Local Stereo Matching with Improved Matching Cost and Disparity Refinement

*Jianbo Jiao, Ronggang Wang**, *Wenmin Wang, Shengfu Dong, Zhenyu Wang, and Wen Gao*

Digital Media R&D Center, Peking University Shenzhen Graduate School
jianbojiao@sz.pku.edu.cn, {rgwang, wangwenmin, dongsf, wangzhenyu}@pkusz.edu.cn, wgao@pku.edu.cn

看来主要创行点在视差优化上

Abstract

Recent local stereo matching methods have achieved comparable performance with global methods. However, there are still some significant outliers existing in the final disparity map. In this paper, we propose a local stereo matching method that employs a new combined cost and a novel secondary disparity refinement mechanism. The combined cost is formulated by a modified color census transform, truncated absolute differences of color and gradients. Symmetric guided filter is used for the cost aggregation. Different from traditional stereo matching, a novel secondary disparity refinement is proposed to further remove remaining outliers. Experimental results on Middlebury benchmark show that our method ranks the 5^{th} out of the 153 submitted methods, and it is the best cost-volume filtering-based local method. Furthermore, experiments on real-world sequences and depth-based applications also validate the effectiveness of our proposed method.

Index Terms

Local, stereo matching, matching cost, disparity refinement

I. Introduction

Stereo matching is one of the most active research areas in computer vision. It is the process of computing a disparity map given a pair of stereo images. As mentioned in [1], a variety of approaches have been proposed. Most stereo algorithms can be categorized into global and local methods. Global methods usually achieve more accurate disparity map with higher computational complexity, while local methods are more efficient.

一些其他的过滤方法

In recent years, local methods based on adaptive support-weight [2] have achieved results comparable to that of global methods using graph cuts [3] or belief propagation [4]. The main idea of these local methods is to measure the likelihood between center pixel and its neighbor pixels by means of adaptive support weight. A high weight indicates they are likely to be on the same object thus with similar disparities. However, this type of methods involves high computational complexity, and the complexity is related to the window size used for aggregation. Later, Rhemann et al. proposed a new approach [5] for cost aggregation by smoothing cost volume and its complexity is independent of the matching window size. Besides, several other cost-volume filtering-based methods have been developed recently, and achieved good performance. In [6], a hardware-efficient bilateral filter was proposed for fast aggregation. In [7], the domain transform was imported so that the cost aggregation can be performed by using 1-D filters. A recursive bilateral filter [8] was introduced by Yang for aggregation. De-Maeztu et al. presented an O(1) method [9] based on a symmetric filter. For matching cost computation, the most commonly used one is the absolute difference (AD) to calculate the difference between the intensity of corresponding pixels to measure the likelihood. After that, cost measures like SAD (Sum of Absolute Difference) and SSD (Sum of Squared Difference) are also widely used. On the other side, gradient-based measures and non-parametric transforms such as rank and census provide a better description of image structure [10]. In [5], Rhemann et al. combined the AD with gradient and obtained impressive results. Mei et al. presented a new cost measure by combining AD and census to reduce the errors caused by individual measures [10]. Although the performance of stereo matching has been improved, there are still some obvious artifacts in the final results. Much effort is taken on cost aggregation improvement, but far less attention is paid on disparity refinement, neither the cost measurement.

融合AD、梯度和Census
（census经过了改进）

In this paper, we propose two strategies to further improve the performance of local stereo matching. Firstly, a new cost measure by merging truncated absolute difference of color, gradients and a modified color census transform is proposed to improve the initial matching performance. Secondly, after the traditional disparity refinement, we propose a secondary refinement approach, which is called "Remaining Artifacts Detection and Refinement" (RADAR) to further refine the results. By means of RADAR, most of the remaining outliers after traditional post-processing are corrected, and a remarkable improvement is achieved. For cost aggregation, we employ the symmetric guided filter proposed in [5] and [9]. Experimental results on Middlebury benchmark [11] demonstrate the effectiveness of our method, and it is one of the best local stereo matching methods. The performance of our method is the best among cost-volume filtering-based methods on Middlebury dataset. In addition, our method works well on real-world sequences, as well as some depth-based applications.

