CONVOLUTIONAL NEURAL NETWORK USING MULTI-SCALE INFORMATION FOR STEREO MATCHING COST COMPUTATION

Jiahui Chen[1] *Chun Yuan*[1]

[1]Graduate School at Shenzhen, Tsinghua University

ABSTRACT

Computing matching cost by Convolutional neural networks(CNNs) work well in fetching accurate dense disparity maps. But these methods still have problems: (1) they always employ equal weights for left and right images in convolutional layers, losing relational information of patches; (2) they don't solve the balance between patches' size and processing efficiency, the larger size the more information but slower. The proposed multi-scale CNN structured method fetches contexts by employing down-sampled images, this increases the matching accuracy without enlarging the input patch. A multi-scale cross-based aggregation algorithm is proposed to further refine the performances. Proposed method achieves challenging performance against state-of-the-art methods with an error rate of 4.77% in KITTI non-occluded section.

Index Terms— pattern matching, stereo vision, convolutional neural network

1. INTRODUCTION

The disparity is utilized to represent the depth as the former is reciprocal with the latter. The stereo matching algorithms often follow the four steps: cost computation, cost aggregation, disparity computation and disparity refinement [1]. Cost computation is exploited to compute matching costs for each pixel in all possible disparity. After the computation, the costs are aggregated from supported regions of each pixel. Then, disparity map is calculated from cost and refined by various refinement algorithms. The various existing algorithms are generally in accordance with these steps, or designed some transition steps between them. The innovations of the proposed method are in the first two steps.

Up to now, a large number of algorithms employ learning algorithms to solve stereo matching problem. Most of them focus on optimizing the stereo indirectly, such as getting pixel features [6], predicting the confidence of matching result [7], setting relationship of super-pixel [11] or just learning object 3D models [15]. Recently, Zbontar et al. exploit CNNs to compute dense matching cost got from learning [5]. Although CNN based methods achieve efficient performance on stereo matching, they usually use a fix patch and equal weights for corresponding patches, losing dependencies between patches and not considering about balance between accuracy and efficiency. Choosing small patches makes better efficiency but it will lose surrounding features, causing bad performance in texture-less or edge-less regions. On the other side, choosing large patches will increase the computational cost but fetch context, achieving high accuracy in texture-less regions.

Many researchers have found that using multi-scale images can increase matching accuracy in texture-less regions without losing much efficiency, such methods as Cross-Scale Cost Aggregation [3] and multi-blocks processing [14]. But these traditional algorithms are usually hard to achieve the same accuracy of learning methods, especially in complicated regions. Deep learning methods fetching multi-scale information in other areas are proved to be effective, for example, Spatial Pyramid Pooling(SPP) net [2] and recurrent network [4].

The proposed algorithm employs multi-scale information through CNN to compute stereo matching cost. the main contributions of proposed algorithm are as follows: (1) a method employing CNN and image pyramid is proposed fetching contexts to improve the matching accuracy in texture-less regions, this employs surrounding pixels by fetching down-sampled images which increases the matching accuracy without enlarging the input patch. (2) dependencies between patches are taken advantage as input instead of single patch to improve the initial matching performance. (3) a multi-scale cross-based aggregation algorithm is employed as aggregation step to fetch gradual deflating support regions for accurate refining. Experimental results on KITTI benchmark [17] demonstrate the effectiveness of our method, which lead an error rate of 4.77% in KITTI non-occluded regions.

2. ARCHITECTURE

Typical stereo algorithms always begin from computing the stereo matching cost $C(p,d)$ for each pixel p in each disparity d. For pixel $p(x,y)$ in left image, its matching cost can be represented by patches surrounding $p(x,y)$ in left image and $p_d(x-d,y)$ in right image. Zbontar et al. provide the method of computing stereo matching cost with CNN [5] achieves accurate dense disparity maps and its results lead the current benchmark.

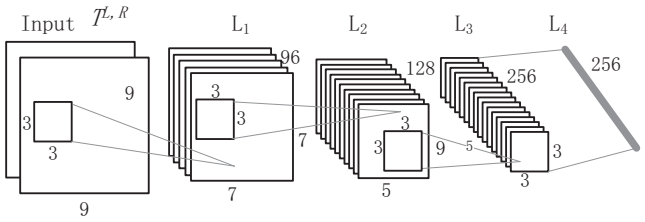

Figure 1. Convolution sub-network architecture

But existing CNN based methods lose the dependencies between corresponding patches and the contexts surrounding patches, this would make the results not accurate enough at texture-less regions. Following their methods, we design an architecture to calculate matching cost by CNN, whose methods of fetching the dependencies and contexts to improve accuracy separately exhibited in section 2.1 and 2.2.

