ROBUST STEREO MATCHING USING CENSUS COST, DISCONTINUITY-PRESERVING DISPARITY COMPUTATION AND VIEW-CONSISTENT REFINEMENT

Duc Minh Nguyen, Jan Hanca, Shao-Ping Lu, Adrian Munteanu

Vrije Universiteit Brussel (VUB),
Department of Electronics and Informatics,
Pleinlaan 2, Brussels, Belgium

iMinds V. Z. W.,
Department of Future Media and Imaging,
G. Crommenlaan 8, Ghent, Belgium

e-mail: {mdnguyen, jhanca, splu, acmuntea}@etro.vub.ac.be

ABSTRACT

Stereo matching has been a one of the most active research topics in computer vision domain for many years resulting in a large number of techniques proposed in the literature. Nevertheless, improper combinations of available tools cannot fully utilize the advantages of each method and may even lower the performance of the stereo matching system. Moreover, state-of-the-art techniques are usually optimized to perform well on a certain input dataset. In this paper we propose a framework to combine existing tools into a stereo matching pipeline and three different architectures combining existing processing steps to build stereo matching systems which are not only accurate but also efficient and robust under different operating conditions. Thorough experiments on three well-known datasets confirm the effectiveness of our proposed systems on any input data.

Index Terms— Stereo matching, depth estimation, disparity estimation, binocular stereo

1. INTRODUCTION

Stereo matching aims to find pixel correspondences between multiple images of the same scene captured from different view points. Those correspondences form a disparity map, which represents a horizontal displacement between the pixels and their matches in another view. Finally, the disparity map can be easily converted into a depth map of the original scene given the intrinsic and extrinsic camera parameters.

Thanks to this close disparity-depth relationship together with a wide range of applications of depth information, stereo matching has been receiving a lot of attention from the computer vision community for many years. Typically, stereo matching methods in the literature generally follow a processing pipeline with four main steps: (1) matching cost computation, (2) matching cost aggregation, (3) disparity calculation and (4) disparity refinement [17]. These methods are most commonly classified into two main groups: local and global methods. Algorithms classified into the first group calculate the disparity value for each pixel separately while the others formulate an optimization problem for the whole image and solve it for the disparity labeling function [17].

In the early days, the stereo matching problem was solved mainly using local methods. After that, due to the appearance of very efficient minimization techniques such as graph cuts and belief propagation, global methods became dominating. Most of the researchers in the domain focused on finding the best way to formulate the stereo matching problem and on improving the optimization algorithms [20, 19, 10, 3, 24]. Global methods proved to generate state-of-the-art results [24, 22, 13] and are widely used until now. Recently, the matching cost aggregation step attracted a lot of attention making adaptive support weight for cost aggregation processing step an active research direction [27, 8, 1, 11, 23]. However, the trend seems to be deviating in the direction of machine learning stereo matching-based algorithms [26, 14], as large datasets containing ground-truth disparity maps became available.

Until now a great number of articles have been published in stereo matching domain and many techniques have been proposed for each of the four main processing steps. Nevertheless, this domain still receives a lot of interest from the community due to the high demand for a method which performs not only accurately but also efficiently and robustly under varying operating conditions. Building an effective system is still challenging due to several reasons. Firstly, designing a good integration to maximize the powers of existing ideas is not straightforward [24, 12]. Secondly, most of the proposed algorithms were only evaluated and compared using the standard Middlebury benchmark [18], which consists of stereo pairs captured under controlled laboratory settings. The problem is that methods focusing only on one dataset are likely to be optimized for the certain input and may not perform well in other conditions. In particular, many methods achieving best results in the Middlebury ranking do not produce accurate disparity maps when applied to KITTI input

images [4], which are captured outdoors using moving cameras. Similarly, multiple top-ranked algorithms fail to generate high quality results for the MPEG FTV dataset [21].

In this paper, we describe a simple, yet effective framework to combine available tools in order to build a stereo matching method which is accurate, efficient and robust under varying test conditions. Based on that, we propose and compare three depth estimation architectures, employing three different widely used disparity computation techniques. All three prototypes use Census transform for matching cost calculation and employ a discontinuity-preserving disparity computation tool. Finally, estimated disparity maps are refined by utilizing weighted-median filtering. The experimental results show (i) the superior performance of the proposed methods compared to their baselines, and (ii) one of the three methods is ranked among the best local methods on the KITTI benchmark.

