlebury provides stereo image pairs with dense ground truth under controlled indoor conditions.

B. Stereo Method for Comparison

Two strategies are generally used to evaluate the performance of stereo matching cost: local and semi-global approaches. A local stereo matching algorithm is often used with the winner-takes-all strategy (WTA). The stereo data cost is computed within a given window size. The final disparity is then picked up using WTA. For a semi-global approach, we selected SGM because it satisfies both the accuracy and processing time required for on-road applications. Hirschmuller [19] proposed an SGM in which cost aggregation is performed as the approximation of a global energy function in 1D with 8 (or 16) directions. SGM has the same memory requirement as BP. The cost function, $E_r(p, f_p)$, that assigns disparity, f_p, to pixel p along direction r is defined as follows:

$$E_r(p, f_p) = D_p(f_p) + \min \left\{ \begin{matrix} E_r(p-r, f_p) \\ E_r(p-r, f_p-1) + C_1 \\ E_r(p-r, f_p+1) + C_1 \\ \min\limits_i E_r(p-r, i) + C_2 \end{matrix} \right\} - \min_k E_r(p-r, k) \quad (8)$$

where the first term, $D_p(f_p)$, is the data cost that assigns the disparity f_p to pixel p. C_1 and C_2 are two penalty constants of the continuity term. Thus, SGM performance is improved when the robust data cost is accurately modeled under challenging conditions.

C. Stereo Data Cost for Comparison

Hirschmuller *et al.* evaluated the performance of various stereo data costs with radiometric differences [20]. In their experimental results, Census obtained the best performance under illumination and exposure changes in the Middlebury dataset. Therefore, we conducted experiments to evaluate the performance of our method with Census. In addition, we also evaluated the performance of the proposed method with various dominant data costs, such as guided image filter (GF) [3], principal component analysis (PCA), and convolutional neural network (CNN) [8].

D. Error Evaluation Method and System Configuration

Root mean square (RMS) [24] and percentage of bad pixel matching are often used to estimate the difference between a stereo matching result and the ground truth disparity at time index t

$$\mathrm{RMS}(t) = \left(\frac{1}{P} \sum_{(x,y)} |d_t(x,y) - g_t(x,y)|^2 \right)^{\frac{1}{2}} \tag{9}$$

where P is the total number of pixels in the image, and $d_t(x,y)$ and $g_t(x,y)$ are the intensity values at pixel (x,y) of the computed disparity image and the ground truth disparity image, respectively. In addition, the average root mean square (ARMS) is used to evaluate the error of stereo matching results for long sequences

$$\mathrm{ARMS}_N = \frac{1}{N} \left(\frac{1}{P} \sum_{(x,y)} |d_t(x,y) - g_t(x,y)|^2 \right)^{\frac{1}{2}} \tag{10}$$

where N is the total number of images in the testing sequences.

The percentage of bad pixel matching in the matching result and the ground truth disparity at time index t are often defined as follows:

$$B(t) = \frac{1}{P} \sum_{(x,y)} \left(|d_t(x,y) - g_t(x,y)| > \alpha \right). \tag{11}$$

A PC with an Intel Core I7 processor, 4 GHz, and 8 GB RAM was used to implement the various data costs. Local stereo matching is implemented with a window size of 1 × 1 to evaluate the performance of the proposed method and other comparison methods using a pixel-wise matching strategy (without using a window-aggregation-based method). This research used 388 images from the KITTI training data set with 776,000 9 × 9 patches for training. 9 × 9 patches are used because they performed well on the testing datasets. For a fair comparison, we also implemented Census using a 9 × 9 window. SGM was implemented using four directions, left, right, top and bottom, with $C1 = 45$, and $C2 = 700$, followed by a 15 × 15 median filter to evaluate the performance of the proposed data cost. Two thresholds of the proposed system, Φ and Γ, are set to 3 and 1, respectively.

