LETTER

Feature Ensemble Network with Occlusion Disambiguation for Accurate Patch-Based Stereo Matching

Xiaoqing YE[†,††a)], ***Nonmember***, **Jiamao LI**[†], ***Member***, **Han WANG**[†], ***and*** **Xiaolin ZHANG**[†], ***Nonmembers***

SUMMARY Accurate stereo matching remains a challenging problem in case of weakly-textured areas, discontinuities and occlusions. In this letter, a novel stereo matching method, consisting of leveraging feature ensemble network to compute matching cost, error detection network to predict outliers and priority-based occlusion disambiguation for refinement, is presented. Experiments on the Middlebury benchmark demonstrate that the proposed method yields competitive results against the state-of-the-art algorithms.

key words: *stereo matching, convolutional neural network, patch-based, occlusion disambiguation*

1. Introduction

Despite having been studied for decades, it remains a challenge to obtain high-accuracy disparities in the field of stereo matching. In contrast to traditional hand-engineered algorithms, recent interest is focused on machine learning strategies to solve such difficulties.

Convolutional neural network (CNN) was first leveraged for computing matching cost, which measured the similarity of two patches exacted from left and right images [1]–[4]. Small patch-based network architecture based on binary classification and followed by multi-step post-processing achieved state-of-the-art performance [1]. Two sorts of networks were investigated in [1] to find an optimized trade-off between time and accuracy. Between them, the faster network adopted dot product to directly measure the similarity score of two patches, whereas the accurate architecture required fully-connected layers to yield the final score. Luo et al. [5] improved the architecture by means of treating it as a multi-label classification problem, in which the labels were all possible disparities. Chen et al. [3] computed two similarity scores separately based on the multi-scale patch pairs and then a fusion was made for the final decision.

In comparison to the aforementioned methods which merely utilize CNN in matching cost estimation stage, end-to-end learning is recently capturing intensive attention [6]. In addition to the matching cost estimation network, another convolutional neural network is also undertaken for obtaining the disparity map in place of winner-takes-all (WTA) strategy [7]. A large synthetic dataset is rendered to train an end-to-end network with images rather than small patches as input [8]. The attained disparity maps does not achieve state-of-the-art performance, yet it is able to recover occlusions, where most patch-based networks fail. Nevertheless, the end-to-end frameworks require strictly large amount of dataset and Graphics Processing Unit (GPU) memory. They are also easy to lose fine details. Thus we follow the patch-based architecture to learn matching cost due to two advantages. First, though the patch-based features tend to be local, they are less prone to overfitting. Second, even small dataset like Middlebury [9] and KITTI [10] are able to provide tens of millions of training patches.

Nevertheless, poor performance was reported in large textureless areas, owing to limited local information with small 11×11 receptive field [1]. 13×13 patches of different resolutions were taken as input of two unattached subnetworks [3]. However, the similarity score of each network was simply fused for final decision. Park et al. [4] enlarged the receptive field by inserting a per-pixel pyramid pooling module before the final decision layer. Thanks to the multi-size pooling unit, larger patches can be taken as input to learn multi-scale information without introducing the fattening effect. Unfortunately, this network is much slower and costs more GPU memory compared with [1] due to d_{max} times extra computation of the pyramid pooling module, where d_{max} denotes the number of disparity levels. Impelled by the above-mentioned works, we investigate a feature ensemble network to reach an optimized trade-off between matching accuracy and computation time.

The main contribution of our work is as follows. First, we achieve more accurate initial disparity maps based on the feature ensemble network at negligible extra overhead. Second, given the raw left disparity map and the left image, we train a dense outlier detection network to predict errors in initial disparity maps instead of traditional hand-crafted left-right consistency (LRC) check. It is indicated that only the computation of the left image rather than a pair of disparity maps is required, reducing the computation cost by half. Third, we further introduce a priority-based strategy to disambiguate the occlusions in complicate scenes and achieve a much lower overall error percentage on Middlebury dataset.

2. The Proposed Approach

The proposed method consists of three main steps: match-

Manuscript received June 2, 2017.
Manuscript revised August 15, 2017.
Manuscript publicized September 14, 2017.

[†]The authors are with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, 200050, China.

