FAST LOCAL STEREO MATCHING USING TWO-LEVEL ADAPTIVE COST FILTERING

Qingqing Yang, Dongxiao Li, Member, IEEE, Lianghao Wang, Ming Zhang

Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China
Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China

ABSTRACT

Recent local stereo correspondence algorithms achieve accurate results by performing effective cost aggregation. In this paper, we solve the cost aggregation problem in the view of cost-volume filtering. A novel concept named "two-level local adaptation" is introduced to guide the proposed filtering approach. Also, a novel post-processing method is proposed to handle both occlusions and textureless regions. The improvement of performance is confirmed by applying it to the proposed stereo correspondence algorithm. The overall method generates competitive results, and outperforms methods that use the similar filtering technique. By implementing the entire algorithm on the GPU, it can achieve about 10 frames/s for typical stereo pairs with a resolution of 640×360 and a disparity range of 20 pixels.

Index Terms— local stereo correspondence, cost filtering, two-level local adaptation, post-processing

1. INTRODUCTION

In local stereo matching algorithms, disparity map is determined by selecting the value with the smallest matching cost from disparity candidates. Thus, cost aggregation becomes the most important step in local stereo algorithms. However, it is not a trivial task as it appears to be. The straightforward aggregation scheme will result in poor disparity maps with fattened edges. To overcome the undesired effect, various algorithms are proposed. Efforts on improving cost aggregation can be classified into two categories: variable support window (VSW) based approaches and adaptive support weight (ASW) based approaches.

Methods in the first category try to find support windows that fit the region size and/or shape, while preventing it from crossing object boundaries. Variable window approach proposed by Veksler [1] performs well when only rectangular support windows are used. For every pixel in the reference image, a square support window is determined by minimizing the local window cost. Zhang et al. [2] propose a fast algorithm in which non-regular support windows are used. One advantage of variable support window based approaches is that integral image technique [3] can be utilized to speed up the aggregation procedure. This makes these cost aggregation schemes relatively efficient.

Adaptive support weight based local methods, which are first introduced by Yoon and Kweon [4], adjust support weights for pixels in a local support window. Variations are also proposed to improve the accuracy. In [5], the authors explicitly deploy smoothness constraint within local objects. Hosni et al. [6] propose to compute the support weights by local geodesic distances. Despite their outstanding performance, one common shortage is their high complexity. Many fast approximations [7, 8] are proposed, but at the price of performance degradation.

Recently, cost aggregation is conducted by filtering on the cost-volume. Edge preserving filters, e.g. bilateral filter [9], are frequently used. He et al. [10] introduced the guided image filtering, which has better behavior near edges. More importantly, it can be implemented exactly under linear complexity. Local methods that deploy it directly report excellent results [11, 12, 13].

In this paper, we propose a new cost-volume filtering method, whose weight kernel is a more general form of the one declared in [10]. A novel concept named "two-level local adaptation" is introduced to guide the proposed filtering approach. A novel post-processing method is also proposed to handle occlusions and textureless regions. The proposed stereo matching algorithm ranks the 9^{th} among about 141 algorithms on the Middlebury stereo evaluation benchmark, and takes the 1^{st} place in all local methods. And the overall stereo matching algorithm can achieve 10 frames/s for typical stereo pairs with a resolution of 640×360 and a disparity range of 20 pixels.

2. TWO LEVEL ADAPTIVE FILTERING

We first define the filter involves the input image C to be filtered, the guide image I, and the filter output $C^{'}$. Then the output value $C_i^{'}$ at a pixel $i = (x, y)$ is defined by

This work was supported in part by the National Natural Science Foundation of China (Grant No. 60802013, 61072081, 61271338), the National High Technology Research and Development Program (863) of China (Grant No. 2012AA011505), the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant No. 2009ZX01033-001-007), Key Science and Technology Innovation Team of Zhejiang Province, China (Grant No. 2009R50003), and China Postdoctoral Science Foundation (Grant No. 20110491804, 2012T50545).

$$C_i^{'} = \sum_j W_{i,j}(I) C_j, \tag{1}$$

where $C^{'}$ is the filtered cost volume. $W_{i,j}(I)$ is the normalized weight of pixel pair (i, j), and depends on the guide image I.

