MC-DCNN: Dilated Convolutional Neural Network for Computing Stereo Matching Cost

Xiao Liu[1], Ye Luo[1], Yu Ye[2], and Jianwei Lu[1(✉)]

[1] School of Software Engineering, Tongji University, Shanghai, China
{1532787,yeluo,jwlu33}@tongji.edu.cn
[2] College of Architecture and Urban Planning, Tongji University, Shanghai, China
yye@tongji.edu.cn

Abstract. Designing a model for computing better matching cost is a fundamental problem in stereo method. In this paper, we propose a novel convolutional neural network (CNN) architecture, which is called MC-DCNN, for computing matching cost of two image patches. By adding dilated convolution, our model gains a larger receptive field without adding parameters and losing resolution. We also concatenate the features of last three convolutional layers as a better descriptor that contains information of different image levels. The experimental results on Middlebury datasets validate that the proposed method outperforms the baseline CNN network on stereo matching problem, and especially performs well on weakly-textured areas, which is a shortcoming of traditional methods.

Keywords: Stereo method · Matching cost · CNN

1 Introduction

In recent years, binocular vision has been widely used in areas such as robots, smart cars, and remote sensing. Most binocular vision systems are based on stereo matching methods. Given a point in three-dimensional space and two images that meet the epipolar constraint, the point will be imaged to two pixels on each image at a same vertical coordinate and different horizontal coordinates. The difference between the horizontal coordinates of matched pixels is called disparity. Using disparity, we can recover the 3D position of matched pixels conversely. The goal of a stereo matching method is to find matched pixels on two images and calculate the disparity of each pixel pair. A typical stereo matching method comprises two steps [1]: firstly, designing a model to compute the matching cost of different disparity of left and right image, then optimizing the matching cost and calculating disparity between two images by using specific prior knowledge. A lot of prior researches have focused on the second step and designed kinds of optimization algorithms to reach a better result from a pre-calculated matching cost [2,3]. However, researchers didn't pay so much attention to computing a better matching cost, which is the basis of subsequent stereo

D. Liu et al. (Eds.): ICONIP 2017, Part III, LNCS 10636, pp. 249–259, 2017.
https://doi.org/10.1007/978-3-319-70090-8_26

methods. Although some traditional stereo correspondence algorithms have been proposed over the past few decades, these algorithms are not intelligent enough to handle situations such as the target pixel is lack of context information.

As we all know, during recent years, convolutional neural networks (CNN) have made great progress and became the mainstream of computer vision. Especially, CNN shows state-of-the-art performance in high level vision tasks including classification, object detection and semantic image segmentation. CNN has high comprehension of image patterns by learning features with more invariance and descriptive power, which are very appropriate for computing matching cost.

Thus this paper focuses on computing matching cost with CNN. We propose a novel CNN architecture, which is named Matching Cost Dilated Convolutional Neural Network (MC-DCNN). Following the work of MC-CNN [4,5], we modify the convolutional layers of MC-CNN from two aspects: using dilated convolution, and concatenating convolutional features of different scales. The former expands receptive field without adding CNN layers and reducing the feature map resolution, while the latter merges features in different scales around the target pixel.

1.1 Related Works

In this subsection, related studies in the field of matching cost are reviewed.

Pixel-wise matching cost is a simple but widely used method [6]. It works pretty good in preserving the structures near the disparity discontinuities. However, the algorithm failed in low-texture and repeated texture area, and is not robust to noise.

Common window-based matching cost, including the sum of absolute or squared difference (SAD/SSD) [7] and normalized cross correlation (NCC) [8], were introduced for providing a more reliable result by using image patches around target pixels. However, outliers frequently occurs near object boundaries when calculating window-based matching cost.

Nonparametric matching cost such as Rank and Census methods solved the above problem but not robust to orientation and distortion because they only rely on the relative ordering of pixel values [10].