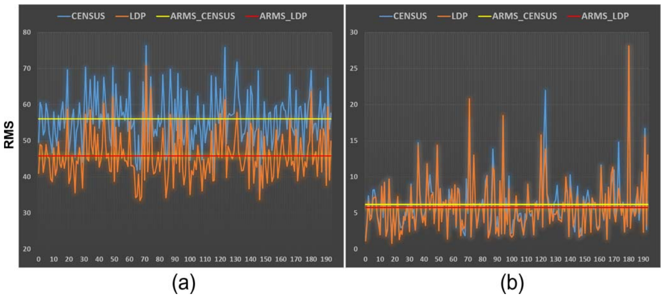


Fig. 5. (a) Experimental results of the proposed and Census data costs using local stereo matching with the KITTI training data set. (b) Experimental results of the proposed and Census data costs using SGM with the KITTI training dataset.

E. Results on the KITTI Dataset

To evaluate the performance of the proposed data cost with a local stereo matching approach, we first used the proposed method and Census costs to compute the data cost. Then, the WTA strategy is applied to produce the final disparity map. Fig. 5(a) shows a comprehensive test of 194 image pairs in the KITTI training dataset using a local stereo matching approach. The average root mean square error of the proposed method (ARMS_DLP) is lower than the average root mean square of Census (ARMS_Census) in this test. Census produces more noise than the proposed method in these tests. The RMS error is reduced by approximately 20% using the proposed method. The performance of Census decreases with large illumination variation changes because the basic assumption of Census, that the gray-level changes monotonically, is violated. In addition, many unknown factors can affect the quality of the input image, such as sun, shadows, and clouds. The proposed data cost produces better results because deep learning can automatically discover stable structures in the input image. Deep learning provides a complex and stable transformation function that can handle the various unknown factors that occur on a real road. Thus, the proposed data cost becomes more robust as more data are used for training.

To evaluate the performance of the proposed data cost using the SGM approach, the proposed method and Census costs are first used to compute the data cost. Then, SGM is performed to produce the final disparity map. The proposed matching cost still produces better results than Census, as shown in Fig. 6. Fig. 5(b) shows the comprehensive experiments with the proposed method and Census using the SGM approach on the KITTI training dataset.

To evaluate the performance of the proposed method with other state-of-the-art methods, we conducted an experiment using 195 pairs of images from the KITTI testing dataset. Table I shows the ranking of the proposed method in comparison with other state-of-the-art stereo methods (when three pixel-error is considered) using an online KITTI benchmark, where out-noc is the percentage of erroneous pixels in non-occluded areas as

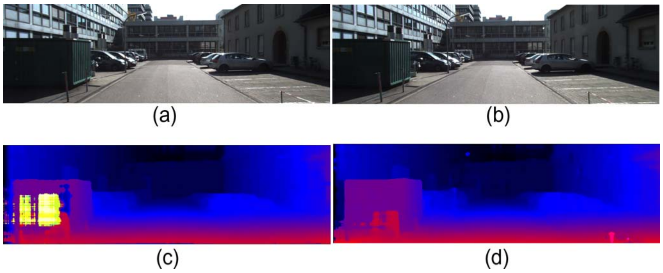


Fig. 6. Experimental results of the proposed data cost and Census using SGM with the KITTI training data set. Pairs of (a) left and (b) right images. Disparity results of (c) Census (RMS = 10.96) and (d) the proposed data cost (RMS = 5.03) on the (a) and (b) image pair.

TABLE I
STEREO EVALUATION USING THE KITTI VISION BENCHMARK SUITE FOR ALL REGIONS IN THE IMAGE

Rank	Method	Out-Noc	Density
2	**MC-CNN**	**2.61** %	100 %
13	StereoSLIC [27]	3.92 %	99.89 %
...	...	...	...
28	RBM	5.18 %	100.00 %
29	ARW	5.20 %	100.00 %
30	**DLP**	**5.28** %	**100.00** %
33	ALTGV	5.36 %	100.00 %
36	CAT	5.57 %	100.00 %
37	**Census**	**5.76** %	**85.80** %
38	TGV2ADC	5.87 %	99.99 %
39	ADCSGM	5.94 %	99.90 %
66	**Guided Filter**	**11.65** %	**100** %
...	...	...	...

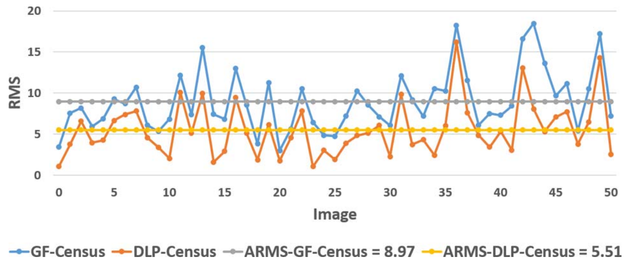


Fig. 7. Root mean square error of the proposed data cost with guided image filtering using the KITTI data set and SGM. ARMS values of the proposed data cost and guided image filtering-based Census are 5.51 and 8.97, respectively.