2.1. Cost computation by convolutional neural network

Patch pairs from left and right images are divided into positive samples and negative samples for training. The positive samples are extracted from corresponding patch pairs which have low matching cost while the negative ones are other patch pairs. For pixel p in left image, the positive sample incorporates a patch $P^L(p)$ with its center position p in left image and a patch $P^R(p_{d_{pos}})$ with the center position $p_{d_{pos}}$ in right image, the negative sample incorporates $P_p{}^L$ and $P^R(p_{d_{neg}})$.

The existing algorithms generally concatenate the convolutional output features trained from single patch and tie the weight with corresponding patch like [5]. The same position in left and right patch may not play the same role for the matching, and the existing method would lose their dependencies. In similarity detection problem [6], flexible weights for corresponding patches lead to higher accuracy than tied weights for pair patches. The proposed method trains corresponding patches in two channels for flexible weights instead of equal. The input $T^{L,R}(p,q)$ represents two-channel patches whose first channel is $P^L(p)$ and the second channel is $P^R(q)$. These two-channel patches are trained through the convolutional neural network to get the result vectors (seen in figure 1) from different scales (figure 2).

2.2. Multi-scale CNN for cost computation

Computing matching cost by CNN will fetch more contexts if we use larger patches as input. But choosing large patches increases the computational cost. The proposed algorithm employs the same size patches followed by a pyramidal process to learn the contexts of adjacent pixels for training.

Since down-sampled images have the same corresponding relationship with original pictures, patches at the same position from different scales have similar influence on matching. The patches from different scales are trained through convolutional layers shown in figure 1. After convolutional layers, the vectors calculated at the same position from different scales are concatenated and passed through fully-connected layers to train multi-scale features (figure 2).

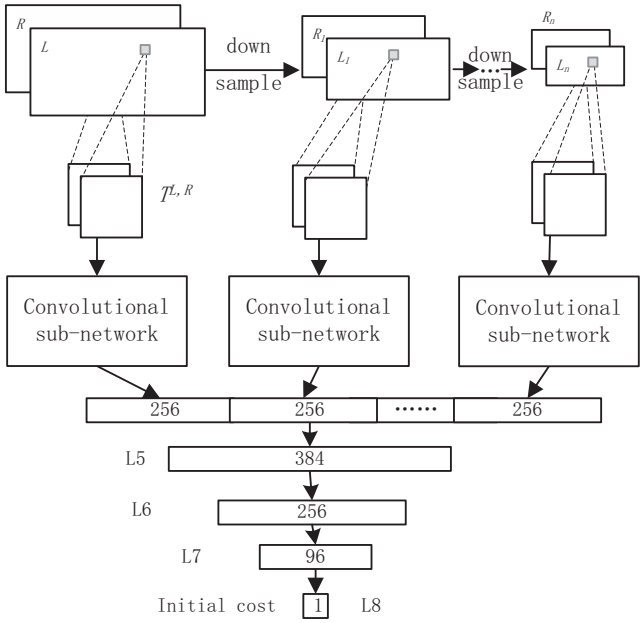

Figure 2. The whole architecture of proposed algorithm.

In our architecture, the effect of pixels in different positions are expressed by the convolutional layers, and the effect of patches from different scales are expressed by the fully-connected layers. Multi-scale algorithm can not only increase the information of contexts for accurate matching but also make the convolutional layers more robust for increasing down-sampled patches as training data.

We apply intensity of image pyramid as input to describe images, the original images L, R are down-sampled repeatedly to fetch the images in different scales L_k, R_k, where the subscript k means the times of down-sampling. The same weights are used in convolutional sub-network L1 to L4 for all scales. And the outputs from L4 of all scales are concatenated and put through four full-connected layers L5 to L8. The final layer L8 has the only output as the initial matching cost.

2.3. Computing Cost for images

The output of the network is used as the initial matching cost:

$$C_{multi}(p,d) = cost(T^{L,R}(p,p_d), T^{L_1,R_1}(p_1,p_{d_1}), \dots), \quad (1)$$

where $cost(*)$ means the output of network, and the input patches is written in brackets. The initial matching costs are calculated by the convolutional results from all scales, employing $T^{L_n,R_n}(p_n,p_{d_n})$ as inputs. For the down-sampled images are quarter of the original images, the testing time would not increase much. In order to keep the balance, three layers from image pyramid are used for initial cost here,

$$C_{init}(p,d) = cost(T^{L,R}(p,p_d), T^{L_1,R_1}(p_1,p_{d_1}), T^{L_2,R_2}(p_2,p_{d_2})), \quad (2)$$

where p_{d_n} means the corresponding pixel of p_d after n times down-sampling.