By applying the guided image filtering [10], the weight kernel can be expressed by

$$W_{i,j} = \frac{1}{|w|^2} \sum_{k:(i,j)\in w_k} (1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}). \tag{2}$$

Here, μ_k and σ_k^2 are the mean and variance of the kernel window ω_k in I. ϵ is a smooth parameter, which plays an equivalent role as similarity parameter in bilateral filtering. And $|w|$ is the number of pixels in window w with fixed dimension $r \times r$.

We remodel the weight kernel by varying the kernel size. Kernel windows are adjusted adaptively for local patches. When variable kernel size is applied, the remodeled weight kernel can be expressed by the following expression:

$$W_{i,j} = \frac{1}{|w_i|} \sum_{k\in w_i} (\frac{1}{|w_k|} \sum_{j\in w_k} (1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon})), \tag{3}$$

where $|w_i|$ and $|w_k|$ are the pixel numbers in kernel window w_i and w_k respectively. This is a general form of the original weight kernel expressed by (2), while (2) is a special case when all the kernel windows have the same size, i.e. $|w_i| = |w_k| = |w|$.

With the adaptive kernel size introduced in (3), a hierarchy of two-level local adaptation is expected: the *pixel level* adaptation and the *patch level* adaptation. The *pixel level* adaptation is achieved as in existing ASW based algorithms: support weights assigned to the surrounding pixels are adaptive to the property of the local patch wherein the center pixel lies. The *patch level* adaptation ensures that the property of local patches are adaptive to the content of the guide image.

To figure out why this additional adaptation level is meaningful in guided filtering, it is necessary to explain the characteristics of the above weight function. The numerator $(I_i - \mu_k)(I_j - \mu_k)$ in (3) is positive if I_i and I_j are located on the same side of the average value μ_k, and is negative otherwise. The value of term $1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}$ will change accordingly, so that pixel pairs on the same side are assigned large support weights and those on the different sides will be suppressed. This property ensures that sharp edges can be preserved after filtering.

The weight kernel heavily relies on two terms: mean (μ) and variance (σ^2), which represent the statistical characteristics of a local patch. To improve the accuracy of the assigned support weights, it is meaningful to make the mean and variance represent the property of local patches more properly. As will be explained in section 3.1 in detail, the support region (used in (3)) is now built on the skeleton stretching in four directions with four arms, which are truncated on the border of two different regions, while the original guided image filtering (expressed by (2)) utilizes square support windows with fixed size, resulting in much more outliers. To be more explicit, we can refer to figure 1. The newly introduced *patch-level* adaptation makes the support weights assigned to the surrounding pixels in a more reasonable manner. And adaptive rectangular kernel window ensures fast implementation as that used in [10] still available by applying the integral image technique [3].

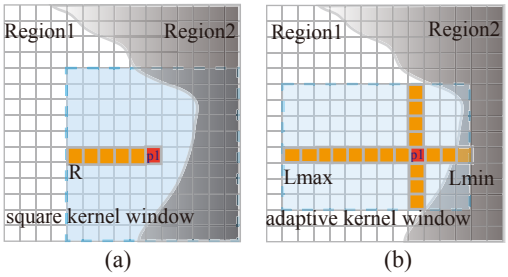

Fig. 1. Kernel windows of (2) and (3). Pixels in shaded area represent outliers. (a) Square kernel window used in (2) and outliers presented in shaded area; (b) Proposed adaptive kernel window used in (3) and outliers presented in shaded area, where fewer outliers are included.

3. STEREO CORRESPONDENCE ALGORITHM

Five steps are carried out to generate the final disparity map. They are: cost computation, local kernel window adjusting, cost-volume filtering, winner-take-all (WTA) disparity selecting and post-processing. Occlusions are detected and handled explicitly in the novel post-processing step.

