Multimed Tools Appl
DOI 10.1007/s11042-016-3932-y

Detecting ground control points via convolutional neural network for stereo matching

Zhun Zhong[1] · Songzhi Su[1] · Donglin Cao[1] · Shaozi Li[1] · Zhihan Lv[2]

Received: 21 May 2016 / Revised: 26 July 2016 / Accepted: 1 September 2016

Abstract In this paper, we present a novel approach to detect ground control points (GCPs) for stereo matching problem. First of all, we train a convolutional neural network (CNN) on a large stereo set, and compute the matching confidence of each pixel by using the trained CNN model. Secondly, we present a ground control points selection scheme according to the maximum matching confidence of each pixel. Finally, the selected GCPs are used to refine the matching costs, then we apply the new matching costs to perform optimization with semi-global matching algorithm for improving the final disparity maps. We evaluate our approach on the KITTI 2012 stereo benchmark dataset. Our experiments show that the proposed approach significantly improves the accuracy of disparity maps.

Keywords Stereo matching · CNN · Ground control points · Matching confidence

1 Introduction

Stereo matching task has been widely studied in computer vision. The depth information computed by stereo matching algorithm can be used in various vision applications, such as 3D reconstruction, object recognition, object tracking, and autonomous navigation.

✉ Shaozi Li
szlig@xmu.edu.cn

Zhun Zhong
zhunzhong@stu.xmu.edu.cn

[1] Cognitive Science Department, Xiamen University, Xiamen, 361005 Fujian, China

[2] SIAT, Chinese Academy of Science, Shenzhen 518055, China

Published online: 22 September 2016

According to [23], the stereo algorithms can be fall into local and global algorithms, on the basis of whether the minimization procedure solving with a global cost function. In a local algorithm, the matching costs between two pixels are computed with a local support window (e.g. 9 × 9), and usually make implicit smoothness assumptions by aggregating pixel-based matching costs. Then, an optimal disparity can be computed based on the aggregated matching costs. In comparison to global algorithms, local algorithms are generally faster, but less accurate since narrow limitation. In contrast, global algorithms make explicit smoothness assumptions and search an optimal disparity by solving an energy based optimization problem. Prevalent global methods include those based on dynamic programming [2], belief propagation [5, 26] and graph cuts [3]. While these global algorithms have achieved impressive results, they usually require substantial computational resources. Other than these two categories, a semi-global block matching stereo algorithm (SGM) [10] was proposed. SGM is also based on a global energy cost function, but it performs optimizations along multiple directions. Compared with global algorithms, SGM has reached a close accuracy with a much lower computational complexity. As a result, a plenty of SGM based modified algorithms have been proposed, and have been successfully applied in the domain of stereo matching [9, 24]. SGM-based methods not only can use stereo pair to compute matching cost, but video pair. There is a very large literature on optical flow, for example see [15–18, 27]. Using a video pair to compute disparity is a special way of structure from motion, the optical flow could be obtained from multiple images. Optical flow has been demonstrated that it is efficient for stereo matching problems [28, 29] by using SGM with video pair.

Although SGM based methods have achieved noticeable results, stereo matching is still unavoidable confronted with the difficulties such as pixel indistinctiveness, depth discontinuities, texture-less regions, and occlusions. These difficulties may cause fail to compute credible matching cost between pixels or support windows. Thus, although matching costs can be computed in a bad case scenario, these matching costs are not reliable enough in any location, which results in the decrease of stereo matching accuracy.

In the past few years, several papers [8, 21, 25] have paid attention to the question whether the computed matching costs are in fact reliable. Haeusler et al. [8] proposed a method to learn a confidence measure from several features, and predicted the confidence by applying the random decision forest to learn a classifier. Similarity, Spyropoulos and Modorhai [25] proposed a learning-based approach to predict confidence, and leveraging the

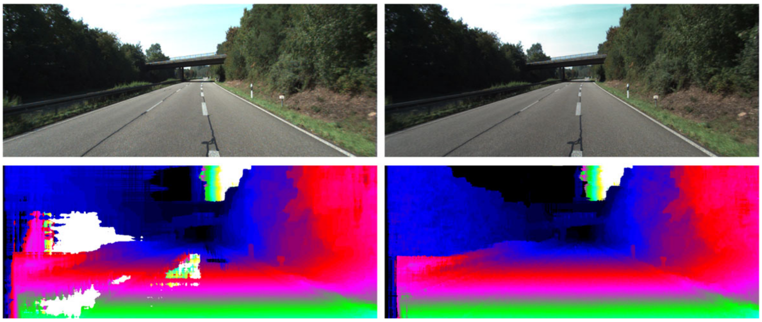

Fig. 1 Example of our algorithm results from KITII 2012 frames. *The top row*: *left* and *right* images; *The bottom row*: disparity map of *left* image computed by SAD matching costs with SGM, and disparity map of *left* image computed by our method

estimated confidence to select pretty reliable pixels as ground control points (GCPs). Park and Yoon [21] selected effective confidence measures via regression forest, and retrained regression forest classifier to predict the confidence of a match, as well as leveraging the predicted confidence to promote the accuracy of disparity maps. All of the above approaches require hand-engineering a set of features for confidence measure, and need training on the basis of specified matching costs, which means we have to train a new prediction classifier while using another matching cost computation algorithm.