A related problem to computing matching cost is learning local image descriptors. Several methods have been suggested for solving the problem of learning local image descriptors, such as boosting [15], convex optimization [16], and convolutional neural networks [17]. However, these methods compared image patches with larger variation, ignored some details which are significant for stereo matching. Moreover, the inclusion of pooling and subsampling to account for larger patch sizes [17] leads to the reduction of resolution.

Žbontar and LeCun [4,5] firstly proposed MC-CNN to compute matching cost of two image patches, which quickly becomes the most popular front-end for stereo matching methods. Their proposed CNN architecture takes a small 11×11 window without the use of pooling, that restricts the receptive field increasing. The method is also post-processed by using cross-based cost aggregation (CBCA) [14], semi-global matching (SGM) [2] and additional refinement procedures.

1.2 Our Motivations and Contributions

Having investigated the literature, we find that convolutional neural network (CNN) is becoming a developing trend to compute the matching cost. However, there is still a lot of room for improvement. MC-CNN [4,5], which is state-of-the-art method and proposed in 2016 to compute the matching cost, still has a 22.81% error rate without post processing on Middlebury datasets [1,11]. Further more, due to the simple architecture of MC-CNN, the limitations of MC-CNN on solving the image patch matching problem are obvious:

(1) MC-CNN has a relatively small respective field (9×9 in KITTI and 11×11 in Middlebury). Consider when people matches two images, the viewer would observe a wide area around the target object to gain context information. However it is hard to enlarge the respective field of it if restricted to original CNN architecture.
(2) The matching cost of the two image patches to be compared is purely determined via a single scale high-level features. However, only using high-level CNN feature leads to the loss of image details. Therefore multi-scale feature, which can represent the target pixel precisely, is needed.

To tackle the aforementioned problems, we propose a novel deep neural network architecture based on MC-CNN. By keeping other parts of MC-CNN unaltered, we made two improvements:

(1) We replace the traditional convolutional operation with dilated convolution to enlarge the receptive field. The dilated convolution operation can expand receptive field exponentially to achieve a better image patch based matching cost computation result. It also avoids the degradation of feature map resolution problem caused by using traditional convolution-pooling operations.
(2) We concatenate features from various convolution layers to incorporate multiple scale context information. The concatenated features increase the ability to describe the target pixel.

Experimental results have shown that our proposed model reaches a better performance than MC-CNN and works well on low texture areas.

2 Architecture of MC-DCNN

In this section, we propose a novel CNN architecture MC-DCNN (Matching Cost Dilated Convolutional Neural Network) for computing the similarity score of two image patches. As is mentioned in Sect. 1.2, we make modifications on the architecture of MC-CNN. MC-CNN is composed of two main parts:

(1) The first part is a pair of siamese networks. The whole network is composed of five convolutional layers (i.e. conv1, conv2, ..., conv5), and each layer has a kernel size of 3×3. Two image patches to be compared are fed into the network as the inputs, and the features extracted from each siamese network are concatenated as the final output of the CNN network.

(2) The second part of the MC-CNN consists of three fully connected (i.e. fc) layers with 384 neurons and a final layer activated by sigmoid function. In order to avoid repeated calculation and reuse the model on the whole image, we replace the fc layers with the convolutional layers of kernel size 1×1. The output of the network is the similarity score of the input patches.

As shown in Fig. 1, in our proposed model MC-DCNN, we made two improvements on the first part of MC-CNN, and keep the original structure of the second part.

(1) We replace traditional convolutional layers with dilated convolution layers. We set dilation rates (1, 1, 2, 4, 8) on the 3×3 convolutional layers successively, increasing the final receptive field from 11×11 to 33×33.
(2) Instead of only using the conv5 output as image feature, we concatenate the output of conv3, conv4 and conv5 as a descriptor that merges multi-scale image information.