Cost volume C is built by computing per-pixel matching cost at given disparity values. We combine a truncated version of Birchfield and Tomasi's sampling-insensitive measure (BT) [14] and the truncated absolute difference on the gradient map. Specially, the matching cost of pixel at $i = (x, y)$, when being assigned disparity d, can be expressed by

$$C_{i,d} = (1 - \alpha) \min(C_{i,d}^{BT}, \tau_1) + \alpha \min(C_{i,d}^{GD}, \tau_2). \tag{4}$$

Here, parameter α balances two sub cost terms, while τ_1 and τ_2 are truncation values. C^{BT} represents the cost of BT measure [14], and C^{GD} is the absolute difference of two gradient images. Computed cost volume is then filtered according to the filtering approach described in section 3. And per-pixel disparity D_i is selected by the simple WTA strategy

$$D_i = \arg \min_{d\in R} C_{i,d}^{'}, \tag{5}$$

where $C^{'}$ is the filtered cost volume, and R is the range of candidate disparity values.

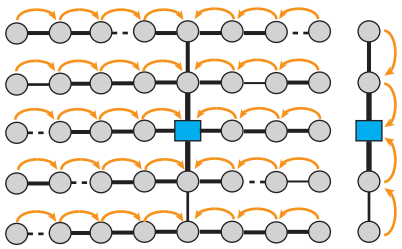

Fig. 2. Simple tree construction and weighted propagation. A simple tree is constructed based on the center pixel, which is represented as blue square, branches with different transport capacity are indicated by lines with different weight, and dashed lines represent branches with the lowest weight.

3.1. Kernel window adjusting

As expressed in (3), the mean (μ_k) and variance (σ^2) values represent the property of local patch w_k. We adjust the size of kernel window aiming at excluding pixels that do not belong to the same region. A moderate percent of external pixels are allowed. Since the edge conservation mainly relies on the adaptive guided filter, the window adjusting policy is not so strict as that used in VSW based algorithms.

In the proposed method, a support window is built upon the skeleton with four arms stretching in four directions. Arm stretching is performed in horizontal and vertical directions separately. Given a specific direction, we search the nearest pixel that has color difference exceeding the threshold τ_a to the center pixel. The color difference $\Delta C_{i,j}$ is computed by

$$\Delta C_{i,j} = \min_{c \in R,G,B} |I_{i,c} - I_{j,c}|, \; L_{min} \leq |i-j| \leq L_{max}, \tag{6}$$

where c is one of the R, G, B color channels. L_{min} and L_{max} are truncation values that prevent arm length being neither too short nor too long.

3.2. Post-processing

Despite the excellent performance of many cost-volume filtering approaches, occlusions must be handled in most local stereo matching algorithms. In this paper, a novel post-processing algorithm is proposed to handle occlusions as well as refine textureless regions.

3.2.1. Weighted cost propagation

For each pixel in the reference image, a simple tree graph is constructed as shown in figure 2. Each node represents a pixel and the root node is the target pixel that is to be processed. Tree branches that connect nodes are weighted by the similarity function

$$T_{p,q} = \exp\left(-\frac{\|I_p - I_q\|}{\sigma^2}\right), \tag{7}$$

where $T_{p,q}$ defines the transport capacity between two connected nodes p and q. I represents the pixel intensity, and σ adjusts the color similarity. Then, the reformulated costs are aggregated through the weighted branches from the peripheral nodes to the root.

3.2.2. Post-processing procedure

Occlusions are first detected by left-right consistency check. Pixels that fail to pass the consistency check are marked as occluded pixels. Cross-check may fail to detect many mismatched pixels, especially in low-texture regions. These pixels are further detected by peak-ratio measuring, which is expressed by

$$M_p^{PKR} = \frac{|C_{p,1} - C_{p,2}|}{C_{p,2}}, \tag{8}$$

where M_p^{PKR} is the calculated peak-ratio of pixel p. $C_{p,1}$ is the best local minimum cost, and $C_{p,2}$ is the second best local minimum cost. Pixels with peak-ratio below a specified threshold η_{PKR} are marked as *unstable* pixels. Cost volume C^P is then reconstructed by the following formulation:

$$C_{p,d}^P = \begin{cases} 0\,, & p \; is \; occluded, \\ |C_{p,d}^{'} - C_p^{best}|\,, & otherwise. \end{cases} \tag{9}$$

Here, $C_{p,d}^P$ is the reformulated cost value of pixel p at disparity candidate d. $C^{'}$ is the filtered cost volume, and C_p^{best} is the best cost value of pixel p after the WTA operation. Then, the reconstructed cost-volume is aggregated by the proposed weighted cost propagation method, and a refined disparity map can be determined by performing another WTA operation. The final disparity map is obtained by replacing the disparity values of pixels that are marked *occluded* and *unstable* with the refined disparity values.

