Proceedings of the International Conference on Multimedia and Human Computer Interaction
Toronto, Ontario, Canada, July 18-19 2013
Paper No. XXX (The number assigned by the OpenConf System)

Real-Time Stereo Matching using Adaptive Window based Disparity Refinement

Zheng Tian, Cheng Xu, Xin Huang
Hunan University, College of Information Science and Engineering
2# Lushan South Road, Yuelu District, Changsha, China, 410082
tianzheng@hnu.edu.cn; cheng_xu@yeah.net; lllstar1984@gmail.com

Xiaodong Wang
Xiamen University of Technology, Department of Computer and Information Engineering
394 South Siming Rd, Xiamen, Fujian, China, 361024
xdwangjsj@xmut.edu.cn

Abstract –In this paper, we propose a real-time stereo matching method based on adaptive window, aiming at the trade-off between accuracy and efficiency in current local stereo matching, Considering that the Census transform has good adaptability to image amplitude distortion, but may introduce matching ambiguities in regions with noise or similar local structures, we combine the Census transform with AD (absolute differences) for matching cost initialization, and adopt an iterative cost aggregation method based on ESAW (exponential step adaptive weight), in order to improve the parallelism and execution efficiency. Furthermore, in disparity refinement stage, we build adaptive window based on pixel's color similarity and Euclidean distance for each unreliable point. and classify the unreliable points as 'occlusion' and 'mismatch', so different refinement strategies could be taken for different classifications. Finally the proposed method is optimized with CUDA (compute unified device architecture) and evaluated on graphic processor. The experiment results show that the proposed method is the most accurate one compared with real-time stereo matching methods listed on the Middlebury stereo benchmark.

Keywords: real-time stereo matching; Census transform; iterative cost aggregation; adaptive window; disparity refinement; CUDA

1. Introduction

Stereo matching is the process of finding corresponding points that represent the same object in different view images of the same scene. The distance between the coordinates of corresponding points in different views, namely disparity, could be used to compute the depth of the scene. Stereo matching, which is widely applied into three-dimensional modeling, motion capture and intelligent navigation for vehicles, is a key technique in stereo vision field. Although the principle of stereo matching is simple, it is still highly challenging to accurately computing the dense depth map, due to interferences in realistic scenes, such as the complex actual situation, severe occlusion, lighting variation and image noise.

Scharstein and Szeliski (2002) summarize and categorize the two-view stereo matching method and its evaluation system in a comprehensive way. According to the difference in optimization methods, dense stereo matching can be classified into global method and local method. Global stereo matching correlates depth estimation of adjacent pixels together through introducing smooth constraints, and estimates the disparity of all pixels by solving a global objective function. Classical methods include graph cut (Boykov et al. 2001) and belief propagation (Scharstein et al. 2003). Disparity maps of high precision can be gained by applying global optimization method, but due to its large number of parameters and high computation complexity, this method cannot be applied to systems with real-time requests. Local stereo matching usually adopts a window-based approach to conduct searching and matching individually for each pixel. Because it is unnecessary to take into account the correlation between adjacent pixels, the implementation of this method is relatively easier and efficient. However, this method calls for an assumption that the depth values of the pixels inside the window are the same, which, under many circumstances (especially in bounding areas where depths are not continuous), cannot be fulfilled; therefore, choosing the proper size for windows is the key point in this method. Too small matched windows will invalidate the smoothing effect and amplify depth noise; whereas too big windows will result in error matching in some fine structures and discontinuous areas. Adaptive weight (Yoon and Kweon 2006) and variable window (Zhang et al. 2009) are two classical methods of local stereo matching, which can ensure relatively high computation precision while significantly reducing computation complexity. Recently, in order to improve the effect of stereo matching on occlusion and uncharacteristic areas, some global-local hybrid methods (Hirschmüller 2008) and segmentation-based method (Bleyer et al. 2010) are proposed, further enhancing accuracy of disparity map on the premise of slightly increased amount of computation. However, because of the fact that dense stereo matching requires the separate calculation of disparity for each pixel in the images, sequential computing based on single processors can hardly meet the real-time requests.