To overcome the problems mentioned above, in this paper, we focus on detecting the ground control points (GCPs) based on CNN, and leveraging the detected GCPs to refine the matching costs. Figure 1 shows an example result of our algorithm. Unlike previous methods which used the hand-engineered features for confidence measure, our approach using convolutional neural network to detect GCPs/reliable points without designing the feature of confidence measure. Moreover, we use the detected GCPs to refine the matching costs and computed the final disparity maps with semi-global block matching (SGM). Figure 2 illustrates the overall flow of the proposed algorithm. The contributions of this paper are summarized as follows:

(1) Firstly, we train a convolutional neural network to learn the matching confidence of each pixel on a large set of pairs of stereo patches where the ground truth disparities is available. Then, we detect the ground control points by the maximum confidence of each pixel over all disparities.
(2) Secondly, the stereo matching costs are refined by utilizing the confidence of the detected GCPs. Then the new matching costs are used to compute final disparity maps with semi-global matching algorithm.
(3) The experimental results on the KITTI 2012 stereo benchmark dataset show that our method significantly improves the accuracy of stereo matching overall all images.

The remainder of this paper is organized as follows. We discuss the related work in Section 2. The CNN model for computing matching confidence is given in Section 3. We then describe the algorithm of GCPs detection, and refinement scheme for matching costs

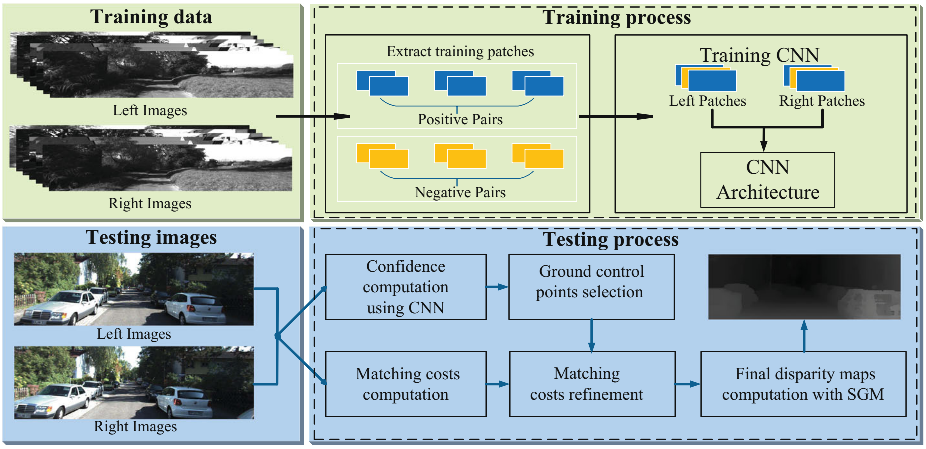

Fig. 2 Overview of the algorithm. The proposed algorithm is mainly divided into five parts: training CNN model, confidence computation by CNN model, ground control points selection, matching costs refinement, and final disparity maps computation

in Section 4. Experimental results and analyses are presented in Section 5. Our conclusion and future work are summarized in Section 6.

2 Related work

In this section, we first briefly review the learning-based methods for predicting the confidence of matching costs. Then, the discussion of several stereo matching costs computation methods based on Convolutional Neural Network.

An early learning-based approach to stereo matching proposed by Lew et al. [14] adopted instance based learning (IBL) to select optimal feature set points for stereo matching. Kong and Tao [12] used nonparametric techniques to train a model to predict the probability of a potential match. The predicted knowledge was then integrated into an MRF framework to improve the depth computation. Later on, Kong and Tao [13] extended their work by learning multiple experts from different matching windows sizes and centers, and the likelihood under each expert was then combined probabilistically into a global MRF framework for improving accuracy. Motten et al. [20] trained a hierarchical classifier for selecting the most promising disparity with the matching costs and spatial relationship of pixels. Peris et al. [22] designed a feature from cost volume, and computed the final stereo disparity using Multiclass Linear Discriminant Analysis (Multiclass LDA).

More recently, in [8, 21, 25], they employed random decision forests to estimate the confidence of the stereo matching costs. Haeusler et al. [8] proposed a method to learn a confidence measure from several features, and predicted the confidence by applying the random decision forest to learn a classifier. Similarity, Spyropoulos and Modorhai [25] used a random decision forests classifier to estimate the confidence of the matching costs, and showed that the confidence information can further be used in a Markov random field framework for improving stereo matching. Park and Yoon [21] measures the confidence by ranking the importance of each match. In addition, they applied the confidence of each match to modify the matching cost, and inserted the new matching costs into stereo method to decrease the error of stereo matching. All of these learning-based approaches have to structure a set features of confidence measures, and train a classifier based on specified matching costs, which indicates that different classifiers should be trained when applying different stereo matching cost computation methods.