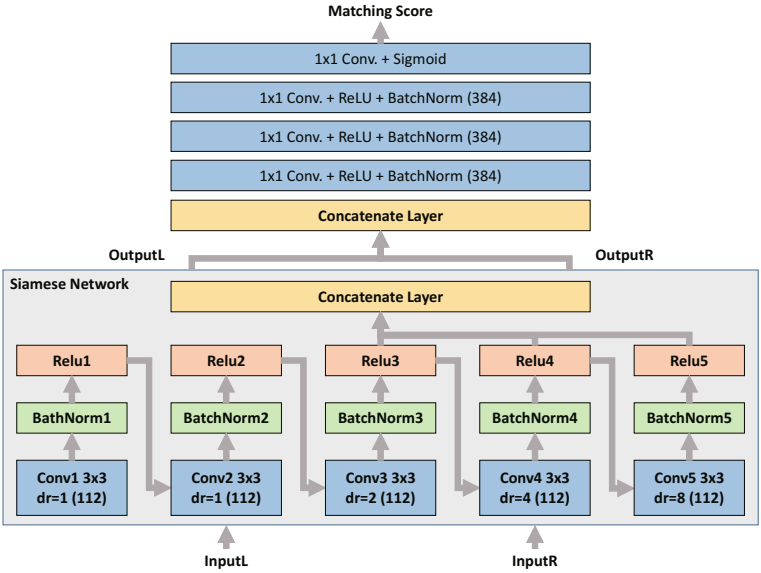

Fig. 1. Architecture of MC-DCNN. Dilation rates (1, 1, 2, 4, 8) were added on the 3×3 convolutional layers. We also concatenate outputs of the last three layers as a final descriptor of input image patch.

In the following subsection, we will firstly introduce the dilated convolution, and then multi-scale feature fusion method.

2.1 Dilated Convolution

It is obviously that enlarging the receptive field of CNN can effectively improve the result of matching cost computing. There are three common methods to enlarge the receptive field: (1) using larger convolution kernels; (2) adding convolution layers; (3) including a few strided pooling layers.

However, the methods are not appropriate for matching cost computing, because: (1) the parameters increase quadratically with the size of convolutional layers, that will reduce operation efficiency; (2) more convolution layers also increase the number of parameters and make the network difficult to train; (3) strided pooling layers can multiply the receptive field by downsampling the feature maps but the target matching cost matrix needs a pixel-level resolution. Even though the resolution can be recovered by fractional strided convolution, small image details filtered by pooling layers are difficult to recover.

In order to enlarge the receptive field efficiently without losing calculation efficiency and feature details, we draw lessons from [12,13], which obtained very good results in semantic segmentation. We also introduce their core idea dilated convolution to our network.

Given a 2-dimensional matrix $\mathbf{I}$ and a convolution kernel $\mathbf{k}$ of size $(2r+1)^2$, the convolution operation at position (p, q) in $\mathbf{I}$ can be defined as:

$$C(p, q|\mathbf{I}, \mathbf{k}) = \sum_{i=-r}^{r} \sum_{j=-r}^{r} \mathbf{I}_{p+i,q+j} \mathbf{k}_{i+r,j+r} \tag{1}$$

As shown in Fig. 2, the receptive fields are enlarged linearly. Three convolutional layers with 3×3 kernel have receptive fields of size 3×3, 5×5 and 7×7 respectively. Given a few convolutional layers of same kernel size $(2r + 1) \times (2r + 1)$, the receptive field size of $n_{th}(n = 1, 2, ...)$ convolutional layers is:

$$\mathbf{S}_n = 2rn + 1 \tag{2}$$

that is, the receptive field is linearly increasing size.

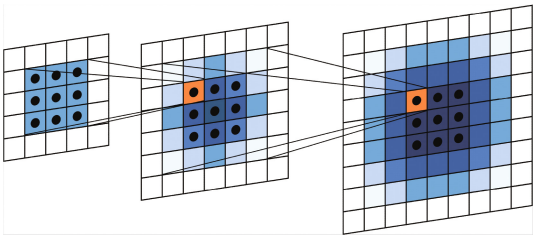

Fig. 2. Three 3×3 convolutional layers without dilation rate. The receptive field sizes are 3×3, 5×5 and 7×7, with linearly increased size.