This paper extends our preliminary work [12] by using an adaptive judging-window to facilitate the RADAR strategy, and performing more experiments to demonstrate the effectiveness of our method. We also provide an in-depth description of the RADAR mechanism and propose a parallel implementation of our method on CPU. Both the proposed cost measure and RADAR are extended to a more common framework for stereo matching, instead of confining to filtering-based method. The remainder of this paper is organized as follows. In section II, the proposed local stereo matching method with new cost measure and RADAR scheme are demonstrated in details. Section III shows the experimental results on Middlebury dataset, real-world sequences, and some depth-based applications. Finally, this paper is concluded in section IV.

II. Proposed Method

This section demonstrates our proposed stereo matching method. First, a cost volume is formulated by our proposed combined cost. Then, a symmetric guided filter is employed for cost-volume filtering. Finally, we propose a RADAR-aided refinement scheme to further improve the accuracy of disparity map. An overview of the whole pipeline and the RADAR scheme are shown in Fig. 1(a) and (b).

A. Combined Matching Cost and Initial Refinement

1) Modified color census transform: Motivated by color census transform [13], we propose a modified color census transform (MCCT) by using a more appropriate method for the census transform (see equation 1). As RGB

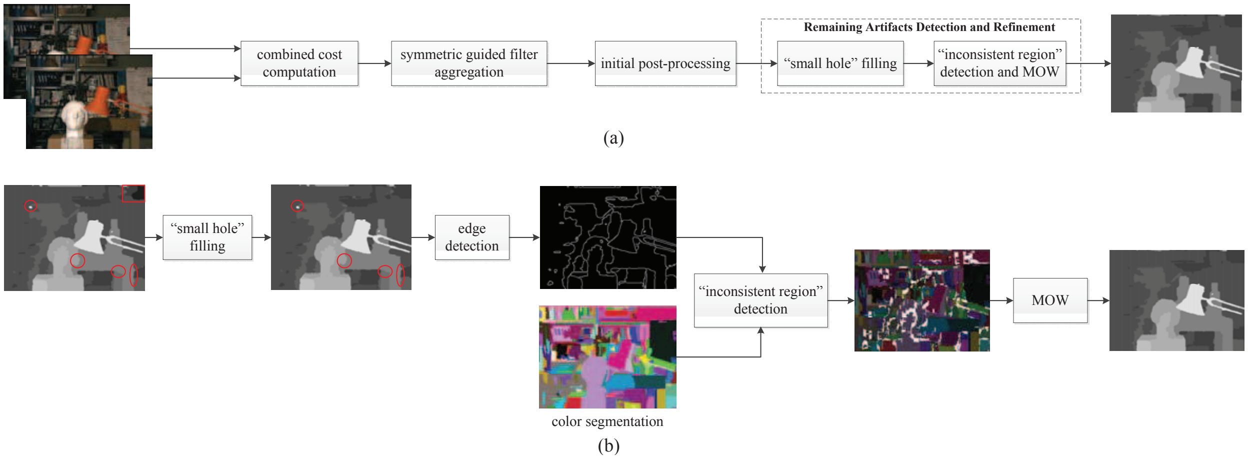

combined cost computation
symmetric guided filter aggregation
initial post-processing
Remaining Artifacts Detection and Refinement
"small hole" filling
"inconsistent region" detection and MOW
(a)
"small hole" filling
edge detection
"inconsistent region" detection
MOW
color segmentation
(b)

Fig. 1. Overview of the proposed method. (a) The whole pipeline. (b) Workflow of RADAR.

color-space is sensitive to radiometric changes, the image is firstly converted to Gaussian color model [14] space. Then the difference between two pixels p and q is measured by the Euclidean distance $D_G(p, q)$, and the mean value of all these distances in the window centered at p is denoted by $D_m(p)$. The MCCT is formulated as follows,

$$MCCT(p) = \bigotimes_{q \in N(p)} \xi(D_m(p), D_G(p, q)) \tag{1}$$

$$\xi(a, b) = \begin{cases} 1, & b < a \\ 0, & otherwise \end{cases} \tag{2}$$

where operator $\otimes$ denotes a bit-wise catenation, and $N(p)$ represents the neighbor pixel set of p. Hamming distance is used to calculate the difference between the two bit-strings generated by MCCT,

$$h(p, d) = Hamming(MCCT_L(p), MCCT_R(p - d)) \tag{3}$$

where d denotes the disparity of two corresponding pixels in left and right images. At last, a robust exponential function is used to normalize the cost,

$$C_{MCCT}(p, d) = 1 - \exp(-\frac{h(p, d)}{\lambda_{MCCT}}) \tag{4}$$

where λ_{MCCT} is a normalizing parameter of 55. Our new MCCT has a better representation of the image structure than the traditional color census [13]. And a comparison of MCCT and traditional color census can be found in our preliminary work [12].