[††]The author is with University of Chinese Academy of Sciences, Beijing, 100049, China.

a) E-mail: qingye@mail.sim.ac.cn

DOI: 10.1587/transinf.2017EDL8122

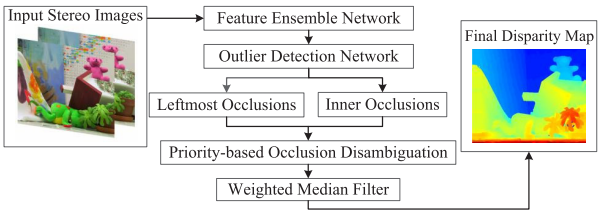

Fig. 1 Framework of the proposed approach. Feature ensemble network is leveraged to obtain matching cost. Errors detected by outlier detection network is refined by occlusion disambiguation and weighted median filter.

ing cost computation network, error detection network and overall refinement (including occlusion disambiguation and weighted median filter), as is shown in Fig. 1.

2.1 Feature Ensemble Network

In this work, we focus on the patch-based network due to its robustness and low requirement for large dataset and GPU memory. In order to preserve fine details, most patch-based networks discard convolution/pooling layers with strides because they lead to a loss of resolution. As a result, the receptive field is restricted to a relatively small size. For example, the patch sizes in [1], [3], [5] are 9×9, 11×11 and 13×13, respectively. They are prone to be ambiguous and produce a noisy matching cost in weakly-textured areas due to limited local context information. In order to increase the receptive field of the network without losing fine details, a per-pixel pyramid pooling module with multiple pooling window sizes instead of multi-stride is adopted, yielding favorable results especially in weakly-textured areas [4]. However, the pyramid pooling module is inserted between the fully-connected layers and final decision layer, i.e., this module has to be recomputed for each possible disparity label, causing a high extra computation cost.

In our feature ensemble network architecture, multi-pooling is employed as [4] does due to its nonparametric feature and efficiency in enlarging receptive field. The module is further modified to achieve an optimized trade-off between time and accuracy. Specifically, first to overcome the problem of high computation cost in [4], the pyramid pooling is appended to the end of the five cascading feature extracting convolutions before fully-connected layers rather than after them. Compared with [4], our pooling module only needs to be computed once over the disparity range, reducing computation cost greatly. Second, the aforementioned operation is also applied to the prior convolution with the pyramid pooling module being replaced by the combination of global max- and average-pooling. The multi-layer and multi-size feature maps are then concatenated to produce coarse-to-fine descriptors. Note that the stride in all pooling modules is set to one so as to preserve resolution. Last, all the fully-connected layers in decision part are substituted for 1×1 convolutions. Our architecture is illustrated in Fig. 2. The multi-pooling module is denoted as:

$$P(F, s, k) = [P(F, s_1, k_1), \ldots, P(F, s_M, k_L)] \quad (1)$$

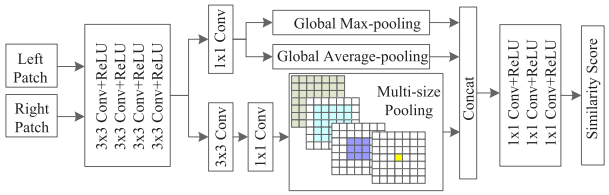

Fig. 2 Feature ensemble network for correspondence estimation.

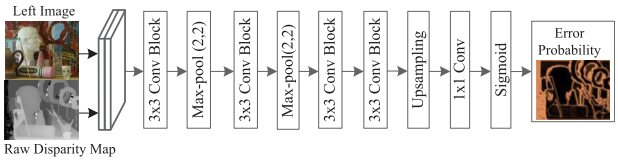

Fig. 3 Outlier detection network. Taken the left image and raw disparity map as input, the network outputs an error probability map.

where F denotes feature maps, s represents different scales and k is the pooling window size without strides.

2.2 Error Detection Network

The raw disparity map is derived from applying WTA strategy to the matching cost computed by our patch-based network. However, the attained disparity map leaves occlusion unsolved, since occlusions are invisible in one of the image pairs where no correspondence can be found. For simplicity we discard complicated cost aggregation and sub-pixel enhancement [1], [4], but mainly focus on restoring occlusions.