Now we extend convolution operation. Let d be the dilation rate. A dilated convolution can be defined as:

$$C_d(p, q|\mathbf{I}, \mathbf{k}) = \sum_{i=-r}^{r} \sum_{j=-r}^{r} \mathbf{I}_{p+di,q+dj} \mathbf{k}_{i+r,j+r} \tag{3}$$

As shown in Fig. 3, there are three dilated convolutional layers with same 3×3 kernels, and dilation rates of 1, 2 and 4 respectively. These layers have also successive receptive field, which have larger size of 3×3, 7×7 and 15×15. When using same kernel size $(2r + 1) \times (2r + 1)$, the receptive field size of $n_{th}(n \geq 2)$ convolutional layers with dilation rate d_n is:

$$\mathbf{S}_n = \mathbf{S}_{n-1} + 2r \cdot d_n \tag{4}$$

which is a recursion formula, meanwhile

$$\mathbf{S}_1 = 2r \cdot d_1 + 1 \tag{5}$$

Simplify the Eqs. (4) and (5), the final form of $\mathbf{S}_n$ is:

$$\mathbf{S}_n = 2r \cdot \sum_{i=1}^{n} d_i + 1 \tag{6}$$

If we set $d_n = 2^{n-1}$, similar to the model shown in Fig. 3 and our proposed model, the receptive field size will be:

$$\mathbf{S}_n = 2r(2^n - 1) + 1 \tag{7}$$

which is exponentially increasing size.

It is noteworthy that none kernel weight is added when using dilated convolution. That means this improvement will not influence the performance efficiency theoretically.

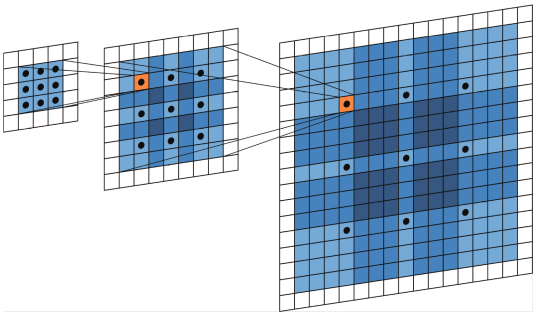

Fig. 3. Three 3×3 convolutional layers with dilation rates 1, 2 and 4 respectively. The receptive field sizes are 3×3, 7×7 and 15×15, with exponentially increased size.

2.2 Multi-scale Feature Fusion

As we know, high-level convolutional features have good invariance, but also lose low level details, which plays an important role in localized matching. To combine features of different level, we concatenate ($\uplus$) the output of conv3, conv4 and conv5 as the final descriptor of target pixel. Let $\mathbf{f}$ be a descriptor symbol, the final output of the siamese network is:

$$\mathbf{f}_{cnn} = \mathbf{f}_{conv3} \uplus \mathbf{f}_{conv4} \uplus \mathbf{f}_{conv5} \tag{8}$$

According to formula (4), the receptive field size of each convolutional layer are 3×3, 5×5, 9×9, 17×17, and 33×33. Features extracted by the first two layers have quite small receptive field, and are low level descriptors that lack of invariance. So we only concatenate the last three convolutional layers as the final descriptor.

3 Experimental Results and Discussion

In this section, we firstly introduce the Middlebury stereo datasets, and then some preparation schemes of the experiments are provided. At last, we conduct the comparison and discussion.

3.1 Middlebury Stereo Datasets

Middlebury stereo datasets [1,11] provide image pairs of indoor scenes with ground truth disparity. The datasets were published in five separates works in the years 2001, 2003, 2005, 2006, and 2014. They also provide an online leaderboard to display a ranked list of all submitted methods. A training set and a test set were provided for training and evaluation with 15 image pairs each, mainly taken from the 2014 dataset. The images are available in three resolutions, full (F), half (H), and quarter (Q). The error is computed at full resolution. If submitted outputs of the method is half or quarter resolution disparity maps, they will be upsampled before the error is computed. We chose to train and evaluate our model on the training dense set with 15 image pairs with half resolution because of limitation of hardware. The dataset were splitted into three parts, each part has 5 image pairs. We compute the error rates of our method by using two parts for training and the other for validation.