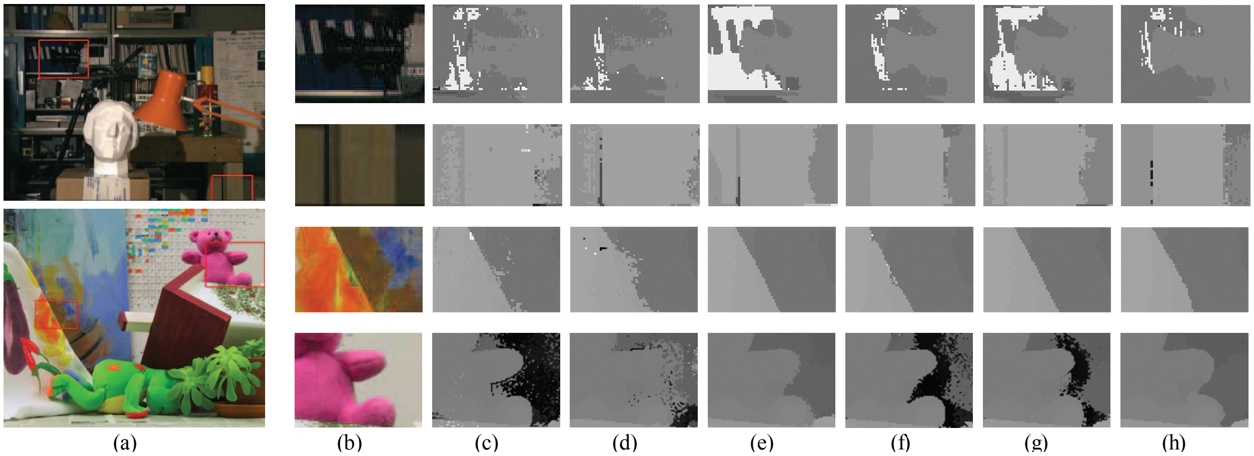


Fig. 2. Cost measure comparison. Top to bottom: repetitive region, dark region, edge, and textureless region. (a) Left images. (b) Close-up of rectangles in (a). (c) to (g): Results of absolute difference (AD), gradient, census, AD+gradient, and AD+census. (h) Results of the proposed combined cost.

2) Combined matching cost: In addition to MCCT, we add two other cost components of truncated absolute differences of color and gradients, which are calculated respectively as follows,

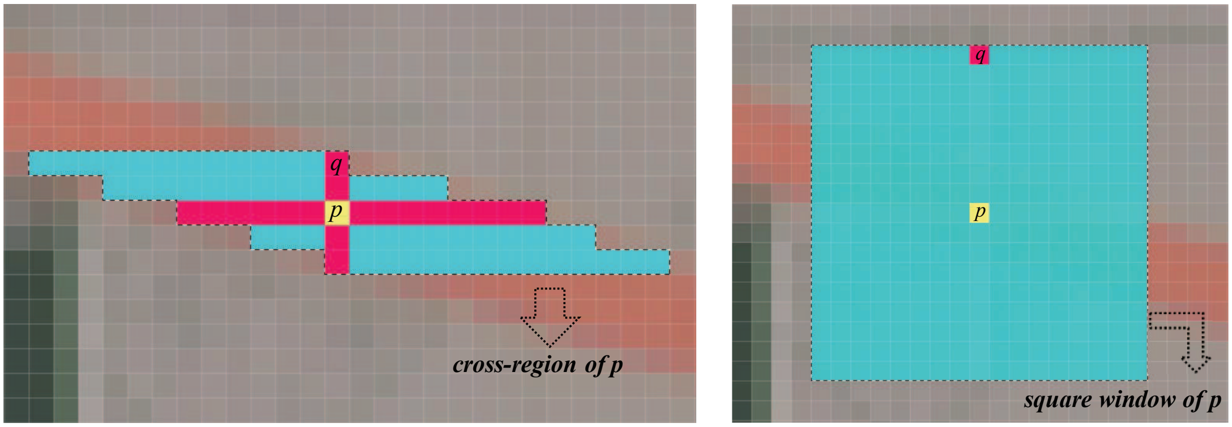
$$\begin{aligned} C_{ADc}(p,d) &= \min(\tfrac{1}{3}\sum_{i=R,G,B} \left\| I_i^L(p) - I_i^R(p-d) \right\|, \lambda_{ADc}), \\ C_{GDx}(p,d) &= \min(\left\| \nabla_x I_L(p) - \nabla_x I_R(p-d) \right\|, \lambda_{GD}) \end{aligned} \tag{5}$$