The most up-to-date literature show that with the performance improvement of GPGPU (general purpose graphic processing unit), using parallel programming techniques like CUDA (compute unified device architecture) to improve and optimize stereo matching has become a research hotpot. Gong et al. (2007) realized parallel acceleration for 6 mainstream local stereo matching methods, and conducted a comparative analysis of their execution performances. Yu et al. (2010) based on the basis of adaptive weight method, put forward an iterative aggregation method using exponential step-size adaptive weight (ESAW), and improved the algorithm's speed by nearly ten times through parallel optimization on GTX8800 graphic processing unit. Zhang et al. (2011) parallel optimized the variable window method, and proposed a fast bit-wised voting method to accelerate the post-processing procedure of the algorithm, enabling the algorithm to reach a speed of nearly 100fps when processing images with a resolution of 320x240. However, due to the dependence on data during the cost aggregation stage, the above methods always make a certain concession in accuracy in order to realize parallelization. For the purpose of compensate for the accuracy loss of the algorithm during parallelization, Kowalczuk et al. (2013) presented a refinement method featuring iteration of horizontal and vertical similar to the aggregation stage to refine the initial disparity image, which effectively improves the accuracy of disparity map. But this method does not differentiate error points according to their types, so the estimation of disparity in occlusion areas is not quite satisfactory.

In this paper, targeting the balance between algorithm's accuracy and execution time, we present a novel real-time stereo matching method. With an adaptive window based disparity refinement process, the proposed solution can achieve the accuracy equivalent of global method, and further supports parallel processing so it can be used for real-time video applications. The rest of the paper is organized as follows. Section 2 describes the procedure of disparity estimation, including matching cost initialization and cost aggregation. Section 3 represents the details of disparity refinement, mainly focusing on occlusion detection and adaptive window construction. Section 4 shows the experimental results. Finally, we conclude with a brief summary in section 5.

2. Disparity Estimation

2. 1. Matching cost initialization

The core work of stereo matching is to find the corresponding points representing the same object in left and right images. Matching cost is the metric function to judge whether two pixels are similar, and it's the only basis for estimation of disparity. Therefore a good matching cost function plays a very important role in improving the algorithm's robustness in different scenes. Hirschmüller and Scharstein (2009) have studied and evaluated the current commonly used matching cost functions, and after numerous experiments, they found that the matching cost function based on Census non-parametric transformation (Zabih and Woodfill 1994) has the most balanced performance in all test collections, with a particularly good adaptability to intensity distortion caused by lighting variation. The main idea of Census transform is to build a p-centered window for a random pixel p, and generate a bit-string for each pixel in the window except p according to formula (1):

$$C_{\text{census}}(I(p)) = \underset{p'\in N(p)}{\otimes} \xi(p,p').(1)$$

$$\xi(p,p') = \begin{cases} 1 & I(p) < I(p'), \\ 0 & \text{Otherwise.} \end{cases}(2)$$

Here $I(p)$ means luminance value of pixel p, and $N(p)$ is the p-centered transforming window. Thus, the similarity measurement between the two pixels can be expressed by the distance between two Census transformed bit-strings, namely, Hamming distance: the smaller Hamming distance is, the higher the matching degree of the two pixels are.

Census transform calculates the matching cost according to the relationship of size between two pixels instead of relying directly on the values, so it possesses a better adaptability to range distortion. However, we find out that only using Census transform as the matching cost may result in following two problems:

1）Census transform has a lower pixel distinguishing degree for pixels in areas with simple textures and repeated structures. Because the specific pixel values are not considered, different pixels in these areas are likely to have the same Census transform values;

2）Census transform has a lower tolerance of image noise. Because Census transform is highly dependent on the central pixel, when noise points appear in the central pixel, the matching result is unpredictable. It is also mentioned by Hirschmüller and Scharstein (2009) that the matching results based on Census transform are not quite satisfactory in an environment of intense noise.

