Binary Stereo Matching

Kang Zhang, Jiyang Li, Yijing Li, Weidong Hu, Lifeng Sun, Shiqiang Yang
Department of Computer Science, Tsinghua University, Beijing, China

Abstract

In this paper, we propose a novel binary-based cost computation and aggregation approach for stereo matching problem. The cost volume is constructed through bitwise operations on a series of binary strings. Then this approach is combined with traditional winner-take-all strategy, resulting in a new local stereo matching algorithm called binary stereo matching (BSM). Since core algorithm of BSM is based on binary and integer computations, it has a higher computational efficiency than previous methods. Experimental results on Middlebury benchmark show that BSM has comparable performance with state-of-the-art local stereo methods in terms of both quality and speed. Furthermore, experiments on images with radiometric differences demonstrate that BSM is more robust than previous methods under these changes, which is common under real illumination.

1. Introduction

Stereo matching, which is to estimate depth or disparity map from two rectified images (left/right view), is a traditional problem in computer vision. It has wide applications in many areas including image-based rendering, robot navigation, etc. State-of-the-art stereo matching algorithms can generate reasonably good depth maps for images under ideally-configured illumination [10]. However, real stereo images usually have radiometric differences between left and right views, making stereo matching much more difficult [6]. Scharstein et al. gave a detail taxonomy and evaluation of stereo matching algorithms in [11] and according to [11], most stereo methods mainly consist of four steps: *cost computation*, *cost aggregation*, *depth optimization* and *depth refinement* (Figure 1). Based on different strategies adopted in depth optimization, stereo methods can be mainly classified into two categories: local methods and global methods.

In stereo matching, cost computation and aggregation is to construct a matching cost volume $C(x,d)$, where x represents an image pixel from one view, d represents one of all possible disparity values for x and $C(x,d)$ is the matching cost when assigning disparity d to pixel x. Cost volume largely determines the performance of stereo algorithms in both local methods and global methods. In local methods, final disparity assignment for pixel x is calculated using winner-take-all (WTA) scheme:

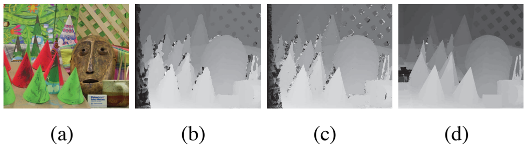


Figure 1. Depth results of BSM after different stages. (a) is the left view of input image pair (Cones dataset [10]); (b)–(d) are depth results after cost computation, cost aggregation and depth refinement

$$d_x = \operatorname*{arg\,min}_{d \in D_d} C(x,d) \tag{1}$$

where d_x is the disparity for pixel x and D_d represents possible disparity ranges (in most of the cases, $D_d = [0, d_{max}-1]$). In global methods, the data term is constructed based on the cost volume. Thus, up-to-date surveys of stereo matching [4, 6, 13] also focus on different approaches applied in cost computation and aggregation steps.

In this paper, we propose a novel cost computation and aggregation approach for stereo matching. By combining BRIEF feature descriptor [2] with a novel binary mask, our method's cost volume is constructed using bitwise operations on binary strings. Then we adopt this approach into WTA scheme, resulting in a local stereo method called *binary stereo matching* (BSM). We have conducted two experiments to demonstrate the performance of our algorithm. We first test our algorithm on Middlebury benchmark [10] and BSM has comparable performance compared with state-of-the-art methods [3, 7, 8, 9] with slightly less time consumption. Furthermore, we also test BSM on datasets with radiometric differences [6]. Experimental results show that BSM is robust to radiometric difference especially to exposure changes, which demonstrates that BSM is much more suitable for unconstrained environment where illumination may have large variations. Another point we want to mention is that the core algorithm of BSM is based on binary and integer computations, so it will still be fast on embedded or mobile devices which do not have powerful floating point units.

The rest of this paper is organized as follows. We review some state-of-the-art local methods in Section 2. Then our cost computation and aggregation approach together with BSM are explained in Section 3. Experimental results and analysis are given in Section 4. Finally we draw conclusion in Section 5

2. State-of-the-Art

In this section, we briefly review state-of-the-art local methods. Bleyer et al. estimate a 3D plane at each pixel by applying PatchMatch [1] into stereo matching and their method is currently top-performer among local methods. Hosni et al. [7] aggregate matching cost by computing geodesic distance from all pixels to the window's center. De-Maeztu et al. [3] and Rhemann et al. [9] both adopt guided filter [5] for cost aggregation and have speed advantages comparing to traditional local methods like [14]. Detailed review of other traditional local methods is proposed in [4, 6, 13]. Overall, most of state-of-the-art local methods use absolute pixel intensity difference for composing cost volume so that their performances drop dramatically under radiometric differences. Some methods explicitly handle radiometric differences like rank and census transform [15], however, their performances on normal images are not so good compared with state-of-the-art local methods [6]. Our binary stereo matching algorithm not only achieves comparable performance with state-of-the-art methods but is also robust to the radiometric differences (especially to exposure changes).