where λ_{ADc} and λ_{GD} are the truncated values [5], and ∇_x is the derivative in x direction. The gradient in y direction is also employed, denoted as C_{GDy}. The final combined matching cost is formulated by merging the above mentioned four cost components,

$$\begin{aligned} C(x,y,d) = {} & \alpha \cdot C_{MCCT} + \beta \cdot C_{ADc} + \gamma \cdot C_{GDy} \\ & + (1 - \alpha - \beta - \gamma) \cdot C_{GDx} \end{aligned} \tag{6}$$

where α, β, γ are weights for different cost components, adjusting the four components' contributions to the total cost. Fig. 2 gives the comparison between the proposed combined cost and individual costs, as well as some other combined costs. All of the disparity maps are initial stereo matching results without any post-processing, and the cost aggregation strategy is the same for all these tests.

3) Symmetric guided filter aggregation: The combined cost for each pixel at each disparity level is stored in a cost volume. In order to preserve both edges in reference and target images, we employ the symmetric guided filter proposed in [5] and [9] for the cost aggregation. After the cost volume is aggregated by symmetric guided filter, the "winner-takes-all" strategy is used for disparity selection, i.e., selecting the disparity label with the lowest cost.



(a)

(b)

Fig. 3. Support region of pixel p. (a) Cross-region. (b) Traditional square window.

Then the initial disparity map is generated.

4) Initial post-processing: There are still many outliers in the initial disparity map. In order to find the inconsistent pixels in left and right images, the left-right consistency check (LRC) is employed. A pixel p is labeled as outlier if it violates the following constraint,

$$|d_{\mathrm{L}}(p) - d_R(p - d_L(p))| < 1 \tag{7}$$

where d_L, d_R are the disparities of the corresponding pixels in left and right images respectively.

Once the outliers are detected, we use a cross-region based voting technique [10] to correct them. The voting operation is done iteratively to be more robust. The cross region of a pixel p is shown in Fig. 3(a). More details about the voting method can be found in [10].

After the cross-region voting, we use the nearest reliable pixel in the scan-line to update the remaining outliers labeled by LRC. A weighted median filter with bilateral filter weights [5] is employed to remove the streak-like artifacts.

B. Remaining Artifacts Detection and Refinement

Some error regions still exist after the initial post-processing. If these artifacts exist in both left and right images, they can-not be detected by just employing the LRC. Thus a secondary refinement scheme named "Remaining Artifacts Detection and Refinement" (RADAR) is proposed in this paper. Fig 1(b) gives a pipeline of RADAR.

1) "Small hole" filling: According to our observation, remaining artifacts are mainly composed of some "small holes" and outliers around object boundary, as shown in Fig. 4. "Small holes" are dark regions with disparities much