Convolutional Neural Networks (CNN) has been rapidly developed in recent years, and has been extensively applied to deal with various computer vision tasks, such as, image retrieval [1, 33], person re-identification [19, 34], and object detection [7, 35]. The most recent year, CNN has been used in stereo matching [4, 30–32] and achieved noticeable results. The destination of these methods is training a CNN with a large set of stereo image patches, and comparing the matching cost between image patches by the trained network. The main difference between them is the architecture of the network. On the Contrast to these methods, we train a CNN to compute matching confidence of each pixel, and detect the GCPs based on the output of CNN.

3 CNN based GCP detection

In this section, we present our CNN model for measuring the matching confidence of each pixel. The training and testing processes of our CNN model are illustrated in Fig. 2.

3.1 Matching confidence

Given a pair of stereo images, left image Φ^L, and right image Φ^R, the matching (similarity) cost at each position $\mathbf{p}$ with different disparities $d \in [0, d_{max}]$ can be compute by:

$$C(p, d) = f(\Phi^L(\mathbf{p}), \Phi^R(\mathbf{p} - d)) \tag{1}$$

where $\Phi^L(\mathbf{p})$ and $\Phi^R(\mathbf{p} - d)$ denote the patches centred at p and $p - d$ in left and right images, respectively. $f(\cdot)$ is the matching cost method.

In a real-world scene application of stereo matching, we may suffer from the difficulties such as depth discontinuities, texture-less regions, and occlusions. These bad scenarios may generate unreliable matching costs, which degrades the overall performance of stereo matching. Thus, if we can estimate the matching confidence that reveal the reliability of matching cost, the stereo matching performance will be promoted with the confidence information. To address this issue, we attempt to use CNN-based model to calculate the matching confidence between patches, where the two patches are centred at the same 3D points having a high matching confidence, and low when they are not.

3.2 Training datasets

In order to compute the matching confidence between patches, the input to the CNN model is a pair of stereo patches, and the output is a inner-product that measures the matching confidence between the inputs.

Hence, a training example is composed of two image patches, one from left image, $\mathbf{p}^L = (x, y)$, and one from right image, $\mathbf{p}_d^R = (x - d, y)$. We sample one negative instance and one positive instance for each pixel in left image, where the true disparity d_t is known.

A negative instance is a small patch collected from right image as follows:

$$\mathbf{p}_d^R = (x - d_t + o_{neg}, y)$$

where o_{neg} is a random value from the interval $[-N_{high}, ..., -N_{low}, N_{low}, ..., N_{high}]$. This random value guarantee that the image patches of negative instances are not centred at the same 3D point.

Similarity, a positive instance is a small patch collected as follows:

$$\mathbf{p}_d^R = (x - d_t + o_{pos}, y)$$

where o_{pos} is a random value from the interval $[-P_{high}, ..., P_{high}]$.

3.3 Architecture

An overview of our architecture is shown in Fig. 3. This architecture is a siamese network. There are two sub-networks in the network that share the same architecture and weights of each layer. The input is a pair of 9 × 9 grayscale image patches, centred at $\mathbf{p}^L = (x, y)$ and $\mathbf{p}_d^R = (x - d, y)$. Each sub-network takes one of the two patches as input, and it is a composition of a number of convolutional layers with rectified linear units (ReLU) layers following all convolutional layers. Since the input of neural network is small patches, we do not adopt any pooling layer. For a pair of patches, we use a fourth layer CNN (as show as the blue layers in Fig. 3) to independently extract feature descriptors $F(\Phi)$ from each sub-network to represent each input patch, and use the inner-product layer to calculate

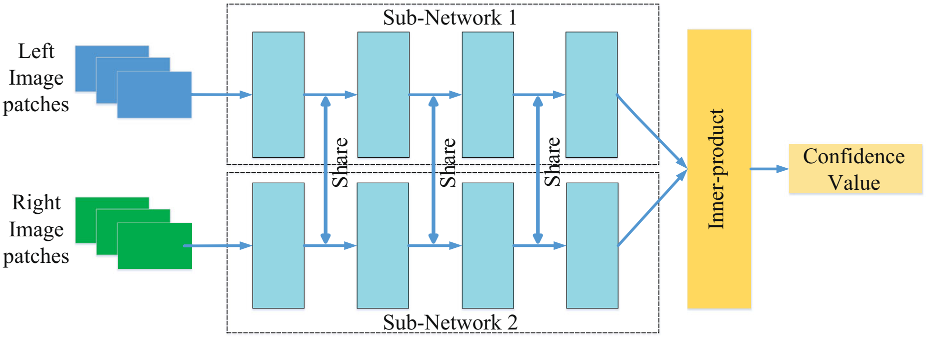

Fig. 3 The Convolutional Neural Network architecture of our training model. Each sub-network contains four convolutional layers followed by ReLu. The confidence value is an inner product of two extracted feature vectors

the matching confidence between two descriptors $F(\Phi^L)$ and $F(\Phi^R)$. The size of each convolution kernel is 3×3, and the number of feature maps in each layer is 64.

We train our model by minimizing a hinge-based loss term which is employed in [32]. The hinge-baed loss term is defined as:

$$\max(0, \epsilon + s_{neg} - s_{pos}) \tag{2}$$

This loss term is computed by considering pairs of training examples centred at the same image position, with one example which $\mathbf{p}^L = (x, y)$ corresponds to $\mathbf{p}_d^R = (x - d, y)$ as positive, and one which not corresponded as negative. s_{pos} denotes the output for the positive instance, s_{neg} denotes the output for the negative instance, and ϵ is a positive real number that we set it to 0.2 in this paper.