To sum up, the main contributions of our paper are as follows:

- A framework to integrate existing stereo matching techniques, in order to build a method with high effectiveness, efficiency and robustness
- A thorough evaluation of different stereo matching methods on three well-known datasets with different characteristics

The remainder of the paper is organized as follows: Section 2 presents our integration framework of different processing techniques and Section 3 explains in detail the proposed prototype methods. Next, Section 4 describes the experiments and their results. Finally, we conclude the paper in Section 5.

2. ROBUST STEREO MATCHING FRAMEWORK

The proposed framework for building a robust stereo matching algorithm is illustrated in Fig. 1. There are three main components in this framework: the census matching cost, the disparity computation technique and the view-consistent refinement. Although more complex methods exist for the matching cost computation and disparity refinement, we opt to use these simple techniques in our systems. With the proposed framework, the complexity of stereo matching depends only on the disparity computation tool. If an efficient local disparity computation technique is employed, the constructed system will be computationally cheap and suitable for systems with limited resources. Otherwise, the algorithm will be appropriate for more complex hardware.

2.1. Matching cost computation

The matching cost computation procedure is the first step in any stereo matching system. A large number of cost metrics have been proposed in the literature. Hirschmuller et al.

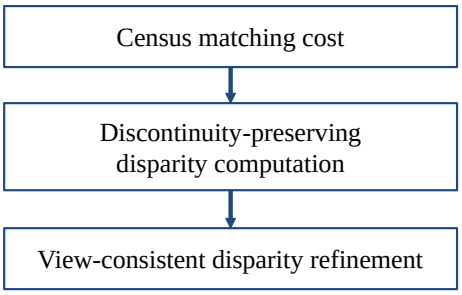


Fig. 1: Proposed integration framework.

carried out a thorough analysis of the performance of a number of cost metrics and proved that the census cost produces the best and the most robust results under radiometric differences including differences in exposure, lighting, vignetting and noise [7]. More advanced methods for matching cost computation, like fusion of multiple methods [15] or machine learning approach [26] have been proposed recently. Nevertheless, the former does not perform well when apply with cost aggregation and disparity refinement whereas the latter has a very high computational complexity and hence is not suitable for systems with limited resources. We selected the census matching cost for our disparity estimation methodology since it brings the best tradeoff between accuracy, robustness and complexity.

The census matching cost is a Hamming distance between two bit strings representing the pixels, in which the bit strings are calculated using the Census transform [25]. This transform compares the value of each pixel p with the surrounding pixels in a window and sets a bit to 1 if the corresponding neighbor has a smaller intensity value than p, and 0 otherwise. Hence, this transform relies on the relative ordering of local intensity values, that is, it has the ability of capturing local image structure, which makes it also reliable for lack-of-texture regions.

2.2. Disparity computation

Although the census matching cost is very effective for dealing with the lack-of-texture problem, its performance around disparity discontinuities is not as good as in case of other matching costs. Nevertheless this problem has to be taken into account while executing the disparity computation step, which is the second component in our systems. Many of such techniques have been proposed in the past [17]. Accurate disparity discontinuities can be captured using both local techniques with adaptive support weight-based cost aggregation, e.g. [27, 8, 23] and global techniques, incorporating disparity discontinuities as a constraint, e.g. [19].

Because the performance of different methods on varying input data is not clear, we have selected two local and one global algorithms for further evaluation. The detailed description will follow in Section 3.

2.3. Disparity refinement

The last step in any stereo matching system is the refinement step, which influences the results significantly as it corrects disparity values for both occluded and non-occluded regions. In our framework, we follow the technique proposed by Hosni et al. [8]. To be specific, the refinement step in our approach consists of four sub-steps:

- Left-right consistency check: detects the unreliable pixels in the disparity map which do not satisfy the view-consistency condition
- Hole filling: fills the disparity values of the unreliable pixels
- Weighted median filtering: smooths the hole-filled disparity map and removes streaking artifacts resulted from the hole filling process
- Un-weighted median filtering: removes the remaining noise

The key part of the described disparity refinement process is weighted median filtering, which replaces the value of a pixel by a weighted median value of its neighbors. The weight assigned to a pixel in the neighborhood of another pixels can be determined using both their spatial and color distances. One big draw-back of the refinement in [8] is that it uses a naive implementation of weighted median filtering, increasing the runtime of the whole algorithms significantly. To overcome this drawback, we use the fast implementation of the weighted median filtering algorithm proposed by Zhang et al. [28], which does not add a noticeable computational burden to the system.