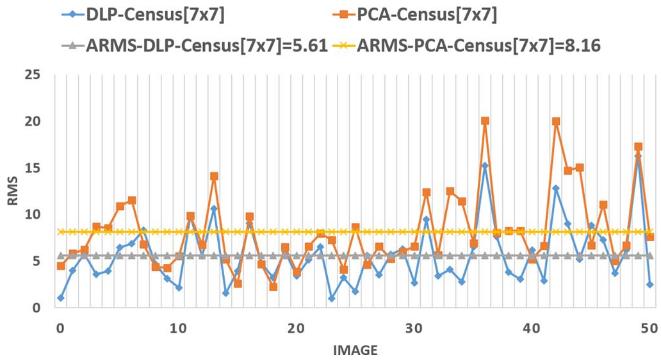


Fig. 8. Root mean square error of the proposed data cost with PCA using the KITTI training data set and SGM. ARMS values of the proposed data cost and PCA are 5.61 and 8.16, respectively.

defined in [9]. The proposed method obtained better results than Census in this test. The KITTI results support our hypothesis that the proposed method outperforms Census in real-road experiments. At the time of this writing, the DLP ranked 30th of 78 submissions, Census ranked 37th. Our current ranking is still low compared top stereo methods, but our proposed system does not investigate advanced post-processing techniques to increase the accuracy of the final disparity map, because we want to evaluate the result of the raw disparity map, directly. More experimental results are available at http://www.cvlibs.net/datasets/kitti.

We now evaluate the performance of the proposed method with guided filtering (GF)[3]. First, simple stereo data cost is initialized using absolute difference (AD). Then, the guided image filter [26] is applied to smooth the stereo data cost at each slide (disparity) in the cost volume. Thus, the proposed guided-filter is considered as a cost aggregation with a given filter kernel size, whereas our proposed DLP is considered a pre-processing step to increase the robustness of the input features. To make a fair comparison, we implemented GF-based stereo matching [3] by initializing the data cost using Census (9 × 9 window size) rather than AD, followed by GF for cost aggregation with a (9 × 9) filter kernel size and epsilon of 0.1. GF-based stereo matching contains a cost aggregation with a filtering mechanism. Therefore, to make a sensible comparison, we conducted an experiment to evaluate the performance of the proposed method with a guided image filtering-based method [3] using semi-global stereo matching rather than a local approach. We used a guided filter-based Census to initialize the data cost for SGM. Fig. 7 shows a comprehensive experiment on the sequence images of the KITTI training dataset. The proposed method obtained better performance than GF in terms of RMS. To further evaluate the performance of the proposed DLP with a similar un-supervised method, principal component analysis (PCA) was used. PCA can be used to reduce the number of input features using principal components. The principal component can be less than or equal to the input features. To evaluate the performance of DLP and PCA, we built an autoencoder network to reduce the input feature from 9 × 9 to 7 × 7, and PCA performed the same task. Census transformation (7 × 7) is applied after executing DLP and PCA to compute the data cost. Fig. 8 shows the performance of the proposed method and PCA on the KITTI training dataset. The proposed method obtained better results than PCA in this test.

We used the KITTI dataset to evaluate the performance of the proposed method with the most related work CNN [8]. CNN [8] achieves ranks 2nd with the KITTI dataset. However, the processing time of their system is slow (95 s/frame for computing only the convolutional transformation) even though it was implemented on the newest Nvidia GeForce GTX Titan GPU (2880 CUDA cores), whereas the processing time of our proposed system is 152 ms/frame on a normal GPU Geforce GTX 750 (512 CUDA cores). In addition, our proposed system does not investigate advanced post-processing techniques to increase the accuracy of the final disparity map, because we want to evaluate directly the result of the estimated disparity map. Table II shows the performance of our proposed method in comparison to other state-of-the-art methods on the reflective region of the KITTI dataset. The proposed DLP (rank 15th