Aiming the above problems and considering that the commonly used AD cost function has better matching effects in areas with smooth textures, in this paper, we adopt the matching cost function that combines AD and Census transform together. For a random pixel p in the left view, establish a p-centered 9x7 window and define the matching cost of p as follows:

$$C(p,d) = \rho(D_{\text{census}}(p,d), \lambda_{\text{census}}) + \rho(D_{\text{AD}}(p,d), \lambda_{\text{AD}}).\ (3)$$

Here d is the presupposed disparity value, D_{census} and D_{AD} are the matching costs based on Census transform and AD respectively, $\rho(D,\lambda)$ is a normalized function, with the following computing formula:

$$D_{\text{census}}(p,d) = D_{\text{Hamming}}(C_{\text{census}}(I_{\text{L}}(x_p, y_p)), C_{\text{census}}(I_{\text{R}}(x_p - d, y_p))),(4)$$

$$D_{\text{AD}}(p,d) = \min\{|\ I_{\text{L}}(x_p, y_p) - I_{\text{R}}(x_p - d, y_p)\ |, \tau\},(5)$$

$$\rho(D,\lambda) = 1 - \exp(-\frac{D}{\lambda}).(6)$$

$I_{\text{L}}(x,y)$ and $I_{\text{R}}(x,y)$ refer to the gray values of corresponding pixels in left and right view respectively, C_{census} is calculated from Formula (1), λ_{census}, λ_{AD} and τ are prior parameters. Normalized function avoids the matching cost overly leaning to one certain type, and the λ is used for easily controlling the weights of the two matching cost, which enables the algorithm to combine well the advantages of Census and AD, and to have a better adaptability to intensity distortion and noise. Figure 1 demonstrates the stereo matching results using AD, Census and AD+Census as matching cost. It can be seen that in Tsukuba and Venus, the effect of AD is better than that of Census, but in Teddy and Cones, the effect of Census is better than that of AD. Nevertheless, AD+Census, combining the advantages of the two, performs the best under the four sequences and with the lowest average error rate of all three methods.

2. 2. Iterative cost aggregation

The work following matching cost initialization is to aggregate the initial matching cost for each pixel and to estimate the disparity value of pixels according to the aggregated matching cost. Matching cost aggregation is the most time-consuming step in local stereo matching algorithm, so it is necessary to exchange limited computation accuracy for a significant reduction of execution time. in this paper, in order to get a balance between execution time and accuracy, we take the strategy of exchanging accuracy for time in the stage of disparity estimation for significantly reducing the execution time; and then take a refinement step in post-processing stage to verify and update the initial disparity values for improving the computation accuracy with little time cost.

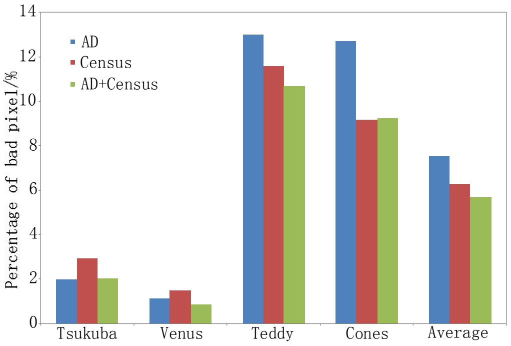


Fig. 1. Results of stereo matching using different matching cost

Inspired by the idea of ESAW (Yu et al. 2010), in this paper, we adopt the exponential step-size iterative aggregation method to estimate disparity. The basic idea of this method is to construct a support window for each pixel, and then to use adaptive weight to aggregate the initial matching costs of the pixels in the window. It simplifies the original 2D aggregation into two 1D aggregations, namely horizontal and vertical aggregation, and uses an exponential iterative process to further reduce the time complexity. After calculating the final matching cost of central pixel under every hypothetic disparity value, we use the common WTA strategy to select the smallest matching cost as the initial disparity value of center pixel.