3. PROTOTYPE METHODS

Three prototype methods have been constructed from our framework presented in the previous section. As mentioned earlier, all proposed architectures share the same matching cost and disparity refinement technique, but they differ in the disparity computation step. This processing stage has the biggest impact on the overall performance of the algorithm and therefore we selected three methods that belong to different categories in the literature.

3.1. Prototype 1: Cross-based algorithm

The first technique, which is the cross-based technique proposed in [27], is a local technique with main focus on the cost aggregation step. The method derives for each pixel an adaptive-shaped support window for cost aggregation based on the color information in its neighborhood. That is, only neighboring pixels with similar color are included in the support region. Thus, the matching cost for a pixel will not be aggregated across object borders which are assumed to occur at color edges, which ensures that the disparity discontinuities are well preserved. Moreover, one strong point of this technique is that it can be implemented very efficiently using an integral image technique [27].

3.2. Prototype 2: Adaptive support weight algorithm

The second architecture makes use of the local technique classified in the literature as an adaptive support weight algorithm. Specifically we employed a well-known cost volume filtering algorithm proposed in [8], which is highly ranked among the local methods on the Middlebury benchmark [18]. The basic idea of this approach is to execute guided image filtering [6] on the matching cost volume formed by the matching costs of each pixel for all possible disparities in the range. Because the color image is used as the guidance and the filter itself has edge-preserving characteristics, this procedure guarantees that the matching cost is not aggregated across color edges. Hence, it ensures that the disparity discontinuities are preserved, as done also in Prototype 1.

3.3. Prototype 3: Belief propagation algorithm

Finally, the third selected method is a global algorithm, formulating the disparity labeling problem using a pair-wise Markov random field (MRF) model. Although such systems appeared in stereo matching literature a long time ago, belief propagation-based methods are still achieving state-of-the-art results [24, 22, 13]. A drawback of the belief propagation algorithm and global disparity computation is the high computational complexity. Nevertheless, the belief propagation algorithm is well-suited for parallel implementation, hence, a GPU implementation can significantly reduce the computational time [5].

Denote by d a disparity labeling function; the energy function to be minimized is the sum of a data term E_{data} and a smoothness term E_{smooth}:

$$E(d) = E_{data}(d) + E_{smooth}(d) \tag{1}$$

$$E_{data}(d) = \sum_{p} C\left(x_p, y_p, d_p\right) \tag{2}$$

$$E_{smooth}(d) = \sum_{p} \sum_{q \in N_p} \rho\left(d_p - d_q\right) \tag{3}$$

in which $C(x_p, y_p, d_p)$ is the census cost of assigning pixel p the disparity value d_p.

The truncated linear model is adopted as the smoothness term. The smoothness cost, incorporated into the model as a prior knowledge, is computed by using mean-shift segmentation [2] on the color image. In particular, the smoothness cost of assigning disparity values d_p and d_q to pixels p, q respectively is calculated as follows:

$$\rho\left(d_p - d_q\right) = w_{pq} \times \lambda \times min\left(|d_p - d_q|, K\right) \tag{4}$$

with

$$w_{pq} = \begin{cases} 1, \text{if } q \in S_p \\ \alpha, \text{otherwise} \end{cases} \quad (5)$$

where K is the truncation value; λ is a constant which controls the influence of the smoothness constraint on the disparity estimation; S_p is the segment that contains pixel p and α is a constant in the range $[0 - 1]$.

This formula ensures that the smoothness cost is less significant if pixels p and q belong to different image segments, allowing the algorithm to assign different disparity values to both pixels. Otherwise, the smoothness cost has a stronger impact on the labeling process, which results in setting the same disparity value to the pixels. Similarly to the previous architectures, this design helps to preserve the disparity discontinuities.

We used 8-connected neighborhood system in our MRF implementation, in which each pixel is connected to eight surrounding pixels. The energy function shown in Eq. (1) is optimized using the accelerated belief propagation algorithm of [10].

4. EXPERIMENTS

Thorough experiments were carried out to evaluate the prototype disparity estimation methods presented in the previous section. For simple identification, we denote the first prototype method built with cross-based cost aggregation technique as 'CS'; the second method built with guided image filtering for cost aggregation as 'GF' and the third prototype methods as 'BP'. The performances of these architectures are compared to their reference implementations, i.e. the techniques proposed by Zhang et al. [27] and Hosni et al. [8], and to the depth estimation reference software (DERS) from MPEG [9].