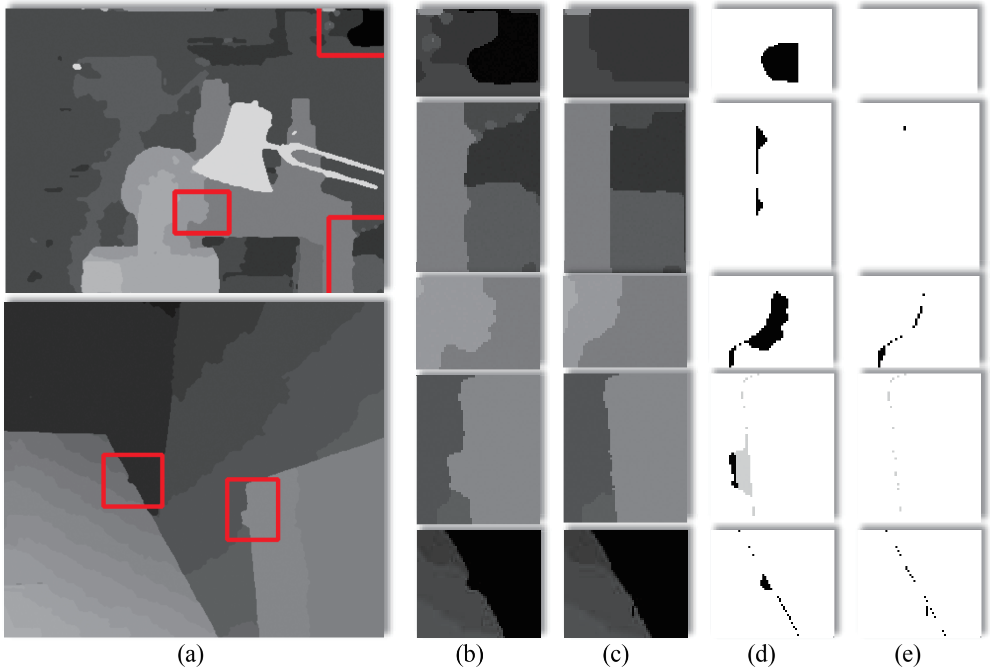


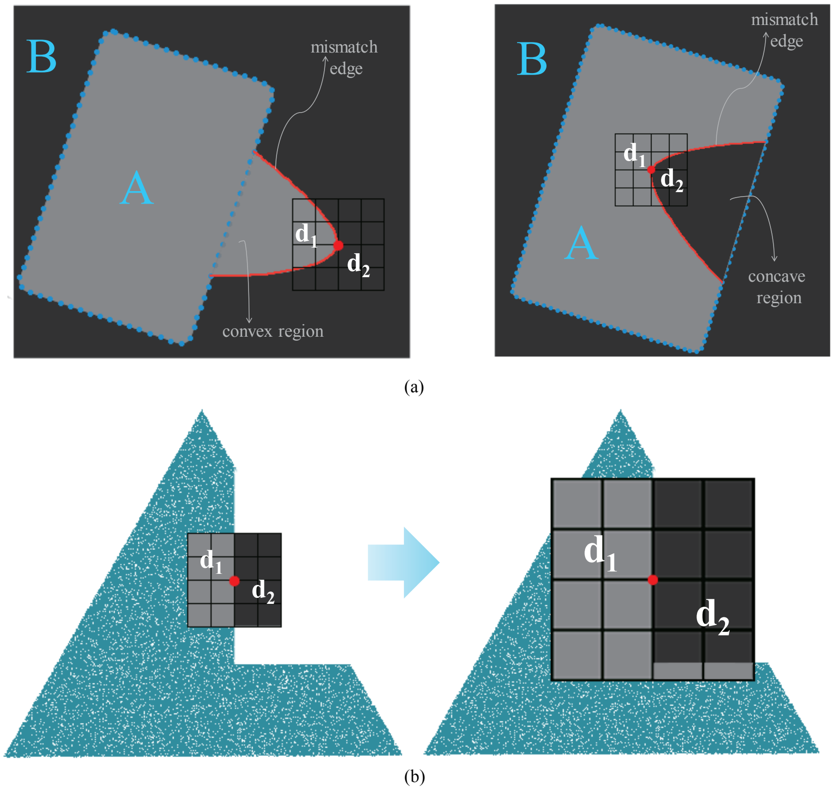
Fig. 4. “Remaining artifacts” before and after RADAR, the first row represents “small hole”; the 2^{nd} to 4^{th} rows represent convex regions; the last row represents concave region. (a) Disparity maps before RADAR. (b) Close-up of rectangles in (a). (c) Results after RADAR. (d) Error maps of (b). (e) Error maps of (c).