Compared to the traditional adaptive weight method, the iterative aggregation method is more suitable for parallel processing, but may lose some accuracy during computation, which is mainly caused by the following two aspects:

1）The traditional adaptive weight method computes the weights both in target image and reference image during the stage of matching cost aggregation, but the parallel method only computes the pixel weight of the target image and abandons the weight of the reference image, as shown in formula (7). This is more suitable for parallel computation, but the matching cost after aggregation is lower discriminative for different pixels.

$$E(p, \overline{p_d}) = \frac{\sum\limits_{q\in N_p, \overline{q_d}\in N_{\overline{p_d}}} \omega(p, q)\omega(\overline{p_d}, \overline{q_d})C(q, d)}{\sum\limits_{q\in N_p, \overline{q_d}\in N_{\overline{p_d}}} \omega(p, q)\omega(\overline{p_d}, \overline{q_d})} \approx E(p, \overline{p_d}) = \frac{\sum\limits_{q\in N_p} \omega(p, q)C(q, d)}{\sum\limits_{q\in N_p} \omega(p, q)} .(7)$$

2）In terms of weight calculation, exponential step-size iterative aggregation method decomposes the computation of a single weight into multiplication of multiple middle pixel weights, as shown in formula (8). This computing method would have error when there is a big difference between the middle pixel r_k and the two target pixels p,q, meanwhile the difference of target pixels themselves are relatively small. This error then lead to mismatch, this kind of mismatching is particularly prominent in the occlusion and edge area of the image.

$$\omega(p, q) \approx \omega(p, r_1) \times \omega(r_1, r_2) \cdots \times \omega(r_k, q). \ (8)$$

However, the proposed method is able to achieve a high level of accuracy while maintaining real-time operation by adopting an adaptive window based disparity refinement method.

3. Disparity Refinement

3. 1. Occlusion and mismatch detection

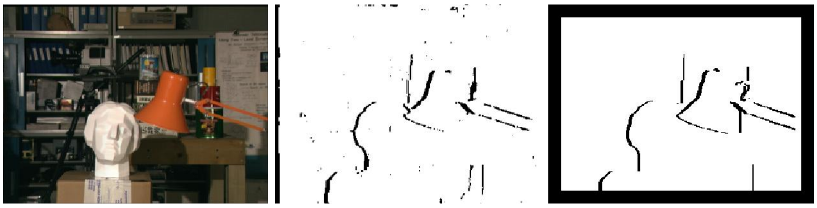
In this paper, we propose a disparity refinement method based on adaptive windows to improve the disparity accuracy with little time cost. First we conduct a consistency check on left and right disparity images through formula (9). d_L and d_R in formula (9) represent left disparity image and right disparity image respectively. Inspired by the work of Hirschmüller (2008), this paper further classifies error points into occluded points and mismatched points. According to the geometrical principle of stereo vision, if the corresponding point of a certain pixel point in the left view cannot be found in the right view, then its corresponding epipolar line in the right view should have no intersection with the disparity value of the right view, otherwise if the point in the scene can be seen in both views, then there must be an intersection. Therefore, we can use this principle to judge whether the error point is caused by occlusion or mismatch.

For the error point p in the left image, define $d_L(x,y_p)$ and $d_R(x,y_p)$ as the disparity values of pixels in same the row of p in left and right views respectively, define p's corresponding epipolar line $e(p,d)$ as a point set about d, as illustrated in formula (10). If $d_R(x,y_p)$ and $e(p,d)$ do not intersect, p should be perceived as the occluded point, otherwise, it should be perceived as the mismatched point.

$$d_{\mathrm{L}}(x_p, y_p) = d_{\mathrm{R}}(x_p - d_{\mathrm{L}}(x_p, y_p), y_p).(9)$$

$$e(p, d) = \{(x, y)_{\mathrm{R}} \mid x = x_p - d, y = y_p\}.(10)$$

Figure 2 shows the results of using the above method to detect occlusion on Tsukuba's left view. In the figure, there are the original image, the results of this paper and the ground truth successively from left to right. As the right disparity image are also needed in occlusion tests, which exist some errors when generated by our method, so as shown in figure 2, there are some differences between the ground truth and the detected occlusion.