In the training process, we input a set of examples over all example pairs, and compute the loss by summing the terms in (2). Thus, we can compute the matching confidence for each pair of patches by using the trained CNN model. We normalize the matching confidence into the interval [0, 1].

4 Improving stereo matching with ground control points

In this section, we propose our ground control points (GCPs) detection approach based on the matching confidence computed by CNN model, and refine the matching costs depending on the confidence of GCPs.

4.1 Detecting ground control points

According to [25], a GCP is a pixel with a high confidence that the computed matching costs are reliable.

Given a pair of stereo images, the pair of stereo patches for input of CNN model are obtained for each pixel and disparity under consideration (e.g. disparity = 128) from left and right image. Then, a matching confidence volume, $Vol(\mathbf{p}, d)$, could be obtained by performing the forward pass on the CNN model.

For each pixel $\mathbf{p}$ in the left image, we can achieve a confidence vector for each disparity (the size of vector is the same as the disparity). The maximum matching confidence, $Cof_c(\mathbf{p})$, is computed by maximizing the matching confidence volume in each disparity:

$$Cof_c(\mathbf{p}) = \max_d Vol(\mathbf{p}, d) \tag{3}$$

The maximum matching confidence indicates the reliability of matching costs for each pixel. We also compute the most confident disparity, $Cof_d(\mathbf{p})$, for each pixel $\mathbf{p}$:

$$Cof_d(\mathbf{p}) = \arg\max_d Vol(\mathbf{p}, d) \tag{4}$$

We propose a simple manner to select GCPs using the maximum matching confidence of each pixel, the selected GCPs can be used to impact neighboring pixels. The selecting criterion is followed by: if $Cof_c(\mathbf{p})$ is larger than a constant threshold θ, pixel $\mathbf{p}$ is a GCP, otherwise not, i.e. unreliable pixels. We define a pixel $\mathbf{p}$ as $\mathbf{p}^+_{GCP}$ if it belongs to GCP, otherwise, define it as $\mathbf{p}^-_{GCP}$.

The main challenge in GCPs selection step is the trade-off between density and accuracy. Considering that too much GCPs are selected, some of them may contain unreliable matching costs that will be propagated to neighboring pixels, which may cause the decreasing of the stereo matching accuracy. Conversely, few GCPs will be not effective enough to improve the matching performance.

Therefore, aiming at selecting reliable GCPs that are benefit to stereo matching, we learn the threshold on the maximum matching confidence $Cof_c(\mathbf{p})$ by cross-validation.

4.2 Refining matching costs with GCPs

In the previous subsection, we described an approach to select GCPs using maximum matching confidence. In this subsection, we present a matching costs refinement approach using the selected GCPs that can be used for the final optimization.

Given a pair of stereo images, the matching costs volume, $C(\mathbf{p}, d)$, can be obtained by a matching cost computation method. In this paper, we use two popular matching cost computation methods, sum of absolute differences (SAD), and census-based Hamming distance (Census). The refinement scheme is a divided into two step. Firstly, based on the observation that $\mathbf{p}^-_{GCP}$ contains unreliable matching costs, we refine the matching costs of $\mathbf{p}^-_{GCP}$ by setting the cost of all disparity to a constant high cost value C^{hi}_{GCP}. As a result, the wrong matching cost could be avoided to pollute other reliable pixels. Secondly, we refine the matching costs of the selected $\mathbf{p}^+_{GCP}$, by setting the matching cost to a constant low cost value C^{low}_{GCP} for the most confident disparity of $\mathbf{p}^+_{GCP}$. The most confident disparity of $\mathbf{p}^+_{GCP}$ is $Cof_d(\mathbf{p})$, that was computed in the last subsection. The matching costs of all the other disparities for $\mathbf{p}^+_{GCP}$ are unmodified. In this way, the matching costs are refined. We define the new matching costs volume as $\hat{C}^{re}(\mathbf{p}, d)$. Therefore, the influence of unreliable pixels are greatly suppressed, and be reigned by more confident disparities. In this way, we can reduce the disturbance from other unreliable disparities in the subsequent optimization process. Moreover, it is worth noting that the proposed matching cost refinement scheme can be used for any matching cost computation algorithms.

In order to demonstrate our approach is robust to various matching cost computation algorithms, we compute two different cost volumes, using SAD and Census, respectively.

The SAD matching costs are computed as follows:

$$C_{SAD}(\mathbf{p}, d) = \sum_{\mathbf{q}\in\Psi(\mathbf{p})} \left|\Phi^L(\mathbf{q}) - \Phi^R(\mathbf{q}d)\right| \tag{5}$$

where $\Phi^L(\mathbf{q})$ and $\Phi^R(\mathbf{q}d)$ are intensities of pixel in left and right image. And $\Psi(\mathbf{p})$ is a set of pixels in a fixed local window centred at $\mathbf{p}$. The coordinate of $\mathbf{p} = (x, y)$, and $\mathbf{q}d = (x - d, y)$.