It is common that large regions with low-texture will appear in many stereo scenes. For post-processing methods relying only on local processing, it is hard to recover accurate disparity for these regions. Streak-line filling will also fail if mismatches are not correctly detected, especially when only cross-check is utilized. Figure 3(c) presents the result when the post-processing method in [11] is used. Large mismatched area occurs in the textureless region surrounded by the red rectangular. The proposed method overcomes this problem by performing weighted propagation over the whole image, and mismatched pixels in textureless regions are also detected using peak-ratio measuring. As shown in figure 3(d), disparity in foreground textureless region can be recovered accurately.

4. EXPERIMENTAL RESULTS

The performance of the proposed stereo matching algorithm is evaluated on the Middlebury stereo evaluation website [15]. Constant parameter settings are used throughout for all four

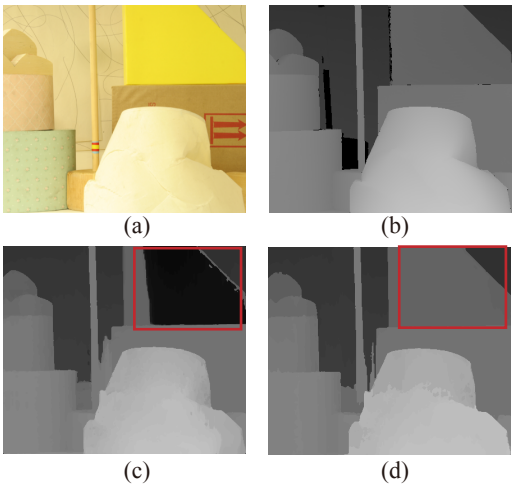

Fig. 3. Comparison of two post-processing methods. (a) Reference image; (b) ground truth; (c) result of post-processing method proposed in [11]; (d) result of the proposed post-processing method.

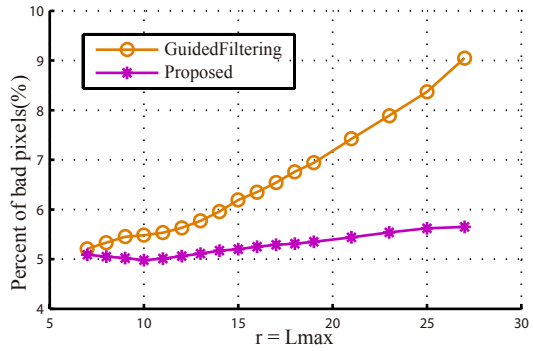

Fig. 4. Performance comparison of guided image filtering and the proposed filtering approach.

benchmark stereo pairs: Tsukuba, Venus, Teddy and Cones. We set parameters $\{\alpha, \tau_1, \tau_2\} = \{0.11, 0.027, 0.008\}$ for cost computation, $\{\tau_a, L_{min}, L_{max}\} = \{0.018, 4, 10\}$ for support window adjusting, $\epsilon = 5 \times 10^{-5}$ for cost-volume filtering and $\{\sigma, \eta_{PKR}\} = \{0.8, 0.3\}$ for post-processing. The evaluation result is summarized in table 1. Our method ranks 9^{th} out of all 141 algorithms as of Nov. 27th, 2012, and is the best local stereo matching method without iterative refinement. The average percent of bad pixel is 4.98% according to the evaluation website.

To compare with the performance of the original guided image filtering, we evaluate the its performance by varying the overall kernel size r. We assign this value to parameter L_{max} to make such comparison. Average percent of bad pixels of all four benchmark images is compared. Comparison result is reported in figure 4. The performance of guided image filtering downgrades significantly when $r > 15$, while the proposed method shows its robustness in this condition.