(a) Original image (b) Detected occlusion (c) Ground truth

Fig. 2. Result of occlusion detection on Tsukuba

3. 2. Disparity refinement based on adaptive window

After classifying error points, different updating strategies can be adopted according to different error types. It can be known that the occluded point usually come from background pixels, therefore they can be updated by values with the minimum disparity in their adjacent background areas; while mismatched point usually occurs on object surfaces with complex textures, in either foreground or background, and the disparities in the adjacent areas do not vary too much, so these error points can be refined according to the statistical result of disparity values in adjacent areas. This paper proposes to build an adaptive window based on color similarity and Euclid distance for each error point. Theoretically, areas with similar colors should possess similar disparity values, the values of error points can be refined on the basis of disparity statistics inside of the window. The construction process of adaptive windows is similar to original variable window (Zhang et al. 2009), for any random error point p, define a four-element set as illustrated in formula (11):

$$S(p) = \{h_p^-, h_p^+, v_p^-, v_p^+\},(11)$$

$$\begin{cases} H(p) = \{(x, y) \mid x \in [x_p - h_p^-, x_p + h_p^+], y = y_p\}, \\ V(p) = \{(x, y) \mid x = x_p, y \in [y_p - v_p^-, y_p + v_p^+]\}. \end{cases}(12)$$

$$\begin{cases} D_{\mathrm{s}}(p, q) < L_1, & (a) \\ D_{\mathrm{c}}(p, q) < \tau_1 \ \& \ D_{\mathrm{c}}(p, q^+) < \tau_1, \text{when} \quad D_{\mathrm{s}}(p, q) < L_2, & (b) \\ D_{\mathrm{c}}(p, q) < \tau_2 \ \& \ D_{\mathrm{c}}(p, q^+) < \tau_2, \text{when} \quad L_2 < D_{\mathrm{s}}(p, q) < L_1. & (c) \end{cases}(13)$$

$$D_{\mathrm{c}}(p, q) = \max_{n=\mathrm{R,G,B}}(| I_n(p) - I_n(q) |).(14)$$

The elements in the four-element set constitute two orthogonal line segments in horizontal and vertical directions where p is located, as illustrated in formula(12). The length of these two segments can be decided under the constraints of color and distance in formula (13), where L_1、L_2、τ_1 and τ_2 are the preset parameters, and their relations meet: $L_1>L_2$ and $\tau_1>\tau_2$, $q+$ means the adjacent pixel following q, $D_s(p,q)$ represents the Euclid distance between p and q, $D_c(p,q)$ is the color difference between p and q, which calculating as illustrated in formula(14).

Being different from original variable window (Zhang et al. 2009), this paper improves the constraints. Formula (13a) set an upper limit value for the size of the window; formula (13b) regulate that not only the color difference between any random pixel q and p, but also the color difference between the q's following pixel and p must be less than a threshold, so that part of the interference of image noise can be excluded; while formula (13c) regulate that when the distance between q and p is more than L_2, a smaller threshold value is needed to further constrain the color difference between q and p, so as to better control the accuracy of the window size, making it possible to gain a relatively large window in areas where the color difference is small, and avoiding the window to be too small in areas with rich textures.

After establishing the foregoing orthogonal segments for each error point, all horizontal segments of the pixels on the vertical segment where p is located are combined together to form the p-centered adaptive window, as shown in formula (15).

$$N(p) = \bigcup_{q\in V(p)} H(q). \quad (15)$$

Figure 3 shows the result of the adaptive window for four pixel points in Teddy left view. It can be seen that the shape of the window can change arbitrarily, and the bounding area of the window is basically consistent with that of the color, whereas in areas with smaller color differences, the window can reach the preset maximum value.

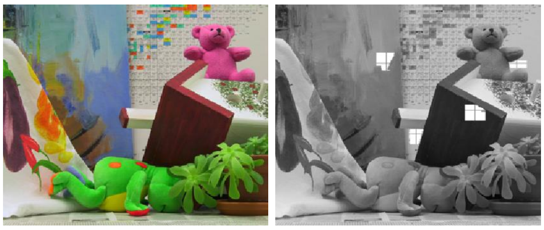


Fig. 3. Examples of adaptive window construction

After establishing the adaptive window, according to the previous analysis, the error points caused by occlusion could be updated by the minimum value of the reliable point in the window; and the error points caused by mismatch could be refined by the reliable point with the largest number inside the window. The concrete refinement method is shown in formula (16), where $N^*(p)$ stands for the set of reliable disparity points in the window centered in p; $\Psi(d_q)$ is the statistical histogram of $N^*(p)$.

$$d_p' = \begin{cases} \min\limits_{q\in N^*(p)} (d_q), & \text{if occlusion,} \\ \arg\max\limits_{q\in N^*(p)} (\psi(d_q)), & \text{if mismatch.} \end{cases} \quad (16)$$

At last, this paper processes a 3x3 median filtering on the disparity image to further remove disparity noise. After that, the procedure of the algorithm is completed.