3. Proposed Approach

In this section we will explain our approach in detail. Our binary stereo matching algorithm also follows the classical four steps as stated before.

In cost computation, our approach is completely different with traditional local methods. We directly introduce BRIEF descriptor [2] into cost computation. Thus, BRIEF descriptor $B(x)$ is calculated for every pixel x in the input image pair. According to [2], $B(x)$ is defined as:

$$B(x) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p_i, q_i) \qquad (2)$$

where $\langle p_1, q_1 \rangle, \langle p_2, q_2 \rangle, \ldots, \langle p_n, q_n \rangle$ are n pairs of pixels. Each pair $\langle p_i, q_i \rangle$ is sampled by an isotropic Gaussian distribution in a $S \times S$ window, which is centered on pixel x. And $\tau(p_i, q_i)$ is a binary function which is defined as:

$$\tau(p_i, q_i) = \begin{cases} 1 & : \quad I(p_i) > I(q_i) \\ 0 & : \quad I(p_i) \leq I(q_i) \end{cases} \qquad (3)$$

where $I(x)$ denotes the intensity of pixel x. After calculating the descriptor, i.e. a binary string for each pixel, the cost volume is constructed as:

$$C(x, d) = \| \; B(x) \; \mathbf{XOR} \; B(x_d) \; \|_1 \qquad (4)$$

where x_d is the corresponding pixel of x with disparity d in another view, **XOR** is a bitwise xor-operation. In short, $C(x, d)$ measures the hamming distance between two binary strings.

Directly using BRIEF for stereo matching is a straightforward thought, which can be implemented by adopting $C(x, d)$ in (4) into WTA strategy as mentioned in (1). However, this naive approach will lead to the well-known edge-fattening problem as shown in Figure 1(b). To solve edge-fattening problem, we invent a novel cost aggregation method by introducing another binary string which we call binary mask. Firstly, we define a weight function for pixel pair $\langle p_i, q_i \rangle$ in (2) as:

$$w(x, p_i, q_i) = \max(SAD(x, p_i), SAD(x, q_i)) \qquad (5)$$

where $SAD(x, y) = \sum_{c \in [L,A,B]} |I_c(x) - I_c(y)|$ is the sum of absolute difference between two pixels in the CIELAB color space. Then we get our bitwise mask function for a given pair $\langle p_i, q_i \rangle$ as:

$$\delta(x, p_i, q_i) = \begin{cases} 1 & : \quad w(x, p_i, q_i) \leq T \\ 0 & : \quad w(x, p_i, q_i) > T \end{cases} \qquad (6)$$

where T is set to be the quarter smallest value in the sequence $w(x, p_1, q_1), w(x, p_2, q_2), \ldots, w(x, p_n, q_n)$. Finally, we can use this mask function to compose a binary mask:

$$\Phi(x) = \sum_{1 \leq i \leq n} 2^{i-1} \delta(x, p_i, q_i) \qquad (7)$$

Consequently $\Phi(x)$ is the proposed binary mask for cost aggregation. Incorporating the binary mask into (4), the new cost volume is defined as:

$$C(x, d) = \| B(x) \; \mathbf{XOR} \; B(x_d) \; \mathbf{AND} \; \Phi(x) \|_1 \qquad (8)$$

According to the definition of the binary mask, it will preserve those pixel pairs who have similar depth with window's center. After adopting our cost aggregation method, the edge-fattening effect is ideally removed as shown in Figure 1(c).

Since local WTA strategy cannot handle occluded area, there are a large amount of errors in this region

Table 1. Depth results evaluation.

Methods	Average Error(%)
PatchMatch[8]	4.59
BSM	**5.42**
CostFilter[9]	5.55
P-LinearS[3]	5.68
GeoSup[7]	5.80

(as shown in Figure 1(c)). Besides, a small amount of random errors appear at non-occluded region due to mismatch. Thus, like other local methods, a depth refinement step is needed for removing these errors [11]. In our BSM algorithm, we propose a voting-based depth refinement method. Firstly, a left/right check using two-view depth maps is performed [8] to classify depth results into two categories: *valid* and *invalid*. Then we just refine those invalid pixels' depth using a voting schema. For an invalid pixel x, its refined depth is calculated as:

$$d_x = \arg\max_{d \in D_d} W(x,d) \tag{9}$$

where $W(x,d)$ represents accumulated weight voted from valid pixels, which is defined as:

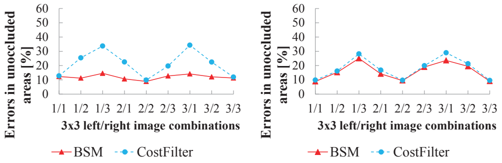
$$W(x,d) = \sum_p \exp(-(\frac{c(x,p)}{\lambda_c} + \frac{e(x,p)}{\lambda_e})) \tag{10}$$

where p represents a valid pixel with disparity d. And the accumulated weight is calculated according to bilateral filter [12] where $c(x,p)$ and $e(x,p)$ are distances between two pixels in color and Euclidean space respectively. In our implementation, parameters for bilateral filter are set as: $\lambda_c = 9, \lambda_e = 16$. As shown in Figure 1(d), errors are corrected after refinement.