smaller than their neighbors. They can be detected by comparing disparities with their neighbors. After detecting hole-pixel, we use the most appropriate disparity in its neighborhood (both horizontal and vertical) to update it. Commonly, the hole-pixel p is updated by the pixel with smaller disparity (background pixel), but if the pixel on the smaller disparity side of p is also invalid (hole-pixel), it should be updated by the pixel on the other side, as shown in follows,

$$\begin{aligned} d_p^* &= \begin{cases} \max\{d_p^{'}, d_p^{''}\}, & d_p^{'} \cdot d_p^{''} \le d_{thres}^2 \\ \min\{d_p^{'}, d_p^{''}\}, & d_p^{'} \cdot d_p^{''} > d_{thres}^2 \end{cases}, \\ d_{thres} &= \rho \cdot d_{\max} \end{aligned} \tag{8}$$

where $d_p^{'}$ and $d_p^{''}$ are the nearest (taking one direction as an example) pixels’ disparities larger than d_{thres}, and $d_{\max}$ is the maximum disparity, while ρ is an empirical penalty of 1/7. The updated disparity is denoted as d_p^*.

2) “Inconsistent region” detection: As shown in Fig. 4, the other type of artifacts is composed of outliers around object boundary. We name these artifacts as “inconsistent regions”, which consist of convex regions (same as “fattening” region [1]) and concave regions. The inconsistent region is detected by checking whether the edges of disparity map coincide with the boundaries of objects in the scene. Canny edge detector [15] is used to extract the



(a)

(b)

Fig. 5. Inconsistent region detection. (a) Decision on convex and concave regions. (b) Adaptive judging-window.

disparity edges. A mean-shift [16] based color segmentation approach is utilized to detect the objects' boundaries. Beforehand, a contrast enhancement operation (histogram equalization on the luminance part of the color image) is performed, and then the image is converted to CIELab space. With this kind of preprocessing, segmentation accuracy is improved, especially on dark regions. If an edge in disparity map does not exactly coincide with the object boundary in the scene, it is labeled as a mismatched edge. Convex regions lie on the foreground side of the mismatched edge, while concave regions on the background side. Fig. 5 shows the way to decide which side is the inconsistent region. As is shown in Fig. 5(a), the inconsistent region is labeled by means of a judging-window (the black grid). The region A circled by blue dotted line represents foreground object, and the rest region B is background. The red line is the detected mismatched edge. At every point (red point) on the mismatched edge a judging-window is formed, which is divided into two areas: foreground area (larger disparity d_1) and background area (smaller disparity d_2). Thus the inconsistent region is labeled on the smaller area side of mismatched edge, e.g., d_1 side on the left image and d_2 side on the right one. The inconsistent (convex or concave) region in Fig. 5(a) is the region encircled by the blue and red lines. However, when the area of d_1 is equal to the area of d_2 (shown in the left image of Fig. 5(b)), the judgment of the inconsistent side would get into a dilemma. Once it occurs, the judging-window becomes size is increased until the size of the two areas are different (shown in the right image of Fig. 5(b)).

3) Modified OccWeight: In order to correct the inconsistent regions detected above, we propose a modified OccWeight (MOW) based on the OccWeight presented in [17]. The OccWeight method corrects a pixel's disparity by choosing the most likely one in a fixed square window around it. However, a squared window (Fig. 3(b)) is not robust. Hence we replace the square window with a cross window as shown in Fig. 3(a). The cross window is consistent with image structure, and provides a much more accurate support region. In addition, the disparity inheritance [17] is also adopted. In a cross window of pixel p, the weight of its neighboring pixel (q) is calculated as follows,

$$w(p,q) = \begin{cases} \exp(-\frac{\Delta c_{pq}}{\phi_c} - \frac{\Delta s_{pq}}{\phi_s}), & if\ q \notin R_I \\ 0 & , otherwise \end{cases} \tag{9}$$

where Δc_{pq} and Δs_{pq} are the color distance and spatial distance between p and q, ϕ_c and ϕ_s are parameters used to normalize the color and spatial distances, respectively [17]. R_I is the set of inconsistent regions. The updated disparity is calculated as,

$$\begin{aligned} d^*(p) &= \arg\max_{d \in D} \Big(\sum_{q \in AW_p} w(p,q) \times m(q,d) \Big), \\ m(q,d) &= \begin{cases} 1, & if\ d(q) = d \\ 0, & otherwise \end{cases} \end{aligned} \tag{10}$$

where D is the set of disparity candidates, and AW_p is the set of pixels in the cross window of p. By employing the MOW, outliers at inconsistent regions are corrected (last image in Fig. 1(b)). Finally, we smooth the disparity map with a median filter to remove the remaining noises. The pseudo code of the RADAR algorithm is given in Algorithm 1.