LRC check is widely adopted to detect outliers, which requires to obtain a pair of disparity maps. The drawbacks of LRC are two manifolds: one is that some outliers fail to be detected due to errors existing in both images (i.e., False Negatives). Another drawback is that some correct pixels in left disparity map can be marked as errors because of inaccurate disparities in right map (i.e., False Positives).

In contrast, given the raw left disparity map and left image as input, we exploit a dense convolutional neural network to detect outliers. The error detection architecture is shown in Fig. 3. The Conv Block consists of convolutions followed by batch normalization and a rectified linear unit. Mean Square Error loss function is adopted to train the network and produce a probability map with the same resolution. The inputs are normalized to zero mean and unit variance and the target labels are either 0 or 1. Pixels are labeled as outliers if their probabilities exceed a threshold.

2.3 Occlusion Refinement

We classify occlusions into two types: the leftmost and inner occlusions. Leftmost occlusions are attributed to right image lacking the related information and their disparities can come from either the background or foreground. Instead, inner occlusions are generally from the background occluded by foreground objects. Firstly, rather than simply filling occlusions with the lowest disparity [11], [12], we

Algorithm 1 Priority-based inner occlusion refinement

1: **for** each inner occluded point x and its support region $N(x)$ **do**
2: **if** valid points are detected within $N(x)$ **then**
3: Find the maximum and minimum valid disparity within $N(x)$, denoted as $D_{\max}$ and $D_{\min}$.
4: Perform maximum suppression by only reserving valid points whose disparity D_y satisfies $D_{\max} - D_y > D_T$. Update $N(x)$ and the corresponding $D_{\max}$ and $D_{\min}$. D_T is a preset threshold.
5: **if** $D_{\max} - D_{\min} > D_T$ **then**
6: Classify the valid support points into two clusters: $C_1 = \{y \,|D_{\max} - D_y \le D_T\}$, $C_2 = \{y \,|D_y - D_{\min} \le D_T\}$.
7: Compare the similarity of x and C_1 and C_2 using Eq. (2). The one with larger similarity is chosen. Go to Step 11.
8: **else**
9: All valid points belong to the same cluster.
10: **end if**
11: Assign the median value of the cluster to the occluded point x.
12: **else**
13: Find points on the periphery of $N(x)$ (denoted as x_p) and repeat Step 2–11. x_p is first recovered if the condition in Step 5 is satisfied, otherwise it is recovered later in left-to-right order.
14: **end if**
15: **end for**

compare the similarity of the occluded pixel and the possible surfaces when multiple surfaces coexist. Secondly, error propagation is likely to occur in large occluded region due to erroneous assignment to the first few occluded points. Thus we introduce a filling priority and first restore outliers whose neighborhood contains more than one background surface, since they are prone to cause ambiguity.

The main steps of inner occlusion refinement are presented in Algorithm 1. Maximum suppression is performed in Step 4 to remove foreground neighboring points. If multiple disparity values of different surfaces coexist within the newly updated support region (i.e., the condition in Step 5 is satisfied), we select the most probable surface by means of clustering and similarity comparison in Step 6, 7. Note that a local region is assumed to involve no more than two background surfaces for simplicity. The similarity of an outlier x and the potential surfaces in Step 7 is computed by the average comparability between x and all the valid points within a certain cluster, which is defined as

$$S(x, C_i) = \frac{1}{\| C_i \|_0} \sum_{y \in C_i} \exp\left(-\Delta c_{xy}/\lambda_c - \Delta s_{xy}/\lambda_s\right) \qquad (2)$$

where $\|\cdot\|_0$ denotes l_0-norm (i.e., the total number of valid points in the ith cluster C_i), exp(.) represents the weight between the occluded point x and one of the valid points y in C_i. Δc_{xy} and Δs_{xy} are color dissimilarity and spatial distance between x and y, λ_c and λ_s are two constants to balance the color and spatial effects. In other words, the closer the color/distance of the occluded point x and the valid points within a certain cluster, the larger the weight will be. The cluster with a larger average weight is selected for Step 11.

The leftmost occlusion refinement is analogous to inner occlusions except that the maximum suppression in Step 4 is discarded, since in this case the disparity can either come from the background or foreground. We restore leftmost occlusions in a right-to-left order opposite to inner occlusions. Lastly a fast weighted median filter [13] is applied to the recovered disparity map so as to further remove small errors.