TABLE II
STEREO EVALUATION USING THE KITTI VISION BENCHMARK SUITE FOR THE REFLECTIVE REGION IN THE IMAGE

Rank	Method	Out-Noc	Avg-All
...	...	...	...
12	StereoSLIC	11.70 %	3.6 px
14	PR-Sf+E	12.42 %	4.0 px
15	**DLP**	**12.50** %	**5.9** px
16	**MC-CNN**	**12.81** %	**4.3 px**
17	PR-Sceneflow	13.21 %	4.0 px
18	TGV2ADC	13.56 %	3.7 px
23	CoR-Conf	14.75 %	5.4 px
...	...	...	...
39	**Census**	**17.30** %	**9.8** px
40	iSGM	17.73 %	8.6 px
...	...	...	...
58	**Guided filter**	**23.99** %	**12.9** px
...	...	...	...

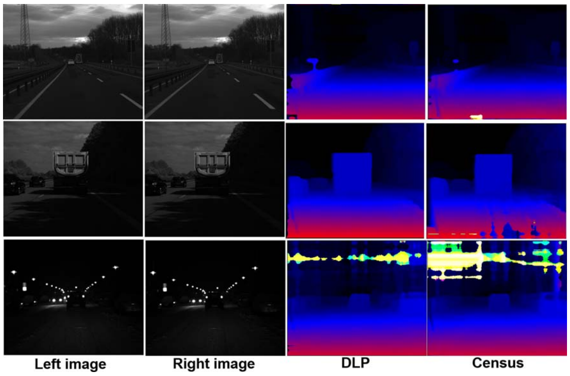


Fig. 9. Experimental results of the proposed data cost and Census using SGM with the HCI dataset. First row is the blinking arrow condition. Second row is a shadow on truck conditions. Third row is a night and snow condition.

of 78 submissions) obtained much better results than Census (rank 39th) and guided image filtering (rank 58th) [3]. The performance of the proposed method (rank 15th) is better than that of the CNN-based method [8] (rank 16th) on the reflective regions (when 5 pixel-error is considered). Thus, the performance of MC-CNN was decreased in reflective regions (rank 2nd in normal conditions, and rank 16th in reflective regions), whereas the proposed DLP obtained stable results (rank 30th in normal conditions, and rank 15th in reflective regions).

F. Results From HCI Dataset

To evaluate the performance of the proposed method with other outdoor datasets, we conducted experiments with the HCI dataset. HCI provides a comprehensive dataset captured under different weather and lighting conditions. Fig. 9 shows the performance of the proposed data cost and Census using SGM approach. The proposed method demonstrates a much better performance than Census by visualizing of the final disparity map in these tests.

To further validate the performance of the proposed method with GF under more challenging conditions, we conducted additional experiments under sunny conditions using the HCI dataset, as shown in Fig. 10. The proposed method outperformed GF-based Census. It is interesting to note that the proposed method, by fusing benefits of DLP, Census, and GF, obtained the best performance, as shown in Fig. 10(d). Thus, the performance of the GF-based Census cost was improved by using DLP as a pre-processing step to provide robust features and GF as a post-processing step to smooth the cost volume. We plan to investigate the benefits by fusing DLP and GF in the future. The transformation function trained only on the KITTI dataset also operates well with the HCI dataset. This occurs since the proposed approach can automatically learn and discover stable features/structure from the raw input image in KITTI dataset, and these learned features also exists in the HCI dataset. Thus, if we can provide diverse datasets for our training model, a range of possible driving conditions can be considered, and a more robust transformation function can be obtained.