Since calculating each pixel-wise cost solitary for images expends much computing resources, we test the whole image instead. Some implementation tips from [5] are adaptive improved to suit the proposed architecture. For each disparity d, the testing data consist of left image and an image translated d pixels from right image. These two-channel images are put through the convolutional layers. The results are the features concatenated by vectors from different scales. The features are passed layers with 1×1 kernels whose weights are same with the fully-connected layers [5]. The network should be passed d_{max} times to acquire whole costs, where d_{max} is the maximum disparity of dataset.

3. MULTI-SCALE CROSS-BASED COST AGGREGATION

The costs from initial computation are usually not precise enough. Some post process can be applied to improve the accuracy, such as cost aggregation and disparity refinement.

3.1. Multi-scale Cross-based Cost Aggregation

In this step, matching cost of each pixel is aggregated from its support regions to reduce noise and matching ambiguities of the initial cost. The support regions are built by a multi-scale cross-based cost aggregation in the proposed algorithm.

The configuration of cross-based region is shown in Figure 3, the improved cross-based aggregation [8] modifies some terminal condition from cross-based method [9] and makes the support regions more accurate. The arm $a^r(p)$ for pixel p extends in direction r unless one of the following conditions is triggered at pixel q:

1. $D^L{}_c(p,q) > \tau$ or $D^L{}_c(q, q+r) > \tau$;
2. $D^L{}_s(p,q) > \delta$;
3. $D^L{}_c(p,q) > \tau_{small}$ if $D^L{}_s(p,q) > \delta_{small}$,

where τ and δ are color and space thresholds, $D^L{}_c(p,q)$ and $D^L{}_s(p,q)$ mean the color and spatial distance in image L between p and q, $q+r$ means the pixel after q through direction r. Following these rules, we can construct arms in direction left, right, top, bottom for each pixel. Support regions for pixel p (figure 3) obey the cost aggregation rules as follows:

$$R^{h0}(p) = \{q | q \in a^v(p_h), p_h \in a^h(p)\}, \quad (3)$$

$$R^{v0}(p) = \{q | q \in a^h(p_v), p_v \in a^v(p)\}, \quad (4)$$

where $a^h(p)$ and $a^v(p)$ mean the pixel sets on the horizontal (left and right) and vertical (top and bottom) arms of p.To stable aggregation effect, two support regions are calculated alternately.

In the proposed method, the cross arms are also formed from sampled images. By using the same parameters, we fetch

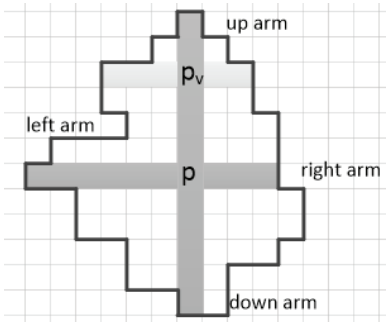

Figure 3. configuration of cross-based region $R^v(p)$

surrounding support regions from L_k, R_k. For the surrounding regions are larger than support regions in original images, we firstly calculate surrounding ones to roughly aggregate the cost. The aggregation algorithm is designed as follows:

$$C_{cross}{}^0(p,d) = C_{init}(p,d),$$

$$C_{cross}{}^i(p,d) = \frac{1}{R^i(p)} \sum_{q \in R^i(p)} C_{cross}{}^{i-1}(q,d),$$

$$R^i(p) = \begin{cases} R^{hk}(p), & \gamma_{k+1} < i \le \gamma_k, \ i \text{ is odd} \\ R^{vk}(p), & \gamma_{k+1} < i \le \gamma_k, \ i \text{ is even} \end{cases}, \quad (5)$$

where the $R^{hk}(p), R^{vk}(p)$ mean the horizontal and vertical support regions from L_k, R_k separately.