4. Experiment and Analysis

The proposed method has been parallel optimized, tested and evaluated on the nVIDIA GeForce GTS450 graphic card (192 CUDA cores and 1G graphic memory). Parameter settings involved in the optimization process are provided in Table 1. Other relevant parameters are set to the default value in reference paper.

Table. 1. Parameter settings

Parameters	λ_{census}	λ_{AD}	τ	L_1	L_2	τ_1	τ_2
Value	50	10	60	32	17	20	6

In terms of accuracy evaluating, this paper verifies the proposed method on the classic Middlebury benchmark: Tsukuba、Venus、Teddy and Cones. The results of disparity estimation are illustrated in figure 4, where the first column is the original image, the second is the ground truth, the third is the results of proposed method and the fourth column is the difference between the result of proposed method and the ground truth, in which the black area means the error on occlusion region and the gray area means the error on non-occlusion region. From the result of disparity estimation, it can be seen that the proposed method has more accurate result on the occlusion area, especially in Venus and Teddy, and the area with repeated textures (the net-like structure in the upper right) in Cones. But for the object's bounding area in Tsukuba and the bottom area of Teddy, the result of proposed method need further improvement.

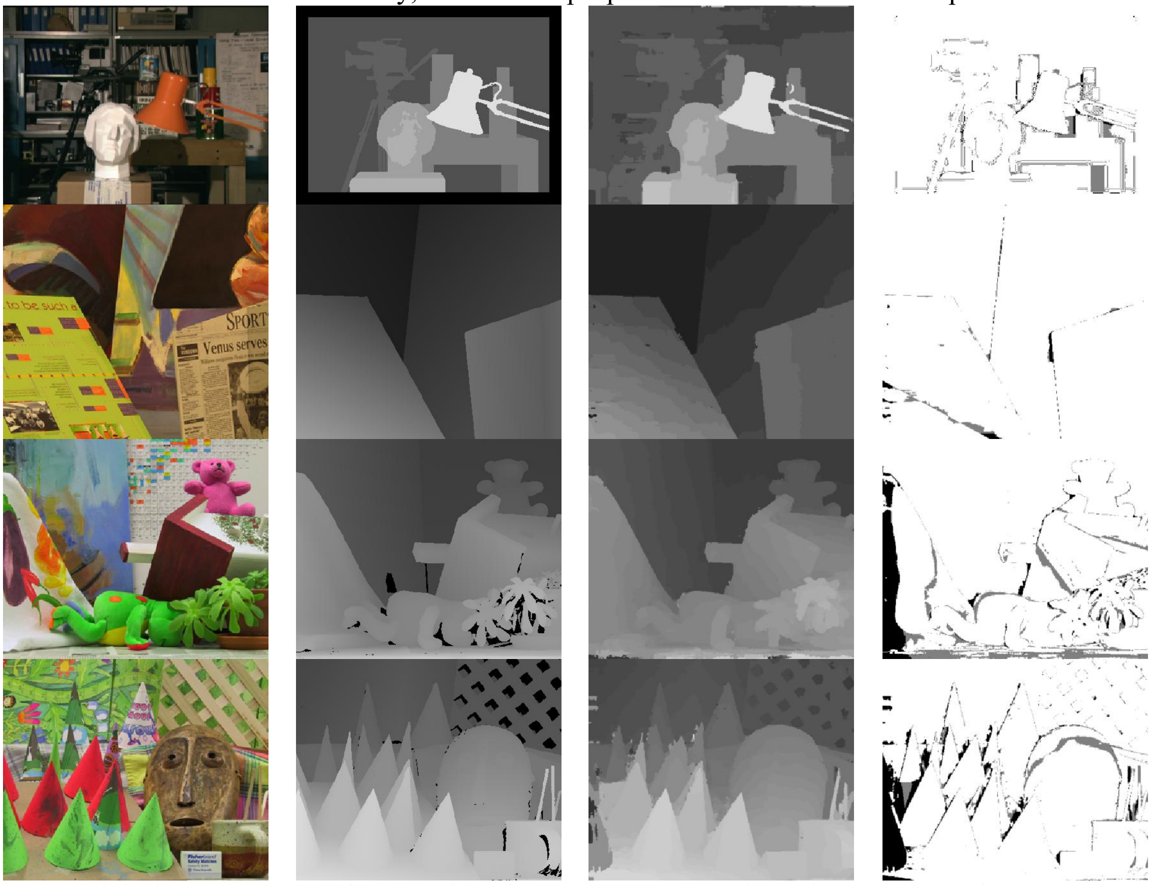

(a) Original Image (b) Ground Truth (c) Proposed Result (d) Errors

Fig. 4. Results of proposed algorithm on the Middlebury datasets

Submitting the third column of image in figure 4 to the Midllebury evaluation website, the quantitive results as shown in table 2 can be achieved. Table 2 displays the error pixels' proportion in all pixels, and the threshold value of error is 1, meaning that if the difference between computed disparity value and the standard value is bigger than 1, then it should be marked as an error. There are three types of errors all together, namely, *nonocc*: the error pixels' proportion in non-occlusion areas; *all*: the error pixels' proportion in all areas; and *disc*: the error pixels' proportion in discontinuous areas. Other algorithms involved in the comparison include: the effective SemiGlob method (SemiGlob), the classic adaptive weight method (AW), the method based on iterative disparity refinement (IterRefine), the method based on adaptive windows (VariableWnd) and the exponential step-size adaptive weight (ESAW). The latter three of them are real-time stereo matching method based on CUDA, and the IterRefine is the most up-to-date real-time stereo matching method in Middlebury database.