And the Census matching costs are defined as follows:

$$C_{Census}(\mathbf{p}, d) = \sum_{\mathbf{q}\in\Psi(\mathbf{p})} XOR(W^L(\mathbf{p}, \mathbf{q}), W^R(\mathbf{p}d, \mathbf{q}d)) \tag{6}$$

where the coordinate of pixel $\mathbf{p} = (x, y)$, $\mathbf{q} = (x, y)$, $\mathbf{p}d = (x - d, y)$, and pixel $\mathbf{q}d = (x - d, y)$. And $W(\mathbf{p}, \mathbf{q})$ is a binary function. $W(\mathbf{p}, \mathbf{q})$ returns 1, while the intensity of $\mathbf{p}$ is larger than $\mathbf{q}$, otherwise zero. $\Psi(\mathbf{p})$ is pixels in a fixed local window centred at $\mathbf{p}$..

4.3 New matching cost with SGM

We compute the disparity map by applying the refined cost volume $\hat{C}^{re}(\mathbf{p}, d)$ to the semi-global matching (SGM) [10] algorithm. Following Hirschmuller [10], the SGM considers the stereo matching as an energy function that minimizes:

$$\begin{aligned} E(D) &= \sum_{\mathbf{p}}(\hat{C}^{re}(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q}\in\Psi(\mathbf{p})} P_1 \cdot \left[\left|D_{\mathbf{p}} - D_{\mathbf{q}}\right| = 1\right] \\ &\quad + \sum_{\mathbf{q}\in\Psi(p)} P_2 \cdot \left[\left|D_{\mathbf{p}} - D_{\mathbf{q}}\right| > 1\right]) \end{aligned} \tag{7}$$

The first item is the sum the refined matching costs of all pixels, so that penalizes disparities $D_{\mathbf{p}}$ with high matching costs. The second item penalty all pixels q having small disparity differences in the neighborhood $\Psi_{\mathbf{p}}$ of $\mathbf{p}$. The last item penalty all pixels q having disparity differences larger than 1 in the neighborhood $\Psi_{\mathbf{p}}$ of $\mathbf{p}$. The minimization of the (7) is an NP-hard problem, instead of performing minimizing $E(D)$ in all directions simultaneously, we perform the minimization in a single direction, and repeat for 16 directions. The cost $M_r(\mathbf{p}, d)$ along multiple paths are defined as:

$$\begin{aligned} M_r(\mathbf{p}, d) &= \hat{C}^{re}(\mathbf{p}, d) + \min(M_r(\mathbf{p} - r, d)), \\ &\qquad M_r(\mathbf{p} - r, d - 1) + \mathbf{p}_1, \\ &\qquad M_r(\mathbf{p} - r, d + 1) + \mathbf{p}_1, \\ &\qquad \min_k M_r(\mathbf{p} - r, k) + \mathbf{p}_2) \end{aligned} \tag{8}$$

The final disparity costs could be obtained by averaging the costs along 16 directions:

$$M(\mathbf{p}, d) = \frac{1}{16}\sum_r M_r(\mathbf{p}, d) \tag{9}$$

Finally, the disparity map $D(\mathbf{p})$ is computed by the Winner-Takes-All (WTA) strategy as follows:

$$D(\mathbf{p}) = \arg\min_d M(\mathbf{p}, d), \quad d \in [0, d_{max}] \tag{10}$$

where d_{max} is the maximum value of possible disparities.

5 Experiments

In this Section, we evaluate our proposed method on the KITTI 2012 stereo benchmark dataset, and compare with the state-of-the-art confidence prediction method (Lev) [21]. We use Caffe [11] to train our CNN model, and the stereo method is implemented in CUDA, with a Nvidia GeForce GTX Titan GPU.

5.1 Data set

The KITTI 2012 stereo benchmark dataset [6] is composed of 194 training and 195 testing stereo images. This benchmark dataset aims to estimate the true disparities for all pixels on the left image. The ground truth disparities for the training images are public provided for researchers, and the one for testing images are withheld, researchers can submit the result to the website for evaluation.

5.2 Details of training CNN model

At training time, we set the mini-batch to 128 pairs of patches. We use a learning rate of 0.001 for the first 2,000k iterations, and decrease the learning rate to 0.0001 for the next 1,000k iterations. We set the momentum to 0.9, and the weight decay to 0.0005. The whole process of training takes about 18 hours.