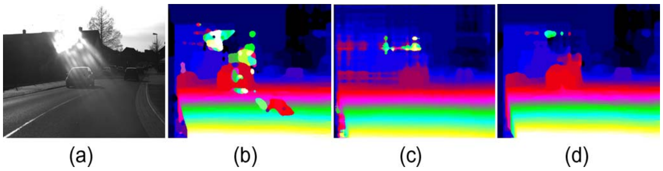
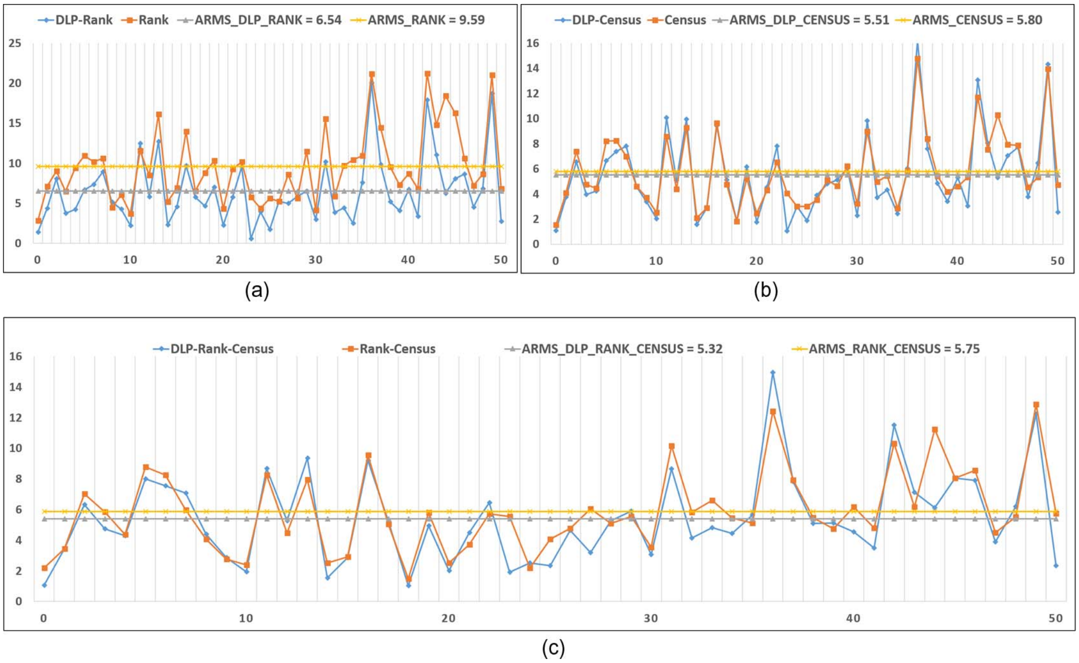


Fig. 10. Experimental results of the proposed data cost and guided image filtering using SGM with the HCI dataset. (a) Image captured under different illuminations. (b) and (c) Disparity results of the guided filter-based census cost and the proposed method DLP-Census, respectively, on (a). (d) Disparity result obtained by fusing the benefits of DLP, Census, and Guided filtering (DLP+Census+GF) on (a).

G. Performance Evaluation of the Proposed Method With Various Transformations Using KITTI and Middlebury Datasets

To independently evaluate the performance of the proposed method, we compare deep learning-based Rank-Census cost (**DLP-Rank-Census**), Census cost (**DLP-Census**) and Rank cost (**DLP-Rank**), as defined in equations (4) and (8), respectively, using the KITTI training data set with available ground truth. Fig. 11(a)–(c) show the performance of the proposed DLP-Rank vs. Rank, DLP-Census vs. Census, and DLP-Rank-Census vs. Rank-Census with SGM, respectively. In the first experiment [Fig. 11(a)], the proposed DLP-Rank-based cost $(\text{ARMS} = 6.54)$ significantly improves the performance of the Rank-based cost $(\text{ARMS} = 9.59)$. In the second experiment [Fig. 11(b)], the proposed DLP-Census-based cost $(\text{ARMS} = 5.51)$ improves the performance of the Census-based cost $(\text{ARMS} = 5.80)$. In the third experiment [Fig. 11(c)], the proposed DLP-Rank-Census-based cost $(\text{ARMS} = 5.32)$ improves the performance of the Rank-Census-based cost $(\text{ARMS} = 5.75)$. Thus, in general, the proposed DLP-based data cost improves the performance of Rank, Census, and their combination.