Table 1. Objective evaluation results according to the Middlebury stereo evaluation platform [15].

Algorithm	Rank	Avg. Error(%)	Error pixels in non-occluded(%)			
			Tsukuba	Venus	Teddy	Cones
Proposed	**9**	**4.98**	**1.04**	**0.17**	**5.71**	**2.44**
CostFilter[11]	25	5.55	1.51	0.20	6.16	2.71
NLFilter[16]	31	5.48	1.47	0.25	6.01	2.87
GeoSup[6]	34	5.80	1.45	0.14	6.88	2.94
PLinearS[13]	37	5.68	1.10	0.53	6.69	2.60
AdaptWeight[4]	74	7.26	1.38	0.71	7.88	3.97

Table 2. Runtime evaluation for benchmark stereo images.

Data Set	Disparity Range	CPU time (s)	GPU time (ms)	speed up
Tsukuba	15	2.12	70	30
Venus	19	2.96	109	27
Teddy	59	8.78	312	28.1
Cones	59	8.76	308	28.5

4.1. Runtime evaluation

Both CPU implementation and GPU implementation are deployed. The runtime is measured on a desktop with Core Duo 3.16GHz CPU and 2GB 800MHz RAM, and no parallelism technique is utilized. The average time consumed by benchmark stereo pairs are: Tsukuba (2.12s), Venus(2.96s), Teddy(8.78s) and Cones(8.76s). The reported runtime is competitive among the state-of-the-art algorithms, which usually need several minutes.

The whole stereo matching algorithm is implemented on a NVIDIA Tesla C2050 GPU. The runtime is reported in table 2. In comparison with the CPU time, the GPU code is about 28 times faster in average for benchmark stereo images. And for typical stereo images with disparity range 20, the average runtime of the proposed method is about $100ms$ on the test GPU. And our GPU code can run faster on new GPUs with higher compute capability.

5. CONCLUSION

In this paper, a new local stereo matching algorithm is proposed, which is based on two-level adaptive cost-volume filtering. Competitive result is reported, which out-performs all local approaches based on adaptive support weight. In addition to the improvement in accuracy, the proposed cost-volume filtering approach also presents its robustness. We have also proposed a novel post-processing method. It can handle both occluded regions and mismatches in low-texture regions efficiently.

6. REFERENCES

[1] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proc. IEEE CVPR*, 2003.

[2] K. Zhang, Lu J., and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. CSVT*, vol. 19, no. 7, pp. 1073–1079, 2009.

[3] Franklin C. Crow, "Summed-area tables for texture mapping," in *ACM SIGGRAPH*, 1984.

[4] Kuk-Jin Yoon and In So Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. PAMI*, vol. 28, no. 4, pp. 650–656, 2006.

[5] Federico Tombari, Stefano Mattoccia, and Luigi Di Stefano, "Segmentation-based adaptive support for accurate stereo correspondence," in *Proc. Advances in Image and Video Technology*. 2007.

[6] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *Proc. IEEE ICIP*, 2009.

[7] Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi, and Neil A. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," in *Proc. ECCV*, 2010.

[8] D. Min, J. Lu, and M.N. Do, "A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy?," in *Proc. IEEE ICCV*, 2011.

[9] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE ICCV*, 1998.

[10] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," in *Proc. ECCV*. 2010.

[11] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE CVPR*, 2011.

[12] Asmaa Hosni, Michael Bleyer, Christoph Rhemann, Margrit Gelautz, and Carsten Rother, "Real-time local stereo matching using guided image filtering," in *Proc. IEEE ICME*, 2011.

[13] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza, "Linear stereo matching," in *Proc. IEEE ICCV*, 2011.

[14] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Trans. PAMI*, vol. 20, no. 4, pp. 401–406, 1998.

[15] D. Scharstein and R. Szeliski, "Middlebury stereo evaluation - version 2," `http://vision.middlebury.edu/stereo/eval`.

[16] Qingxiong Yang, "A non-local cost aggregation method for stereo matching," in *Proc. IEEE CVPR*, 2012, pp. 1402–1409.