4.1. Experimental setup

Three different well-known datasets were employed in our experiments. We divided each dataset into non-overlapping training and testing sets. The training set was used to tune the methods' parameters while the testing one was used to evaluate the efficiency of each system.

The most popular input, namely the Middlebury data, contains images taken in laboratory settings. Stereo image pairs in this set come with ground-truth disparity maps, obtained using structured light techniques [18, 16]. In total, we selected 31 stereo image pairs, of which 12 were used for training and 19 for testing.

In order to guarantee the satisfactory performance in different lighting conditions, we evaluated our methods using another two datasets, i.e. the KITTI dataset [4] and the MPEG-FTV dataset. The first one contains 194 training stereo image pairs and 195 testing pairs. Unlike the Middlebury dataset, only gray-scale input images are used for tests, which makes this dataset more challenging. Moreover, the images are captured outdoors under different weather conditions.

Finally, we selected Balloons and Newspaper multi-view video sequences [21] from the MPEG-FTV collection. Although both sequences were recorded indoors, they are very challenging due to multiple light sources and complicated textures (or lack of textures). We took only two views from each sequence and temporally sampled the two sequences each 20 frames. This resulted in 30 stereo pairs in total, 15 per video.

Disparity estimation methods can be evaluated in direct or indirect way. If ground-truth disparity maps are available, the percentage of bad matching pixels (BPP) [17] or the error rate is used for evaluation of the algorithm. A pixel is considered a 'bad pixel' if its disparity differs from ground-truth disparity value by more than a threshold (1.0 and 3.0 for Middlebury and KITTI respectively). The percentage of bad pixels may be evaluated either for all pixels in the images or for only non-occluded pixels, resulting in two different evaluation figures, denoted by BPP_all and BPP_nonocc respectively.

In case ground-truth disparity maps are not available, indirect assessment of the quality has to be performed. We evaluated the disparity maps by first synthesizing new views using the generated disparity maps and then calculating the objective image quality comparing the synthesized images with the captured images. This evaluation was conducted due to the fact that depth-image-based rendering (DIBR) is one of the most important applications of depth maps. In such case, the quality of disparity image itself is not as important as the quality of the synthesized picture. The renderer available as a part of HEVC Multiview reference software [1] was used for DIBR.

In our experiments, we first trained and evaluated all three methods independently on the Middlebury and KITTI datasets. To be specific, we tested the algorithms to select the best maximum aggregation radius L for the CS method, filter window radius r for the GF method and constant balancing data and smoothness costs λ in case of the BP method. All other parameters for disparity computation are fixed according to the reference papers. Parameters for the matching cost computation and refinement are also fixed for all prototype methods. Only the best performing method among the three was selected to run on the MPEG-FTV dataset.

The best parameters obtained during the training step are presented in Table 1.

Table 1: Parameters learned during the training step.

	L	r	λ
Middlebury dataset	9	9	1.0
KITTI dataset	7	9	1.0

[1] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/

4.2. Experimental results

Middlebury dataset

The final results for both the non-occluded pixels and the whole images are shown in Fig.2. As can be seen from this figure, our CS method has much higher accuracy than its reference method [27] when applied to the Middlebury dataset. In particular, the proposed implementation has 15.07 % lower error rate for all pixels.

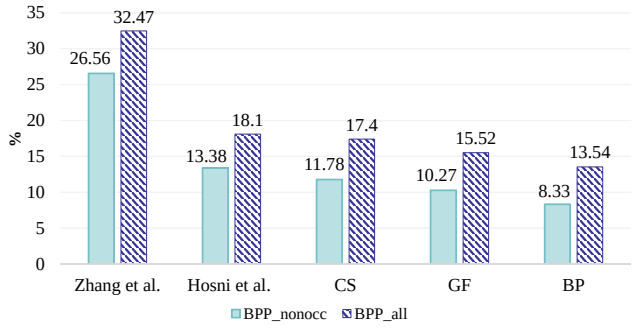


Fig. 2: Average error rates on the Middlebury testing set.