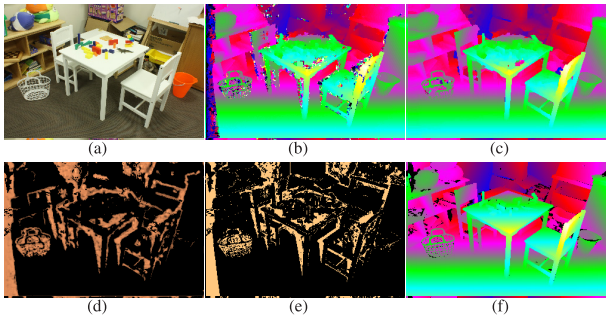

Fig. 4 Results of each step in the proposed framework. (a) Left image, (b) raw CNN output, (c) refined disparity map, (d) estimated error probability map of (b), (e) real error map of (b), (f) ground truth disparity map.

3. Experiment Results

Evaluations are carried out on the Middlebury benchmark. We adopt the same patch size 37×37 as [4]. Pooling window size $k = [27, 9, 3, 1]$ and scale vector $s = [1, 2]$. Parameters of the first five convolutions are borrowed from [1] to initialize the Siamese section in feature ensemble network. $\{\lambda_c, \lambda_s\} = \{16/255, 14/255\}$ in Eq. (2).

First we report the performance of each step in our architecture in Fig. 4. Our feature ensemble network is able to produce a less noisy disparity map at the CNN output (Fig. 4 (b)). Outliers can be directly predicted in a single image through error detection network (Fig. 4 (d)) rather than performing LRC across the left and right disparity maps. The prediction result is basically consistent with the ground truth error map (Fig. 4 (e)). After the occlusion disambiguation and filtering, we obtain the refined disparity map with noise suppressed and most occlusions recovered (Fig. 4 (c)).

Next in order to verify the effectiveness of our matching cost estimation architecture, we first compare the raw outputs of different convolutional neural networks regardless of the post-processing procedures. Algorithms in [1] and [4] are denoted as MC-CNN-acrt and LW-CNN, respectively. By comparing the upper row of *Adirondack* and *ArtL* in Fig. 5, we can see that our approach (Fig. 5 (c)) outperforms MC-CNN-acrt by generating more accurate disparity maps with less noise, especially in weakly-textured areas. This is because the proposed architecture sees a larger field and incorporates multi-layer context information whereas MC-CNN-acrt only provides limited local information, which is difficult to distinguish true matches from false pairs when the local region tends to be textureless. Our network achieves comparable results compared with LW-CNN while being four to ten times faster than it.

The refined results are shown in the lower rows of *Adirondack* and *ArtL* in Fig. 5. Unlike [1], [3], [4] that harness cost aggregations such as SGM [14] to refine the disparity map, we discard them and instead perform occlusion handling directly to raw disparity maps where error prob-

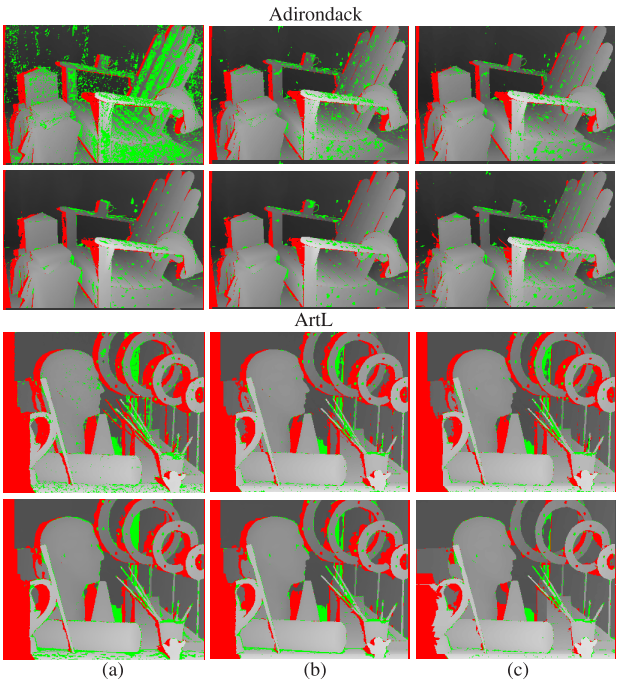

Fig. 5 Disparity maps of raw CNN outputs and after refinement (Error threshold = 1 at half-size). (a) MC-CNN-acrt [1], (b) LW-CNN [4], (c) Ours. For each subfigure (*Adirondack* and *ArtL*), the upper and lower row corresponds to raw CNN outputs and the refined results, respectively. *Nonocc* errors are colored in green and errors in occluded region are in red.