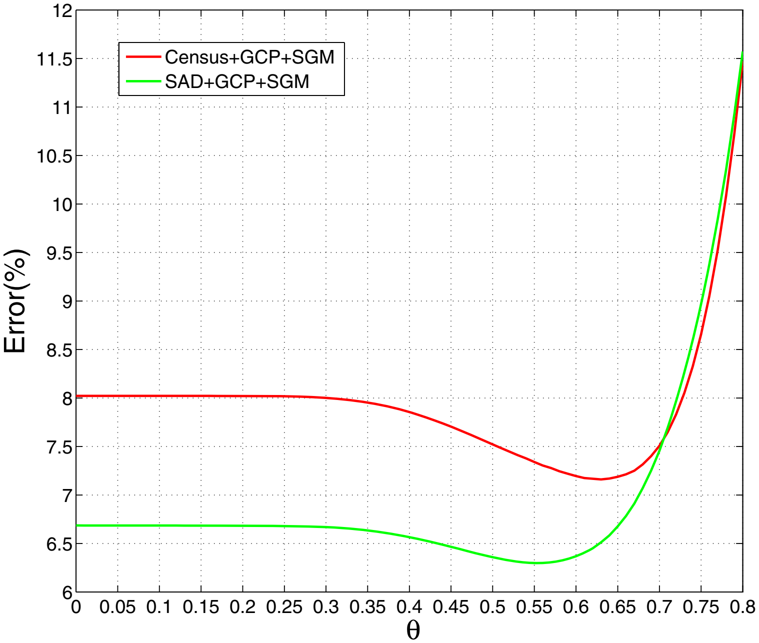

Fig. 4 Evaluate the performance under different confidence parameter settings of θ. We fix the parameters $\{C_{GCP}^{hi} = 200, C_{GCP}^{low} = 1.3\}$ for Census, and $\{C_{GCP}^{hi} = 5, C_{GCP}^{low} = 0.001\}$ for SAD, respectively

Table 1 Error rates of disparity maps with SGM apply different matching cost methods

Method	Error
Census+SGM	10.46 %
SAD+SGM	12.04 %
Census+GCP+SGM	7.19 %
SAD+GCP+SGM	6.29 %

Our method (+GCP) gives lower error rates than that of without GCP refining

5.3 Parameter settings

In this paper, we set the support window size to 9 × 9 for both SAD and Census. For the penalty items in the SGM, we set $\{P_1 = 1, P_2 = 14\}$ for SAD, and $\{P_1 = 4, P_2 = 128\}$ for Census, respectively. The training parameters of CNN are set to $\{N_{high} = 8, N_{low} = 4, P_{high} = 1\}$. We fix the parameters $\{C_{GCP}^{hi} = 200, C_{GCP}^{low} = 1.3\}$ for Census, and $\left\{C_{GCP}^{hi} = 5, C_{GCP}^{low} = 0.001\right\}$ for SAD, respectively. The matching costs are ranged in [0, 80] for Census, and [0, 3.2] for SAD,respectively.

5.4 Confidence parameter analysis

We first evaluate the performance under different confidence parameter settings of θ. Results evaluated on KITTI training dataset are summarized in Fig. 4. It can be seen that the best result is obtained nearby $\theta = 0.60$. Setting too large value of θ would not achieve improving results even bring about decrease sharply. Setting too large value of θ may consider a major of pixels as unreliable points, result in losing the information of reliable pixels, which does not help to improve the matching accuracy even harm the matching performance. A too low value may consider most of the pixels to GCPs, this may only use the information of CNN matching confidence, and fail to avoid the influence of unreliable pixels. We set $\theta = 0.55$ for SAD, and $\theta = 0.60$ for Census in all following experiments.

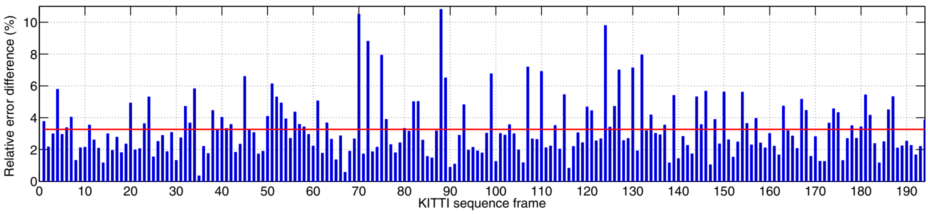

Fig. 5 Overall improvement compared Census+GCP+SGM with Census+SGM algorithm. For each frame, we estimate the accuracy improvement (*blue column*) of our proposed algorithm using Census matching costs. The *red line* indicates average improvement overall all frames. Our algorithm outperforms the Census+SGM algorithm in all frames

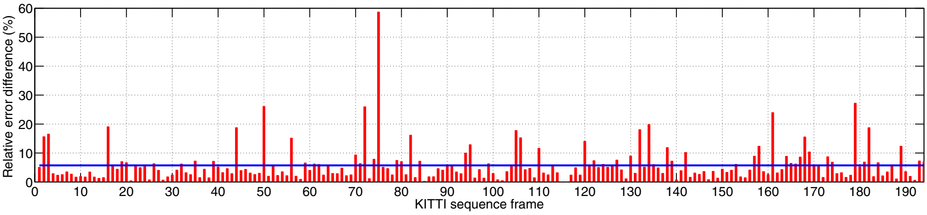

Fig. 6 Overall improvement compared SAD+GCP+SGM with SAD+SGM algorithm. For each frame, we estimate the accuracy improvement (*red column*) of our proposed algorithm using SAD matching costs. The *blue line* indicates average improvement overall all frames. Our algorithm outperforms the SAD+SGM algorithm in all frames

5.5 Stereo accuracy

We first estimate the accuracy improvement of our approach while applying different matching costs computation methods, SAD and Census. The results evaluated on KITTI training dataset are presented in Table 1. The proposed method using the new matching costs that are refined by GCPs significantly improves the accuracy of disparity maps under different matching cost computation methods. The average error rate of the method which uses the original Census matching costs (Census+SGM) is 10.46 %, while the modified method (Census+GCP+SGM) which uses refined matching costs reduces the error rate by 3.26 %. Similarly the error rate of SAD+GCP+SGM is significantly reduced by 5.75 %. The overall improvement is detailed in Figs. 5 and 6. Obviously, our proposed approach consistently improves the accuracy of stereo matching over all images. In addition, our approach is robust to different matching cost computation methods. Figure 7 contains some examples for the disparity maps produced by our method.