Table 1. Splitted training dense set

Name					
Set1	PlaytableP	ArtL	Playtable	Playroom	Recycle
Set2	Teddy	Piano	PianoL	Jadeplant	Pipes
Set3	Adirondack	Vintage	Motorcycle	MotorcycleE	Shelves

3.2 Training Set Preparation

Because our model has a larger receptive field than MC-CNN, we build our own training samples with image patches of size 33×33 segmented from left and right image. Each sample is a pair of image patches centered at position p in left image and position q in right image. For each pixel $p = (x, y)$ in left image, given the ground truth disparity d of p, we generate one positive sample at position q_{pos} and one negative sample at position q_{neg} in right image:

$$q_{pos} = (\langle x - d \rangle, y) \tag{9}$$

$$q_{neg} = (\langle x - d + o_{neg} \rangle, y) \tag{10}$$

where $\langle x - d \rangle$ denotes rounding of x, and o_{neg} is a random number chose from $(-18, -2)$ and $(2, 18)$. We produce samples pixel by pixel in each image pairs in accordance with the above rules, and finally obtain 28 million samples in total.

3.3 Comparison and Discussion

In order to evaluate our method, we compare the result of MC-DCNN with the baseline MC-CNN and other published methods. As is mentioned in Subsect. 3.2, we train and evaluate our model with the divided training dense set for three times. Each time we use samples extracted from two subsets listed in Table 1, about 20 million samples for training, and compute the disparity of the other five image pairs on the remaining subset. We also perform the post-processing pipeline on our raw-disparity map. The pipeline consists of a series of stereo method, including cross-based cost aggregation [14] and semi-global matching [2]. A median filter and a bilateral filter are also performed to smooth the disparity map.

Table 2 shows the results of our method and MC-CNN. Our method outperforms MC-CNN in both raw results and results after post-processing. Quantitatively, the accuracy is improved by 7.2% and 0.61%, respectively. We also compare our method with published methods in Table 3.

Table 2. Comparison of MC-DCNN and MC-CNN. Results on the training dense set of Middlebury datasets. The avg. error represents the percentage of bad pixels with threshold 2.0.

Methods		Avg. error
MC-DCNN	Raw-disparity	**15.71**
	After post-processing	**9.65**
MC-CNN-acrt	Raw-disparity	22.91
	After post-processing	10.26

Table 3. Comparison of our method and published methods.

Methods	Author	Resolution	Avg. error
MC-DCNN	After post-processing	Half	**9.65**
NTDE [19]	Kim et al. (2016)	Half	9.94
MC-CNN+TDSR [20]	Drouyer et al. (2017)	Full	10.2
MC-CNN-acrt [4]	Zbontar et al. (2015)	Half	10.26
MC-CNN+RBS [18]	Barron et al. (2016)	Half	10.8
MC-CNN-fst [4]	Zbontar et al. (2015)	Half	11.7

Moreover, according to [5], as the number of training sample increasing, the error rate of the matching will be decreased. Since we only use the half number of MC-CNN's training samples, better performance is expected when more training samples are used.

Figure 4 shows comparison of disparity maps between our method and MC-CNN. From this figure we can see that our proposed network works well on weakly-textured areas such as floors and walls. This further validates that features extracted via our proposed new network show strong capability on image matching problem.