From table 2 we can see that since ESAW method is AW based method with a simplification on aggregation process, it's not as effective as the AW method, which is consistent with the previous analysis. IterRefine method has conducted disparity refinement on the basis of ESAW, and has a better effect even than AW method. However, the proposed method is better than IterRefine, bringing the highest accuracy among all real-time stereo matching methods, and its average error rate is equivalent to that of SemiGlob method, with nearly 1 point lower than AW method and 30% higher than that of ESAW method. This should be mainly attributed to the accurate computation results in occlusion and complex texture regions.

Table. 2. Quantitative results on the Middlebury datasets. / %

Algorithm	Avg. % Bad Pixels	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
Proposed	5.69	1.66	2.04	8.82	0.42	0.84	3.10	5.36	10.7	14.7	2.98	9.26	8.40
SemiGlob	5.76	2.61	3.29	9.89	0.25	0.57	3.24	5.14	11.8	13.0	2.77	8.35	8.20
AW	6.67	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26
IterRefine	6.20	1.45	1.99	7.59	0.40	0.81	3.38	7.65	13.3	16.2	3.48	9.34	8.81
VariableWnd	7.60	1.99	2.65	6.77	0.62	0.96	3.20	9.75	15.1	18.2	6.28	12.7	12.9
ESAW	8.21	1.92	2.45	9.66	1.03	1.65	6.89	8.48	14.2	18.7	6.56	12.7	14.4

In terms of execution time evaluation, the proposed method is mainly compared with ESAW method, which is widely recognized as a good algorithm of real time. Besides, the proposed method is an improved method based on ESAW. Table 3 provides the average execution time of the two methods on four Middlebury evaluating sequences, which is the total time of estimating two disparity maps for both left and right views. It can be seen that the proposed method, having a procedure of disparity refinement which ESAM does not have, is slower than ESAW. The speed of processing Tsukuba by proposed method can reach nearly 20fps. But for images with larger resolution ratio and more disparity hypothesis levels, the proposed method cannot meet the real-time requirements. However, it's worth mentioning that the algorithm's execution time is closely related to the performance of GPU and the degree of parallel optimization. Considering the good parallel potential of proposed method, if using a GPU with more CUDA cores and larger memory, or conducting more in-depth parallel optimization according to the architecture, the execution time of the algorithm will be markedly improved.