Table 1 The average error percentage of different methods.

Methods	Raw CNN Output		After Refinement	
	Non-occ err	*All* err	*Non-occ* err	*All* err
MC-CNN-fast [1]	25.41%	34.73%	12.06%	21.90%
MC-CNN-acrt [1]	22.55%	32.10%	10.42%	20.07%
LW-CNN [4]	11.82%	21.65%	8.56%	18.08%
The proposed method	**11.69%**	**21.43%**	9.98%	**16.21%**

ability exceeds 0.8. Despite of the multi-step postprocessing, the final disparity maps in MC-CNN-acrt and LW-CNN still contain many errors in occlusions (colored in red in Fig. 5 (a)–(b)). Due to the priority-based occlusion handling and median filter, our refinement is able to recover leftmost and inner occlusions to a great extent. Note that the remaining errors in *ArtL*'s leftmost occlusions (marked in red) in the second row of Fig. 5 (c) are caused by cylinder surface, thus the restored disparities are mostly within the 4-pixel error. Since we do not pay special attention to utilizing multiple methods to smooth the mismatches, our refined results still have some errors in non-occluded (*nonocc*) regions.

Apart from the qualitative comparisons, we also quantitatively compare error percentage in *all* and *nonocc* regions on Middlebury *'training dense'* dataset at half resolution with error threshold equals to 1. As is illustrated in Table 1, our algorithm outperforms MC-CNN-acrt both in raw CNN outputs and final results. Our method even achieves a lower error rate than LW-CNN in raw CNN outputs with less computation time. Since we did not focus on finding a best combination of smoothing parameters via multiple cost aggregations and sub-pixel enhancement, our performance in *nonocc* region is not as good as LW-CNN. More importantly, our architecture achieves a lowest error rate in *all* region, outperforming the other state-of-the-art algorithms.

4. Conclusion

Patch-based learning stereo matching is feasible to implement since it does not require high GPU memory and large dataset, but it also fails in weakly-textured areas due to limited local information. In the letter, we propose a feature ensemble network and an error detection network, followed by occlusion handling to fully utilize the multi-layer and multi-size context information. Experiments demonstrate its competitiveness in weakly-textured areas and occlusions.

Acknowledgments

This work was sponsored by Natural Science Foundation of Shanghai (No.17ZR1436000).

References

[1] J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," J. Machine Learning Research, vol.17, no.1, pp.2287–2318, Jan. 2016.
[2] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," Proc. IEEE CVPR, pp.4353–4361, June 2015.
[3] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs," Proc. IEEE ICCV, pp.972–980, Dec. 2015.
[4] H. Park and K.M. Lee, "Look wider to match image patches with convolutional neural networks," IEEE Signal Process. Lett., 2016.
[5] W. Luo, A.G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," Proc. IEEE CVPR, pp.5695–5703, June 2016.
[6] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," CoRR, vol.abs/1703.04309, 2017.
[7] A. Shaked and L. Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," CoRR, vol.abs/1701.00165, 2017.
[8] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," Proc. IEEE CVPR, pp.4040–4048, June 2015.
[9] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," IEEE Proc. GCPR, pp.31–42, Sept. 2014.
[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," Proc. IEEE CVPR, pp.3354–3361, June 2012.
[11] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," Proc. IEEE ICCV, pp.467–474, Nov. 2011.
[12] X. Sun, X. Mei, S. Jiao, M. Zhou, and H. Wang, "Stereo matching with reliable disparity propagation," Proc. Int. Conf. 3DIMPVT, pp.132–139, May 2011.
[13] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter," Proc. IEEE CVPR, pp.2830–2837, June 2014.
[14] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," IEEE Trans. Pattern Anal. Mach. Intell., vol.30, no.2, pp.328–341, Feb. 2008.