3.2. Disparity Refinement

After calculating the matching cost, scanline optimization [12] and Winner-Take-All (WTA) [16] are employed to acquire the accurate dense disparity maps. The disparity results from WTA still contain outliers in occlusion and discontinue regions. We employ the left-right cross consistency check [10] to find the inconsistent pixels. A pixel p would be labeled outlier if the disparity of its corresponding pixel in right image is different from the left one:

$$d^R\big(p - d^L(p)\big) - d^L(p) > \delta_d, \quad (7)$$

where $d^L(p)$ is the disparity of pixel p computed in image L,. δ_d is the threshold.

The common method to refine outliers is filling them with reliable disparities nearby. Once the outliers are detected, linear interpolation is used to deal with the outliers. And the subpixel enhancement [13] is also employed to improve the details.

4. EXPERIMENTAL RESULTS

The results of proposed method are submitted on KTTI stereo 2015 benchmark, which provides 194 pairs of 1242×375 images and their disparity maps. The disparity maps provided are discontinue, the places without clear disparity would be labeled as 0 in ground truth (like Figure 4(b)). The proposed method works better in background with the error rate of 3.38% whose results can enter top 5. The quantitative analysis of innovations is provided below.

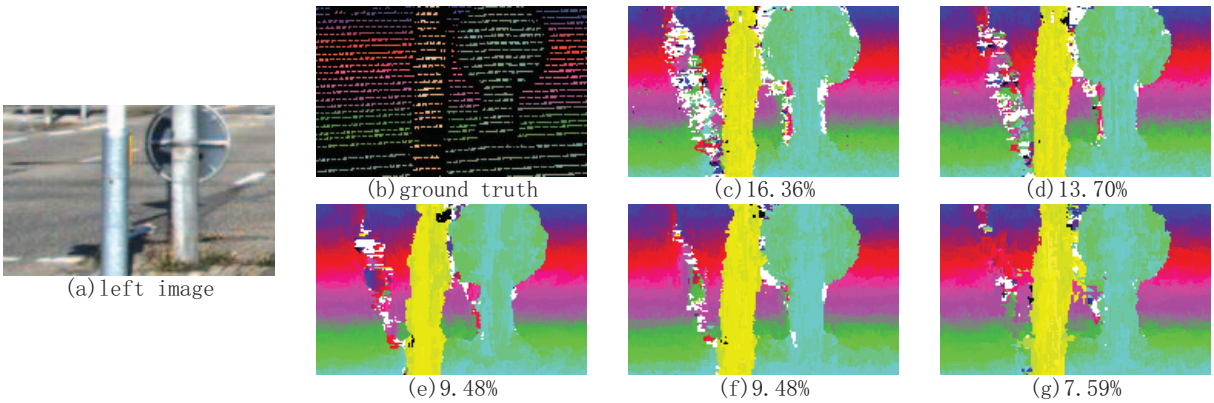

Figure 4. detail difference between the proposed cost calculating method and others, detailed patch and its error rate. (a) original image patch for compare (b) ground truth (c) mc-CNN [5] using parameters same with proposed method (d) 2 channel CNN trained by 9×9 patch (e) 2 channel CNN trained by 17×17 patch (f) proposed CNN for 2 scale (g) proposed CNN for 3 scale

	Accuracy(%)	test time	#PARM
AD-Census[8]	57.190	456ms	-
CNN[5]	84.755	32.334s	18M
2ch CNN	85.537	74.220s	18M
2ch large CNN	89.680	169.62s	45M
2 scale CNN	91.084	91.382s	19M
3 scale CNN	91.605	96.657s	20M

Table 1. Cost matching compare of traditional and CNN, the CNNs use hyper-parameters same as proposed, the CNNs all have 9×9 input patch, while large has 17×17

4.1. Multi-scale CNN for cost computation

We test the KITTI dataset without post processing. The experimental results confirm the proposed algorithm improving the accuracy of matching cost. Testing data is input through different networks with similar hyper-parameters without any other processing. Their accuracy, testing efficiency and numbers of parameters are shown in Table 1.

From the Table 1, we can see that the CNN structure with small patches as input works faster with fewer parameters but has low accuracy; and the CNN with large patches is more precise but slower. Besides, traditional methods can't get accurate results like methods using CNN. Comparatively speaking, the proposed multi-scale CNN structure possesses both accuracy and efficiency. A patch from KITTI 2015 is selected to show the differences in Figure 4, the error rates shown there are just for the selected patches. Table 1 and Figure 4 both show that the multi-scale architecture can achieve similar accuracy with large input CNN, while the proposed methods are more efficient.