It is clear from the detailed comparison between the two designs showed in Fig. 3 that our method outperforms its reference for all input images selected from the Middlebury test set. The accuracy of the two methods is similar only in cases in which the input images are highly textured, such as Cloth1, Cloth3, Cloth4. Otherwise, our CS method produces much better results. For example, for Flowerpots, CS has the error rate of 20.43% while that of Zhang et al. [27] is 50.36%

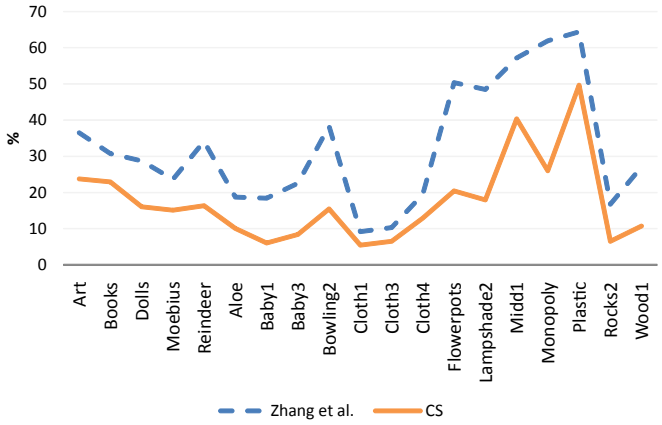


Fig. 3: Detailed error rates of CS and its reference method [27] on the Middlebury testing set.

Similarly, our GF architecture produces more accurate results than its reference method [8]. On average, the proposed GF method has BPP_nonocc and BPP_all lower by 3.11% and 2.52% respectively. The detailed accuracy comparison between the two methods on each input stereo pair is shown in Fig. 4. Our method gains if the input images contain large texture-less regions, as in Lampshade2, Monopoly and Plastic examples. If this is not the case, both implementations perform on par.

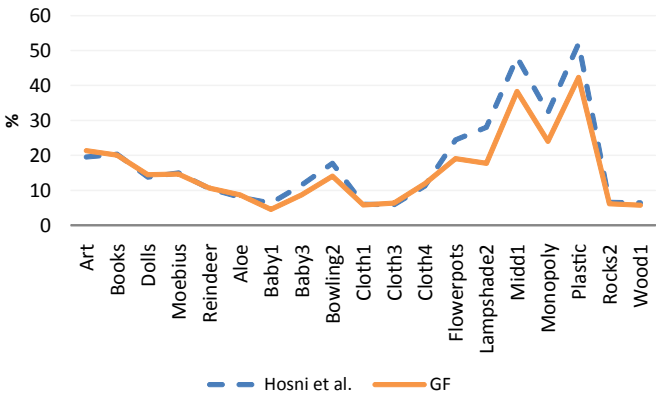


Fig. 4: Detailed error rates of GF and its reference method [8] on the Middlebury testing set.

It can be seen that both proposed local architectures outperform the reference designs when applied to inputs containing lower amounts of textures. This is mainly caused by using Census for matching cost calculation, which performs better than the truncated absolute difference or its fusion with the gradient (as in [27] and [8]) in such regions. Among the three methods, the BP method has the highest accuracy, with error rates of only 8.33 % and 13.54 % for non-occluded and for all pixels respectively. Sample results of all three methods applied to two input pairs are shown in Fig.7a. The first image pair is highly textured while the other has large texture-less regions. On these example, it can be seen that the prototype methods perform much better than the reference methods in the texture-less regions.

KITTI dataset

The error rates computed for the KITTI dataset are shown in Fig. 5. It is clear that all proposed methods outperform the baseline implementations when applied to this data. Among the three proposed systems, BP performs once again the best. The proposed CS scheme performs the worst, but it is important to note that the gain of our modified algorithm is the largest when compared against its reference design.

The same pattern can be observed in Fig. 6, in which the average disparity error is presented. This result is very meaningful, as it shows not only the number of wrongly measured disparity values, but also their magnitudes. The CS method significantly improves its reference method, while the BP method is still the best among the prototype methods with an average error rate of 1.3 pixels for non-occluded regions and 1.6 pixels for all regions. Finally, we note that the proposed GF method is ranked among the best local methods on the KITTI benchmark.