To further validate the performance of the proposed data cost with the Middlebury dataset, six stereo image pairs are selected. Fig. 12 shows the performance of DLP-Rank-Census, DLP-Census and DLP-Rank compared with the original Census-Rank, Census, and Rank using six pairs of images with different illuminations in the Middlebury dataset. DLP-Rank obtained better results than Rank in terms of RMS error in all six testing cases. However, for bad pixel rate, Rank obtained a better result than DLP-Rank in three cases ([I1E0-I1E2],[I1E0-I2E2], [I1E0-I3E2]). For RMS error, DLP-Census obtained a better

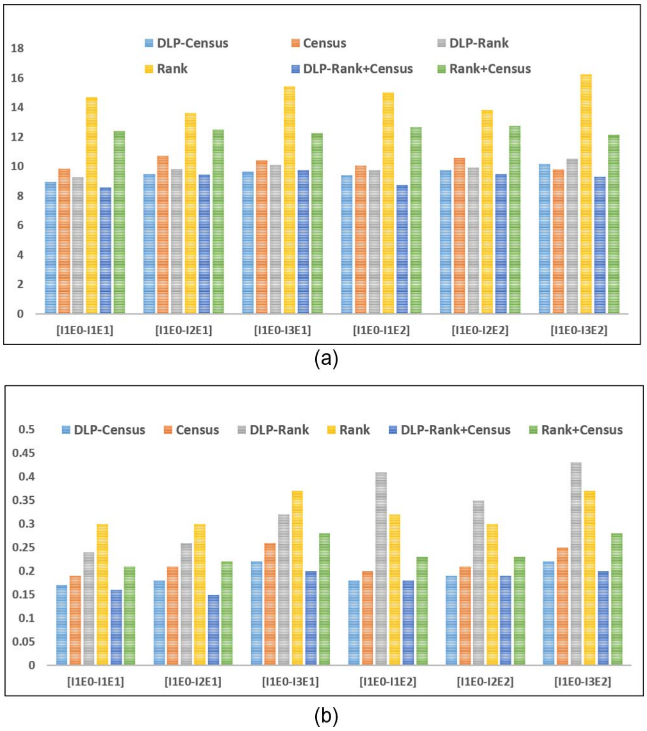


(a)

(b)

(c)

Fig. 11. Experimental results of DLP-Rank-Census, DLP-Census, DLP-Rank, Rank-Census, Census, and Rank using the SGM with the KITTI dataset.



(a)

(b)

Fig. 12. Experimental results of the proposed data costs with Census on bowling images from the Middlebury dataset under different illumination (I) and exposure (E) conditions using SGM. (a) Root mean square error of the compared methods: (ARMS-DLP-Census = 9.58), (ARMS-Census = 10.24), (ARMS-Rank = 14.81), (ARMS-DLP-Rank = 9.9), (ARMS-Rank-Census = 12.46), and (ARMS-DLP-Rank-Census = 9.55). (b) Percentage of the bad pixel matching for the compared methods using ABPP: (ABPP-Census = 0.22), (ABPP-DLP-Census = 0.19), (ABPP-Rank = 0.34), (ABPP-DLP-Rank = 0.33), (ABPP-Rank-Census = 0.24), and (ABPP-DLP-Rank-Census = 0.18).

result than Census in five testing cases; however, for the last example [I1E0-E3E2], Census's RMS error was slightly lower than that of DLP-Census. For the combination of Rank and Census cost, the proposed DLP-Rank-Census performed better than Census-Rank in terms of RMS error and percentage of bad pixel matches in all test cases. In general, DLP-Rank-Census (ARMS = 9.55, ABBP = 0.18) performed better than Rank-Census (ARMS = 12.46, ABBP = 0.24), DLP-Rank (ARMS = 9.9, ABBP = 0.33) performed better than Rank (ARMS = 14.81, ABBP = 0.34), and DLP-Census (ARMS = 9.58, ABBP = 0.19) performed better than Census (ARMS = 10.24, ABBP = 0.22) when average root mean square (ARMS) and average bad pixel percentage (ABPP) are considered.

H. Performance Discussion of the Proposed Method With Multi-Layers

We have not yet discussed the benefits of DLP in multiple layer networks. In this paper, we intended to maintain the size of the input features. Therefore, the number of hidden units is set equal to the number of input units. In all experiments, by maintaining the feature size, we found that the performance of the proposed model is best in the first hidden layer, where it can be used to improve the performance of existing stereo methods. On the other hand, the proposed deep learning model can discover more interesting and meaningful structures of the input data when the number of hidden units is not equal to the number of input layers, as discussed in the recent work of Le *et al.* [6]. Therefore, in future work, we will design a deep network with descending units $(9\times9) \Rightarrow (7\times7) \Rightarrow (5\times5) \Rightarrow (3\times3)$ or increasing units $(9\times9) \Rightarrow (11\times11) \Rightarrow (13\times13) \Rightarrow (15\times15)$, in order to more completely understand the benefits of the deep learning model in a stereo matching field. In addition, there are limitations of the proposed method. The proposed method tries to automatically learn a robust transformation function (or identity function) using a large unlabeled dataset. However, it still needs a mechanism to slightly correct