In the left region of patch, there is a region without much texture. We can see that the larger input patch for training, the more accurate result of background. The algorithms trained from small patches usually lose these regions, since the features there are similar with others nearby. Once enlarging the size of patch, patches in texture-less regions are more likely to be distinguished. The proposed 3-scale CNN means down-sample the original images twice. This is equivalent with an architecture trained from 36×36 patches. Fetching more contexts, the proposed algorithm works much more accurate than existing matching cost algorithms.

4.2. Multi-scale Cross-based Cost Aggregation

The mismatching outliers in texture-less regions usually form outlier regions containing the pixels similar with each other. Aggregating cost may repeat mismatching directly in some outlier regions if support regions are not large enough. But too large support regions may aggregate the slant planes into flat planes. The proposed aggregation method with gradual deflating support regions can solve this problem.

The cross-based aggregation employing multi-scale works better than single scale under the same iteration times and parameters. The multi-scale method reduces error percentage of 0.354% for the 2-scale results comparing with the improved cross-based aggregation [8] with the same iterations.

5. CONCLUSION

This paper presents a novel method using multi-scale CNN to compute stereo matching cost. We solve the matching problem in texture-less regions by down-sampling contexts. The image pairs are calculated in two channels fetching dependencies instead of single to reduce mismatching. The multi-scale cross-based aggregation algorithm is employed for further refinement. The dense disparity map calculated by proposed method are submitted to the KITTI benchmark and achieves challenging performance.

Acknowledgments This work is supported by the National High Technology Development Plan (863), under Grant No.2011AA01A205,by the National Significant Science and Technology Projects of China, under Grant No.2013ZX01039001-002-003; by the NSFC project under Grant Nos. U1433112, and61170253.

REFERENCES

[1] Marr, David. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Henry Holt and Co., Inc., 1982.

[2] Wan, Xu, et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 37.9(2014):1904-1916.

[3] Zhang, Kang, et al. "Cross-Scale Cost Aggregation for Stereo Matching." *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* IEEE Computer Society, 2014:1590-1597.

[4] Pinheiro, Pedro, and R. Collobert. "Recurrent Convolutional Neural Networks for Scene Labeling." *in International Conference on Machine Learning (ICML)* 2013:82-90.

[5] Zbontar J, Lecun Y. "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches" *Eprint Arxiv* (2015).

[6] S. Zagoruyko and N. Komodakis. "Learning to compare image patches via convolutional neural networks." *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)* IEEE Computer Society, 2014. 2, 5

[7] Spyropoulos A, Komodakis N, Mordohai P. "Learning to Detect Ground Control Points for Improving the Accuracy of Stereo Matching" *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* IEEE Computer Society, 2014:1621 - 1628.

[8] Xing Mei, Xun Sun, Mingcai Zhou, Haitao Wang, Xiaopeng Zhang, et al. "On building an accuratestereo matching system on graphics hardware." *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 467–474, 2011

[9] Zhang, Ke, J. Lu, and G. Lafruit. "Cross-based local stereo matching using orthogonal integral images." *IEEE Transactions on Circuits & Systems for Video Technology* 19.7(2009):1073-1079.

[10] Scharstein, Daniel, and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms." *International Journal of Computer Vision(IJCV)* 47.1-3(2002):7-42.

[11] Huang, Xiaoshui, C. Yuan, and J. Zhang. "Graph cuts stereo matching based on Patch-Match and ground control points constraint." *Advances in Multimedia Information Processing – PCM 2015*. Springer International Publishing, 2015.

[12] Heiko, Hirschmüller. "Stereo processing by semiglobal matching and mutual information.." *IEEE Transactions on Pattern Analysis & Machine Intelligence(PAMI)* 30.2(2008):328-341.

[13] Yang, Qingxiong, et al. "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling.." *IEEE Transactions on Pattern Analysis & Machine Intelligence(PAMI)* 31.3(2009):2347-2354.

[14] Einecke, N., and J. Eggert. "A multi-block-matching approach for stereo." *Intelligent Vehicles Symposium (IV), 2015 IEEE* IEEE, 2015.

[15] Guney, Fatma, and A. Geiger. "Displets: Resolving stereo ambiguities using object knowledge." *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* IEEE, 2015.

[16] Chang, Xuefeng, et al. "Real-Time Accurate Stereo Matching Using Modified Two-Pass Aggregation and Winner-Take-All Guided Dynamic Programming." *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* IEEE, 2011:73-79.

[17] Menze, Moritz, and A. Geiger. "Object scene flow for autonomous vehicles." *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE* Conference on IEEE, 2015..