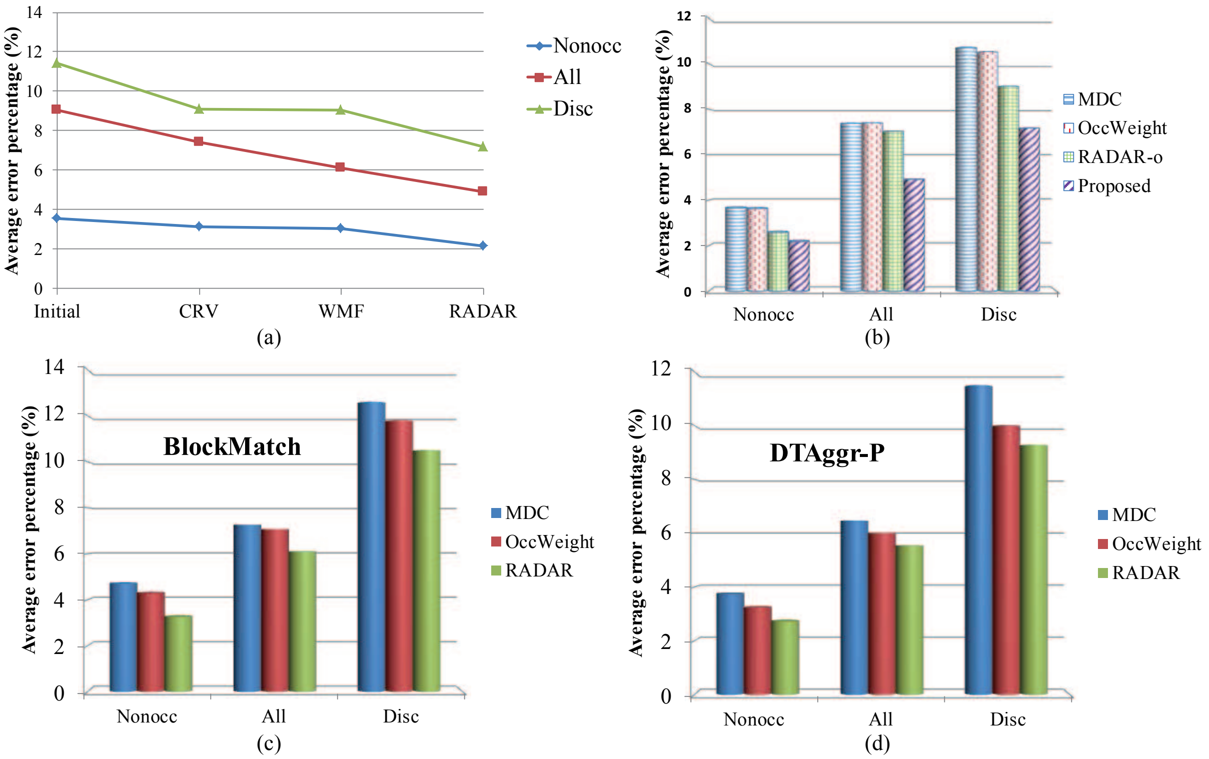
In order to evaluate the effectiveness of our proposed RADAR-aided refinement pipeline, we test it on the Middlebury dataset (*Tsukuba*, *Venus*, *Teddy*, and *Cones*) [11], and use the evaluation measures of "Nonocc", "All", "Disc", representing the non-occluded regions, all regions, and regions near discontinuities, respectively. For each measure, the average value of the four images in the dataset is calculated. Fig. 6(a) demonstrates the effectiveness of each step in the proposed refinement pipeline. "CRV" and "WMF" represent the cross-region voting and weighted median filter respectively. As shown, the error decreases after each step of our pipeline. In addition, we compare the RADAR-aided disparity refinement with the fattening region refinement proposed in [18] (denoted as MDC) and the referenced method of OccWeight [17], as shown in Fig. 6(b). For MDC, we use the fattening detection method in [18], while for OccWeight, the region-detection method in this paper is employed. Furthermore, the RADAR-only (RADAR-o) item without the initial post-processing is also evaluated. All of these methods are based