Secondly, we compare our approach to the state-of-the-art confidence prediction method [21]. The results evaluated on KITTI training dataset are shown in Table 2. Our proposed method achieves an error rate of 7.19 % which is lower than that of the lev [21] by 2.19 %. It demonstrates that our method obtains more reliable GCPs, and makes full use of the confidence information of GCPs for improving the accuracy of stereo matching. Specifically, our proposed method could directly be applied to any other matching costs computation methods, while Lev [21] requires to train a new predictor for another matching costs. Note that, in order to observably reflect the improvement of our method, all the results in our experiments are computed without any post-processing algorithms.

We also compare the prediction time of our approach to Lev [21]. Lev requires 2.2s to predict the confidence per image, while our approach only takes 0.3s, our approach is more efficient.

Table 2 Error rates of disparity maps comparison. The initial matching costs are computed by Census

Method	Census+SGM	Our	Lev [21]
Error	10.46 %	7.19 %	9.38 %

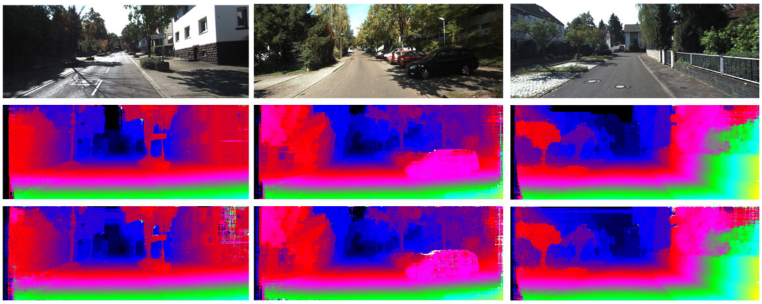

Fig. 7 Examples of disparity maps on the KITTI 2012 data set. *From top to bottom: left* images; disparity maps of SAD+GCP+SGM; and disparity maps of Census+GCP+SGM

6 Conclusion

This paper presents a Convolutional Neural Network based approach that is able to detect the ground control points (GCPs) according to the matching confidence of each pixel. We first learn a Convolutional Neural Network to estimate the confidence of each pixel. Then we select GCPs of image depending on the confidence. In addition, we present a robust approach to obtain a new matching costs by refining the matching costs with the GCPs confidence, which can further be used to compute the final disparity maps. Experiments on KITTI 2012 stereo dataset demonstrate that our approach significantly improves the accuracy of stereo matching on overall images, and our approach achieves an impressive result that surpasses the current leading learning-based method. Furthermore, our proposed method can be applied in various matching cost computation methods. In the future work, it is a worth direction while applying optimal flow to our prediction model.

Acknowledgments We thank Wenjing Li for helpful discussions and encouragement. This work is supported by the Nature Science Foundation of China (No.61202143, No.61572409), the Natural Science Foundation of Fujian Province (No.2013J05100) and Fujian Provi-nce 2011 Collaborative Innovation Center of TCM Health Management.

References

1. Arandjelović R, Gronat P, Torii A, Pajdla T, Sivic J (2015) NetVLAD: CNN architecture for weakly supervised place recognition. arXiv:1511.07247
2. Bobick AF, Intille SS (1999) Large occlusion stereo. IJCV 33(3):181–200
3. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. TPAMI 23(11):1222–1239
4. Chen Z, Sun X, Wang L, Yu Y, Huang C (2015) A deep visual correspondence embedding model for stereo matching costs. In: ICCV, pp 972–980
5. Freeman WT, Pasztor EC, Carmichael OT (2000) Learning low-level vision. IJCV 40(1):25–47
6. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: the kitti dataset. Int J Robot Res:0278364913491297
7. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision (pp. 1440–1448)
8. Haeusler R, Nair R, Kondermann D (2013) Ensemble learning for confidence measures in stereo vision. In: CVPR. IEEE, pp 305–312