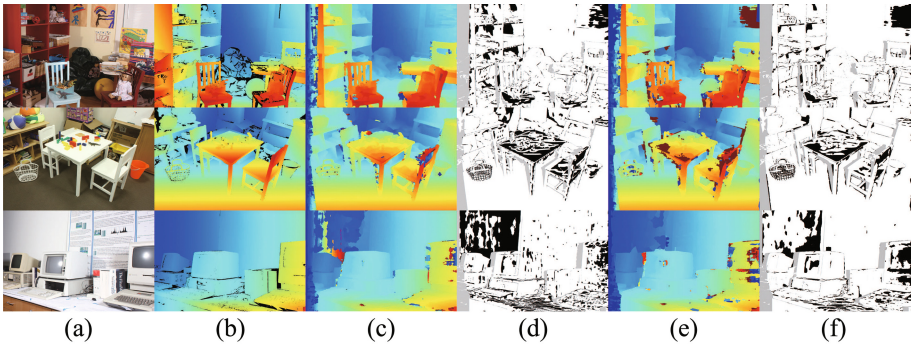

Fig. 4. Comparison of our method and MC-CNN. From left to right: (a) left image of image pair, (b) ground-truth disparity map, (c) result of MC-CNN, (d) error map of MC-CNN, (e) result of our method, (f) error map of our method. Black pixels on error maps represent bad disparity results with threshold 2.0.

4 Conclusions and Future Work

In this paper, we focus on computing matching cost of two image patches by CNN. A novel CNN architecture is proposed to learn features with more invariance and descriptive power for computing matching cost. We improve our model by two main thoughts: (1) enlarge the receptive field of CNN, and (2) merge

multi-scale features. Firstly, we introduce dilated convolution to gain a large receptive field without adding parameters and losing resolution. Secondly, we concatenate the features of last three convolutional layers as a better descriptor that contains information of different image levels. The experiment results prove that the proposed model performs well on weakly-textured areas, which is a shortcoming of previous methods. In the future, we will explore hyper-parameters of our model, and improve the execution efficiency on some other frameworks. We will also transplant the components of our model in other dense image tasks.

References

1. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: IEEE SMBV, pp. 131–140 (2001)
2. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. PAMI **30**(2), 328–341 (2008)
3. Woodford, O., Torr, P., Reid, I.: Global stereo reconstruction under second-order smoothness priors. IEEE Trans. PAMI **31**(12), 2115–2128 (2009)
4. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: IEEE CVPR, pp. 1592–1599 (2015)
5. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. JMLR **17**(1), 2287–2318 (2016)
6. Birchfield, S., Tomasi, C.: Depth discontinuities by pixel-to-pixel stereo. IJCV **35**(3), 269–293 (1999)
7. Kong, D., Tao, H.: A method for learning matching errors for stereo computation. BMVC **1**, 2–11 (2004)
8. Heo, Y.S., Lee, K.M., Lee, S.U.: Robust stereo matching using adaptive normalized cross-correlation. IEEE Trans. PAMI **33**(4), 807–822 (2011)
9. Hirschmuller, H., Innocent, P.R., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. IJCV **47**(1–3), 229–246 (2002)
10. Hirschmuller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans. PAMI **31**(9), 1582–1599 (2009)
11. Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: Jiang, X., Hornegger, J., Koch, R. (eds.) GCPR 2014. LNCS, vol. 8753, pp. 31–42. Springer, Cham (2014). doi:10.1007/978-3-319-11752-2_3
12. Chen, L.C., Papandreou, G., Kokkinos, I.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062 (2014)
13. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
14. Zhang, K., Lu, J., Lafruit, G.: Cross-based local stereo matching using orthogonal integral images. IEEE Trans. CSVT **19**(7), 1073–1079 (2009)
15. Trzcinski, T., Christoudias, M., Lepetit, V.: Learning image descriptors with boosting. IEEE Trans. PAMI **37**(3), 597–610 (2013)
16. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning local feature descriptors using convex optimisation. IEEE Trans. PAMI **36**(8), 1573–1585 (2014)
17. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: IEEE CVPR, pp. 4353–4361 (2015)

18. Barron, J.T., Poole, B.: The fast bilateral solver. In: ECCV, pp. 617–632 (2016)
19. Kim, K.R., Kim, C.S.: Adaptive smoothness constraints for efficient stereo matching using texture and edge information. In: IEEE ICIP, pp. 3429–3433 (2016)
20. Drouyer, S., Beucher, S., Bilodeau, M., Moreaud, M.: Sparse stereo disparity map densification using hierarchical image segmentation. In: ISMM, pp. 172–184 (2017)
21. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: IEEE CVPR, pp. 5695–5703 (2016)