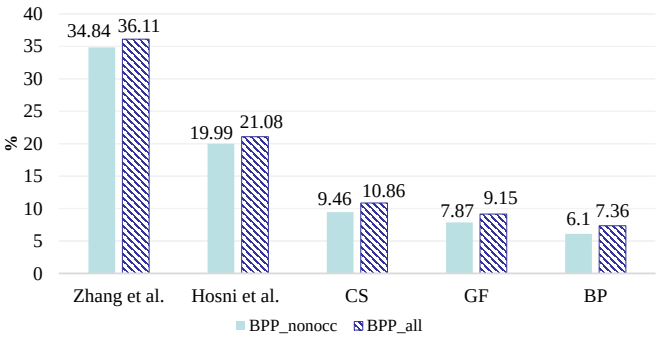


Fig. 5: Average error rates on the KITTI testing set.

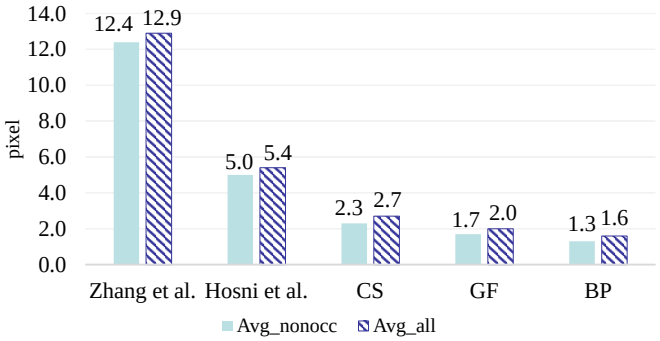


Fig. 6: Average disparity errors on the KITTI testing set.

Saygili et al. also constructed an architecture using their adaptive cost fusion strategy with the same disparity computation as the GF method [15]. They reported average error rates of 11.2% and 13.1% for non-occluded and all regions on the KITTI dataset with that architecture, which are much higher than those obtained with the proposed GF method.

The disparity maps generated by our architectures are more accurate and smoother than those generated by their reference methods, as can be observed in Fig.7b, in which the results of all methods on the sample input from this dataset are shown.

MPEG FTV dataset

The test results on Middlebury and KITTI datasets showed that BP performs the best among the three proposed methods. As a result, we selected the BP system to conduct the experiments on the MPEG FTV input. The objective quality of synthesized images measured as peak signal to noise ratio (PSNR) of images synthesized using the depth maps generated by our BP implementation and MPEG's DERS software are summarized in Table 2.

Table 2: PSNR values of BP method and the DERS software.

	BP method	DERS
Balloons	38.15 (dB)	39.10 (dB)
Newspaper	35.60 (dB)	35.48 (dB)

It can be seen that the proposed BP method performs poorer than DERS on Balloons and slightly better in case of the Newspaper sequence. However, it should be noted that DERS uses as input three views at the same time whereas our BP method uses only two. DERS also has a mode in which manually created data is added as an input to help the algorithms.

An example of an image synthesized using disparity maps generated by our BP method is illustrated in Fig.7c. In this example, the erroneous areas in synthesized image are highlighted. Overall the synthesized image is visually acceptable when compared to the original image.

4.3. Discussion

It can be clearly seen from the experimental results that the proposed CS and GF local stereo matching architectures outperform their reference algorithms. Although we do not have a direct baseline to evaluate our BP method, conducted analysis proved that even a simple implementation of the global algorithm can achieve the best results among the three methods. Moreover, disparity maps calculated using the proposed BP scheme are comparable to those generated using the state-of-the-art DERS system on MPEG FTV data.

As experienced during the training procedure, the parameters of all the three prototype methods do not significantly differ between various inputs. For example, the GF method produced the best results on both Middlebury and KITTI datasets using the same guided filter radius. The same observation can be made for the BP method. This shows that the prototype methods built from our framework are robust under changing operating conditions.

We also conclude that, although more advanced and complex architectures have been proposed in the literature, we can build reliable systems by maximizing the powers of the well-known simple techniques thanks to the proper design of the whole system.

It is important to note that drawing conclusions about the good performance of the local methods based only on results on the Middlebury dataset may not be valid. Our experiments proved the superior performance of global algorithms when compared to local matching techniques. This coincides with the conclusion drawn by the authors of the KITTI dataset [4], who stated that the standard Middlebury dataset may no longer be a good and fair dataset for stereo matching benchmarking.