4. Experimental Results and Analysis

In this section, we conduct a set of experiments to demonstrate the effectiveness of the proposed stereo algorithm.

4.1. Comparison with state-of-the-art



(a) Different exposure (b) Different lighting

Figure 2. Matching 3×3 left/right image combinations that differ in exposure or lighting conditions.

To compare with state-of-the-art local methods, we test BSM on standard datasets from Middlebury website [10]. In our implementation, we set $n = 4096, S = 26$, and the standard variance for the isotropic Gaussian distribution to be 4. We use the same parameters for all datasets. Comparison between our algorithm with other methods is presented in Table 1. We just list average error rate here to save space and detailed comparison can be found from our submission on Middlebury website [10]. In addition, we give a rough comparison of BSM's computational time with the fastest local methods. Since different methods are implemented under different platforms and there are many programming techniques (parallel computing or not) which affects the speed of the algorithm, we only choose P-LinearS [3] as representative of the fastest local methods for comparison, which has similar hardware configuration and implementation technique with our method (both algorithms are tested with one core and one process). Table 2 shows test result on Teddy dataset, which involves the largest computational cost among all four datasets. Our algorithm is much faster than [3] using a slightly better CPU.

Table 2. Speed evaluation.

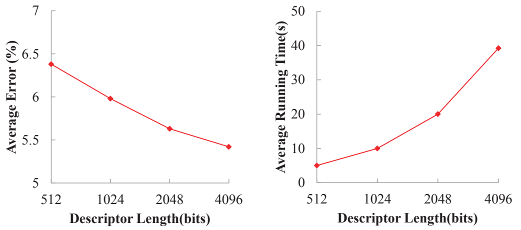
Methods	Processor Frequency(Hz)	Running Time(s)
BSM	**2.67**	**50**
P-LinearS[3]	2.13	94

4.2. Resistance to radiometric differences

The traditional four datasets from Middlebury are configured under ideal illumination. For real images, there may be some radiometric changes. Thus, Hirschmuller et al. proposed new datasets incorporating exposure and lighting changes and evaluated some stereo methods on these datasets [6]. These new datasets give rectified image pairs under three different exposures and three lighting conditions. We conduct the same experiment as that mentioned in [6] and use the same evaluation methodology. For comparison, we also test CostFilter[9] on these datasets (using the source code provided by the author). As shown in Figure 2, comparing to CostFilter[9], BSM has much better performance under exposure changes. As for lighting changes, both BSM and CostFilter do not show good performance. As stated in [6], it is of great difficult to handle local radiometric changes caused by changing the location of the light sources.

4.3. Influence of descriptor length

It is easy to be proved that our algorithm's computational complexity is $O(whnd_{max})$ (w and h are image width and height respectively). Thus running speed of BSM is mainly determined by the descriptor length n.



(a) Quality

(b) Speed

Figure 3. The influence of descriptor length on the performance of BSM.

Also, the descriptor length affects depth map's quality because longer descriptor implies a dense sampling. To show the influence of descriptor length on the performance of BSM, we test BSM with different n on the traditional four datasets from Middlebury. Experimental result is shown in Figure 3, which is consistent with the analysis above. This interesting property of BSM makes it easy to gain different tradeoff between speed and quality in different scenarios.

5. Conclusion

In this paper, a novel cost computation and aggregation approach for stereo matching is proposed. Combining our cost computation and aggregation approach with WTA strategy, we design a new local stereo method called binary stereo matching. The proposed algorithm is mainly based on binary and integer computations, so it is fast and fits for embedded or mobile devices. Experimental results show that BSM has a better performance either on traditional stereo datasets or on new datasets with radiometric differences. In the future, we will definitely incorporate our cost computation and aggregation approach into global optimization and implement BSM on GPU to achieve real-time matching.

6. Acknowledgement

This work has been partially supported by the Development Plan of China (973) under Grant No. 2011CB302206, the National Natural Science Foundation of China under Grant No. 60833009/60933013, and the Research Grant of Tsinghua-Tencent Joint Lab.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 24:1–24:11, New York, NY, USA, 2009. ACM.

[2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision*, pages 778–792, 2010.

[3] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza. Linear stereo matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1708 –1715, nov. 2011.

[4] M. Gong, R. Yang, L. Wang, and M. Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. *International Journal of Computer Vision (IJCV*, 2007.

[5] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proceedings of the 11th European conference on Computer vision: Part I*, ECCV'10, pages 1–14, Berlin, Heidelberg, 2010. Springer-Verlag.

[6] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.

[7] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2093 –2096, nov. 2009.

[8] C. R. Michael Bleyer and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference*, pages 14.1–14.11. BMVA Press, 2011.

[9] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3017 –3024, june 2011.

[10] D. Scharstein and R. Szeliski. Middlebury stereo vision website. http://vision.middlebury.edu/stereo/.

[11] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47:7–42, April 2002.

[12] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society.

[13] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.

[14] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):650 – 656, april 2006.

[15] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the third European conference on Computer Vision (Vol. II)*, pages 151–158, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.