TABLE III
TRAINING TIME ON CPU AND GPU OF THE PROPOSED METHOD

Total images	Image resolution	CPU	GPU
388	1226x370	∼10 hours	∼ 2.5 hours

the transformation function using a supervised fine-tuning approach with a small labeled dataset. (This is the other reason why the proposed system obtained better result than Census, GF, and PCA, but it still ranks 30th in the KITTI benchmark). For example, Le *et al.* built an object classification system based on an auto-encoder solution [6]. Their proposed system demonstrated a state-of-the-art performance by using the benefits of unsupervised and supervised learning approaches. Moreover, it is worth noting that Zbontar and Lecun's supervised model achieved state-of-the-art performance with the KITTI dataset [8]. Therefore, we plan to investigate the benefit of unsupervised and supervised models to increase the accuracy of the proposed system. First, we plan to use a large unlabeled dataset to automatically train and learn a robust transformation function using deep layer network. Second, a supervised approach will be applied to correct the transformation function using a small label data-set (if we first allow children to learn and discover everything they want, then slightly teach them what is correct, and after which they continue to learn independently based on this knowledge).

I. Fast Processing Time Using GPU Implementation

We evaluated the performance of the proposed system in term of training time and processing time (runtime) using both CPU and GPU. For 388 KITTI training dataset images, the training time of the proposed system in PC and GPU is described in Table III. The processing time (runtime) for computing a transformation of the proposed method is approximately 13 seconds on a CPU. Most of the computation is due to from matrix multiplication that is easy to parallel using a GPU. Therefore, in this research, we also designed and implemented the proposed transformation on a GPU GeForce GTX 750 architecture to accelerate its processing time. First, given the KITTI image size (1226 × 370), we implemented the first kernel to extract 9 × 9 pixels at each location of the image. The result of this step is matrix Ψ_L with a size of 1226 × 370 × 81. A second kernel was then implemented to calculate the matrix multiplication between Ψ_L and the weight matrix W (81 × 81) using the efficient sparse matrix multiplication techniques [25]. For the image sizes in KITTI (1226 × 370) and Middlebury (417 × 370), the total processing time is approximately 75.88 milliseconds/frame and 25.45 milliseconds/frame, respectively. Thus, the total processing time for the transformation of left and right images is approximately 152 milliseconds/frame for the KITTI image, and 51 millisecond/frame for the Middlebury image. Fig. 13 shows the processing time (runtime) of the proposed method with various image and input sizes. The processing time of the proposed method is suitable for real-time applications with the Middlebury image size (417 × 370) and is near real-time with the KITTI image size (1226 × 370). We plan to investigate a more advanced GPU architecture to improve the processing time of the proposed method with high resolution images.

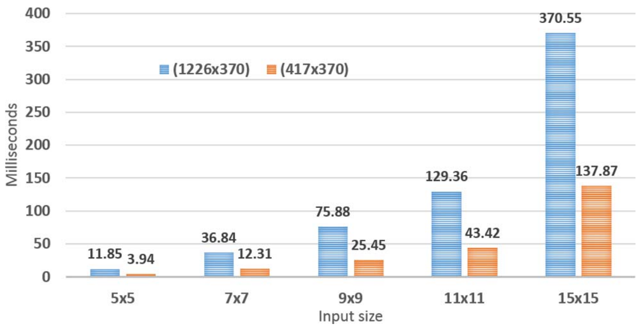


Fig. 13. Processing time of the proposed method with various image resolutions ([1226 × 370] and [417 × 370]) and window sizes using GeForce GTX 750 GPU. Execution time is measured on a single image.