Table. 3. Execution time of proposed algorithm and ESAW. / ms

Algorithm	Tsukuba	Venus	Teddy	Cones
	384x288, 11 disp. levels	434x383, 16 disp. levels	450x375, 44 disp. levels	450x375, 44 disp. levels
Proposed	53.7	114.8	219.1	232.6
ESAW	34.8	75.3	174.3	202.4

4. Conclusion

This paper proposes a real-time stereo matching method based on adaptive window disparity refinement. The main features of this method include: 1) in association with the advantage of AD and Census transform, the proposed method possesses a better adaptability to the variations of the image's intensity distortion; 2) in the stage of disparity refinement, based on the assumption that areas with similar colors should have similar disparity values, an adaptive window is built for each error point, and the occlusion error and mismatch error are refined respectively according to statistics of disparity values in the adaptive window, which enable the proposed method to have better estimation results in areas with discontinuous depths; 3) the proposed method takes high advantage of parallelization, and after parallel optimization, it can reach real-time process for Tsukuba images. Compared to the existing real-time stereo matching methods listed on the Middlebury evaluation platform, the proposed method is the most accurate one. The next step is to conduct further parallel optimization for the algorithm to implement real-time process on the mainstream format images. In addition, there are too many preset parameters in proposed method, and the use of adaptive parameter mechanism is also worth researching.

Acknowledgements

This work was supported by a Fundamental Research Funds for the Central Universities (No.531107040421) and Hunan Provincial Innovation Foundation For Postgraduate（CX2011B137）. The authors are particularly grateful to the support from the R&D Centre of STMicroelectronics in Singapore.

References

Bleyer M., Rother C., Kohli P. (2010). Surface stereo with soft segmentation. “Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,” CA, USA, June 13-18, pp. 1570-1577.

Boykov Y., Veksler O., Zabih R. (2001). Fast Approximate Energy Minimization via Graph Cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence, 23, 1222-1239.

Gong M., Yang R., Wang L., Gong M. (2007). A performance study on different cost aggregation approaches used in real-time stereo matching. Int. Journal of Computer Vision, 75, 283-296.

Hirschmüller H. (2008). Accurate and efficient stereo processing by semi-global matching and mutual information. IEEE Trans. on Pattern Analysis and Machine Intelligence, 30, 328-341.

Hirschmüller H., Scharstein D. (2009). Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans. on Pattern Analysis and Machine Intelligence, 31, 1582–1599.

Kowalczuk J., Psota E.T., Perez L.C. (2013). Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences. IEEE Trans. on Circuits and Systems for Video Technology, 23, 94-104.

Scharstein D., Szeliski R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. Journal of Computer Vision, 47, 7-42.

Scharstein D., Szeliski R. (2003). High-accuracy stereo depth maps using structured light. “Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,” Wisconsin, USA, June 16-22, pp. 195-202.

Yang Q., Wang L., Yang R., Stewenius H., Nister D. (2008). Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. IEEE Trans. on Pattern Analysis and Machine Intelligence, 31, 492-504.

Yoon K.J., Kweon I.S. (2006). Adaptive support-weight approach for correspondence search. IEEE Trans. on Pattern Analysis and Machine Intelligence, 28, 650-656.

Yu W., Chen T., Franchetti F., Hoe J.C. (2010). High performance stereo vision designed for massively data parallel platforms. IEEE Trans. on Circuits and Systems for Video Technology, 20, 1509-1519.

Zabih R., Woodfill J. (1994). Non-parametric local transforms for computing visual correspondence. “Proc. Of European Conf. on Computer Vision,” Stockholm, Sweden, May 2-6, pp. 151–158.

Zhang K., Lu J., Lafruit G. (2009). Cross-based local stereo matching using orthogonal integral images. IEEE Trans. on Circuits and Systems for Video Technology, 19, 1073-1079.

Zhang K., Lu J., Yang Q., Lafruit G., Lauwereins R., Gool L.V. (2011). Real-Time and Accurate Stereo: A Scalable Approach with Bitwise Fast Voting on CUDA. IEEE Trans. on Circuits and Systems for Video Technology, 21, 867-878.