5. CONCLUSIONS

This paper presents a simple yet effective framework for building stereo matching algorithms employing the census matching cost, discontinuity-preserving disparity computation and view-consistent disparity refinement. With this framework, we aimed to create computationally efficient yet robust and accurate stereo matching systems. Three prototype methods

were built, employing three well-known discontinuity-preserving disparity computation methods of both local and global categories.

All proposed architectures were thoroughly evaluated using three different datasets captured under different conditions. The experimental results showed that the proposed approaches bring significant gains relative to their reference techniques. Moreover, our implementations proved to achieve better results for any input. In particular, the proposed methods outperformed their reference designs, improving the results for up to 25% of pixels. This demonstrates the effectiveness of the proposed framework for designing practical stereo matching systems.

6. REFERENCES

[1] D. Chen, M. Ardabilian, and L. Chen. A novel trilateral filter based adaptive support weight method for stereo matching. In *PROC. BMVC*, 2013.

[2] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002.

[3] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient belief propagation for early vision. In *PROC. CVPR*, pages 261–268, 2004.

[4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *PROC. CVPR*, pages 3354–3361, 2012.

[5] S. Grauer-Gray, C. Kambhamettu, and K. Palaniappan. GPU implementation of belief propagation using cuda for cloud tracking and reconstruction. In *PROC. PRRS*, pages 1–4, 2008.

[6] K. He, J. Sun, and X. Tang. Guided image filtering. In *PROC. ECCV*, pages 1–14, 2010.

[7] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(9):1582–1599, 2009.

[8] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(2):504–511, 2013.

[9] O. Stankiewicz K. Wegner. DERS software manual. MPEG2014/M34302, July 2014.

[10] T. F. Marshall and F. T. William. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *PROC. ICCV*, pages 900–906, 2003.

[11] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang. Segment-tree based cost aggregation for stereo matching. In *PROC. CVPR*, pages 313–320, 2013.

[12] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *ICCV Workshops*, pages 467–474, 2011.

[13] M.G. Mozerov and J. van de Weijer. Accurate stereo matching by two-step energy minimization. *IEEE Trans. Image Process.*, 24(3):1153–1163, 2015.

[14] M-G. Park and K-J. Yoon. Leveraging stereo matching with learning-based confidence measures. In *PROC. CVPR*, 2015.

[15] G. Saygili, L. van der Maaten, and E.A. Hendriks. Adaptive stereo similarity fusion using confidence measures. *Comput. Vis. Image. Und.*, 135:95 – 108, 2015.

[16] D. Scharstein, H. Hirschmller, Y. Kitajima, G. Krathwohl, N. Nei, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *PROC. GCPR*, pages 31–42. 2014.

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

[18] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *PROC. CVPR*, pages 195–202, 2003.

[19] J. Sun, N-N. Zheng, and H-Y. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25:787–800, 2003.

[20] O. Veksler Y. Boykov and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[21] C. Lee Y-S. Ho, E-K. Lee. Multiview video test sequence and camera parameters. MPEG2008/M15419 document, 2008.

[22] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *PROC. ECCV*, pages 756–771. 2014.

[23] Q. Yang. Stereo matching using tree filtering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(4):834–846, 2015.

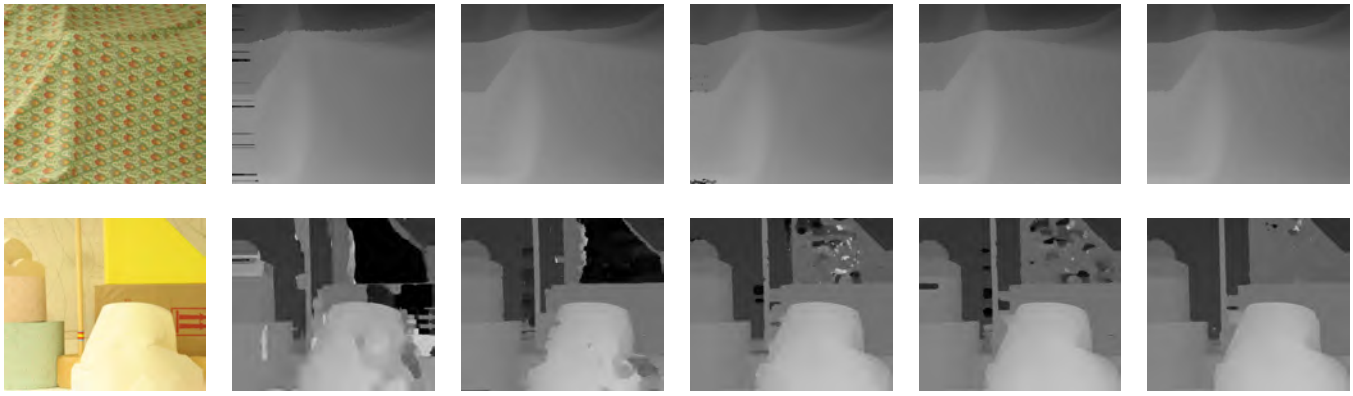
[24] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(3):492–504, 2009.