V. CONCLUSION

Most existing local transformations for stereo matching are designed heuristically using expert knowledge. Therefore, they require a lot of time to investigate and design a suitable feature for specific conditions. In this paper, we investigated deep learning to develop a robust stereo data cost that improves the performance of existing stereo matching without requiring much expert knowledge. We hope that this research will provide new insight regarding the benefits of deep learning-based unsupervised approaches for stereo matching. The accuracy of a stereo matching algorithm depends on how much data can be mined in the training process rather than on designing a feature for specific conditions. This paper investigates only a simple deep learning model for the stereo matching field. The proposed deep learning model significantly improves the performance of existing stereo matching algorithms and outperforms the state-of-the-art stereo data costs of Census and guided filtering. We plan to perform comprehensive research to fully elucidate the benefits of the deep learning model by using unsupervised and supervised approaches and investigating a more advanced GPU (or FPGA) architecture to improve the performance of the proposed method.

REFERENCES

[1] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. Eur. Conf. Comput. Vis.*, 1994, vol. 801, pp. 151–158.

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

[3] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother, "Real-time local stereo maching using guided image filtering," *Int. Conf. Multimedia Expo*, pp. 1–6, 2011.

[4] V. D. Nguyen, D. D. Nguyen, S. J. Lee, and J. W. Jeon, "Local density encoding for robust stereo matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 12, pp. 2049–2062, Dec. 2014.

[5] Y. Bengio, "Learning deep architectures for AI," *J. Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[6] Q. V. Le *et al.*, "Building high-level features using large scale unsupervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1–11.

[7] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[8] J. Zbontar and Y. L. Cun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. CVPR*, 2015, p. 1.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[10] Middlebury Vision Evaluation, 2010. [Online]. Available: http://vision.middlebury.edu/stereo/

[11] S. Meister, B. Jahne, and D. Kondermann, "Outdoor stereo camera system for the generation of real-world benchmark data sets," *Opt. Eng.*, vol. 51, no. 2, pp.1–7, Mar. 2012.

[12] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[13] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, Oct. 2006.

[14] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2003, vol. 1, pp. 556–561.

[15] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE Conf. Comput. Vis.n Pattern Recognit.*, 2011, vol. 1, pp. 3017–3024.

[16] Q. Yang, "A non-local cost aggregation method for stereo matching, 2012," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1402–1409.

[17] Z. Liu and R. Klette, "Dynamic programming stereo on real-world sequences," in *Proc. ICONIP*, 2008, vol. 5506, pp. 527–534.

[18] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[19] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[20] H. Hirschmuller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1582–1599, Sep. 2009.

[21] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 153–160.

[22] G. Hinton, "A practical guide to training restricted Boltzmann machine," Tech. Rep., Univ. Toronto, Toronto, ON, Canada, 2010.

[23] Q. V. Le *et al.* , "On optimization methods for deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[24] S. Morales, T. Vaudrey, and R. Klette, "Robustness evaluation of stereo algorithms on long stereo sequences," in *Proc. IEEE Intell. Veh.*, 2009, pp. 347–352.

[25] N. Bell and M. Garland, "Efficient sparse matrix-vector multiplication on CUDA," NVIDIA, Santa Clara, CA, USA, Technical Report NVR-2008-004, Dec. 2008.

[26] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. ECCV*, 2010, pp. 1–14.

[27] K. Yamaguchi, D. McAllester, and R. Urtasun, "Robust monocular epipolar flow estimation," in *Proc. CVPR*, 2013, pp. 1–8.

[28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," presented at the International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 2010.

[29] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. PAMI*, 2013, pp. 1397–1409.

[30] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Oct. 2013.

Vinh Dinh Nguyen received the B.S. degree in computer science (*magna cum laude*) from Nong Lam University, Ho Chi Minh City, Vietnam, in 2007 and the M.S. and Ph.D. degrees in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2012 and 2015, respectively.

Since 2015, he has been with Sungkyunkwan University as a Researcher. His research interests include computer vision, image processing, and graphics processing unit computing.

Hau Van Nguyen received the B.S. degree in electronic physics from University of Science, Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam, in 2006. He is currently working toward the Ph.D. degree in the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea.

His research interests include computer vision, image processing, and graphics processing unit computing.

Jae Wook Jeon (S'82–M'84) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1990.

From 1990 to 1994, he was a Senior Researcher with Samsung Electronics, Suwon, South Korea. Since 1994, he has been with the School of Information and Computer Engineering, Sungkyunkwan University, Suwon, as an Assistant Professor, where he is currently a Professor. His research interests include robotics, embedded systems, and factory automation.