9. Hermann S, Klette R (2013) Iterative semi-global matching for robust driver assistance systems. In: ACCV. Springer, pp 465–478
10. Hirschmüller H (2008) Stereo processing by semiglobal matching and mutual information. TPAMI 30(2):328–341
11. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia. ACM, pp 675–678
12. Kong D, Tao H (2004) A method for learning matching errors for stereo computation. In: BMVC, vol 1, p 2
13. Kong D, Tao H (2006) Stereo matching via learning multiple experts behaviors. In: BMVC, vol 1, p 2
14. Lew MS, Huang TS, Wong K (1994) Learning and feature selection in stereo matching. TPAMI 16(9):869–881
15. Li W, Chen Y, Lee J, Ren G, Cosker D (2016) Blur robust optical flow using motion channel. arXiv:1603.02253
16. Li W, Cosker D (2016) Video interpolation using optical flow and laplacian smoothness. Neurocomputing
17. Li W, Cosker D, Brown M, Tang R (2013) Optical flow estimation using laplacian mesh energy. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2435–2442
18. Li W, Cosker D, Zhihan L, Brown M (2016) Nonrigid optical flow ground truth for real-world scenes with time-varying shading effects. IEEE Robotics and Automation Letters 2(1):231–238
19. Liang Z, Zhi B, Yifan S, Jingdong W, Shengjin W, Chi S, Qi T (2016) Mars: a video benchmark for large-scale person re-identification. In: European conference on computer vision. Springer
20. Motten A, Claesen L, Pan Y (2012) Trinocular disparity processor using a hierarchic classification structure. In: IEEE/IFIP 20th international conference on VLSI And system-on-chip (VLSI-SoC), 2012. IEEE, pp 247–250
21. Park MG, Yoon KJ (2015) Leveraging stereo matching with learning-based confidence measures. In: CVPR, pp 101–109
22. Peris M, Maki A, Martull S, Ohkawa Y, Fukui K (2012) Towards a simulation driven stereo vision system. In: ICPR. IEEE, pp 1038–1042
23. Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV 47(1-3):7–42
24. Spangenberg R, Langner T, Rojas R (2013) Weighted semi-global matching and center-symmetric census transform for robust driver assistance. In: Computer analysis of images and patterns. Springer, pp 34–41
25. Spyropoulos A, Komodakis N, Mordohai P (2014) Learning to detect ground control points for improving the accuracy of stereo matching. In: CVPR. IEEE, pp 1621–1628
26. Sun J, Zheng NN, Shum HY (2003) Stereo matching using belief propagation. TPAMI 25(7):787–800
27. Vedula S, Baker S, Rander P, Collins R, Kanade T (1999) Three-dimensional scene flow. In: The proceedings of the seventh IEEE international conference on computer vision, 1999, vol 2. IEEE, pp 722–729
28. Yamaguchi K, McAllester D, Urtasun R (2013) Robust monocular epipolar flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1862–1869
29. Yamaguchi K, McAllester D, Urtasun R (2014) Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: European conference on computer vision. Springer, pp 756–771
30. Zagoruyko S, Komodakis N (2015) Learning to compare image patches via convolutional neural networks. CVPR
31. Žbontar J, LeCun Y (2015) Computing the stereo matching cost with a convolutional neural network. CVPR
32. Zbontar J, LeCun Y (2016) Stereo matching by training a convolutional neural network to compare image patches. J Mach Learn Res 17:1–32
33. Zheng L, Wang S, Tian L, He F, Liu Z, Tian Q (2015) Query-adaptive late fusion for image search and person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1741–1750
34. Zheng L, Zhang H, Sun S et al (2016) Person re-identification in the wild. arXiv:1604.02531
35. Zhong Z, Lei M, Li S, Fan J (2016) Re-ranking object proposals for object detection in automatic driving. arXiv:1605.05904

Zhun Zhong received the M.S. Degree in Computer Science and Technology in 2015 from China University Of Petroleum, Qingdao, China. He is currently working towards his Ph.D at Xiamen University. His research interests include machine learning, object detection, and stereo vision.

Songzhi Su received the B.S. degree in Computer Science and Technology from Shandong University, China, in 2005. He received M.S. and Ph.D degree in Computer Science in 2008 and 2011, both from Xiamen University, Fujian, China. He joined the faculty of Xiamen University as an assistant professor in 2011. His research interests include pedestrian detection, object detection and recognition, RGBD based human action recognition and image/video retrieval.

Donglin Cao is assistant professor of natural language processing at Department of Cognitive Science, Xiamen University in China. He graduated in Computer Science department of Xiamen University, China, in 2000. He earned his master degree in Computer Science at Xiamen University, China, in 2003. He received his PhD degree in Computer Software and Theory from Institute of Computing Technology, Chinese Academy of Sciences, in 2009. His research interests are about Web information retrieval, Large-scale Content Mining, Natural Language Processing and Image Automatic Annotation. In his related research directions, he has published more than 10 papers in domestic and international journals and conferences.

Shaozi Li received the B.S. degree from the Computer Science Department, Hunan University in 1983, and the M.S. degree from the Institute of System Engineering, Xi'an Jiaotong University in 1988, and the Ph.D. degree from the College of Computer Science, National University of Defense Technology in 2009. He currently serves as the Professor and Chair of School of Information Science and Technology of Xiamen University, the Vice Director of Fujian Key Lab of the Brain-like Intelligence System, and the Vice Director and General Secretary concurrently of the Council of Fujian Artificial Intelligence Society. His research interests cover Arti- ficial Intelligence and Its Applications, Moving Objects Detection and Recognition, Machine Learning, Computer Vision, Natural Language Processing and Multimedia Information Retrieval, Network Multimedia and CSCW Technology and others.

Zhihan Lv is an engineer and researcher of virtual/augmented reality and multimedia major in mathematics and computer science, having plenty of work experience on virtual reality and augmented reality projects, engage in application of computer visualization and computer vision. His research application fields widely range from everyday life to traditional research fields (i.e. geography, biology, medicine). During the past years, he has completed several projects successfully on PC, Website, Smartphone and Smartglasses.