[25] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *PROC. ECCV*, pages 151–158, 1994.

[26] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *PROC. CVPR*, 2015.

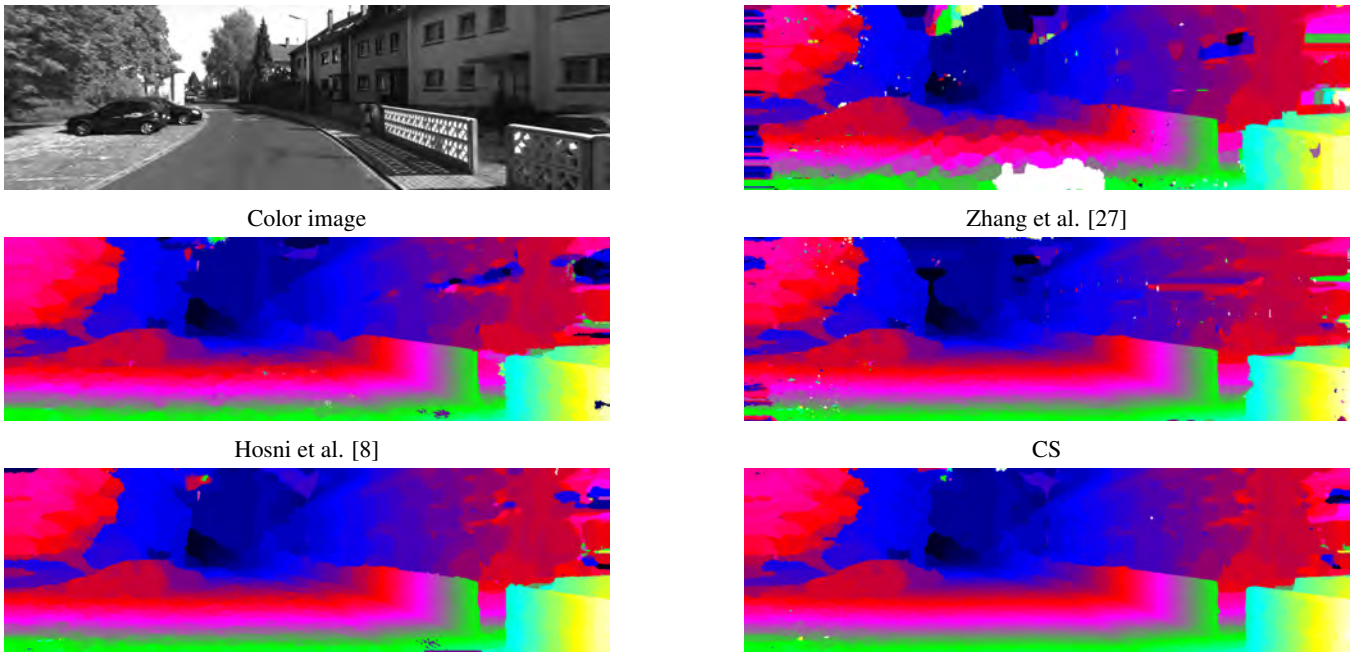
[27] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE Trans. Circuits Syst. Video Technol.*, 19(7):1073–1079, 2009.

[28] Q. Zhang, L. Xu, and J. Jia. 100+ times faster weighted median filter (wmf). In *PROC. CVPR*, pages 2830–2837, June 2014.



Color image | Zhange et al. [27] | Hosni et al. [8] | CS | GF | BP

(a) Middlebury dataset.



Color image | Zhang et al. [27]

Hosni et al. [8] | CS

GF | BP

(b) KITTI dataset (Disparity maps are shown as color-coded images).

Synthensized view from BP's result | Original view

(c) MPEG FTV dataset.

Fig. 7: Sample results of the prototype methods and reference methods on the three datasets.