Algorithm 1 Remaining Artifacts Detection and Refinement (RADAR)

Input: disparity map D (disparity of p is d_p) and color image I

Output: refined disparity map D^*

```
 1: ————————————“small hole” filling————————————
 2: $d_{thres} \leftarrow \rho \cdot d_{\max}$
 3: $d_p^{'}, d_p^{''} \in \{$dispairty of the nearest pixels to $p$ $\}$
 4: if $d_p^{'} \cdot d_p^{''} > d_{thres}^2$ then
 5:     $d_p^* \leftarrow min\{d_p^{'}, d_p^{''}\}$
 6: else
 7:     $d_p^* \leftarrow max\{d_p^{'}, d_p^{''}\}$
 8: end if
 9: ——————inconsistent region detection (See Fig. 5)——————
10: $E^* \leftarrow$ Canny edge detection on disparity map
11: $I^* \leftarrow$ contrast enhancement on color image $I$
12: $S^* \leftarrow$segmentation on $I^*$ in CIELab space
13: for each edge $e$ in $E^*$, each segment $s$ in $S^*$ do
14:     if $e$ through $s$ then
15:         mismatch edge $\leftarrow e$
16:     end if
17: end for
18: for each point on mismatch edge do
19:     if area $d_1 <$ area $d_2$ then
20:         $R_I \leftarrow$ area $d_1$
21:     else if area $d_1 >$ area $d_2$ then
22:         $R_I \leftarrow$ area $d_2$
23:     else
24:         while area $d_1$ == area $d_2$ do
25:             increase the size of judging-window
26:         end while
27:         goto 19
28:     end if
29: end for
30: ————————————————MOW————————————————
31: for each error pixel $p$ in $R_I$ do
32:     $AW_p \leftarrow$ adaptive window of $p$
33:     for $q \in AW_p$ do
34:         $w(p,q) \leftarrow$ weight of $p, q$
35:     end for
36:     $d^*(p) \leftarrow \arg\max_{d \in D}(\sum_{q \in AW_p} w(p,q) \times m(q,d))$
37: end for
```



(a) (b) (c) (d)

Fig. 6. (a) Improvements of each step in the refinement pipeline. (b) Comparison with other approaches. (c) Comparisons on BlockMatch. (d) Comparisons on DTAggr-P.

on the same initial disparity map, i.e., based on our proposed combined cost and aggregation. As shown in Fig.6(b), the refinement pipeline proposed in this paper performs the best among all these methods. In order to validate the universality of the RADAR scheme, comparisons under the platform of other local methods are also performed as shown in Fig. 6(c) and (d). BlockMatch and DTAggr-P [7] are chosen for the comparison, their disparity refinement steps are replaced by the methods to be compared. Our RADAR scheme still obtains the best performance under these local methods.

III. Experimental Results

This section shows an experimental evaluation of our proposed method. Here, the evaluation is performed over two test sets. One is the Middlebury dataset [11], while the other is some typical real-world sequences. Applications in 3D reconstruction and virtual view synthesis are also shown. The parameters used in the experiment are chosen empirically and kept constant as $\{\alpha, \beta, \gamma\} = \{0.011, 0.15, 0.1\}$. These parameters are obtained based on 35 images with ground truth disparity maps on Middlebury datasets according to an optimization process aiming at obtaining

TABLE I
QUANTITATIVE EVALUATION OF THE PROPOSED METHOD COMPARED WITH OTHER LOCAL METHODS ON MIDDLEBURY.

Algorithm	Rank	*Tsukuba*			*Venus*			*Teddy*			*Cones*			Rank*
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
ADCensus[10]	2	1.07	1.48	5.73	0.09	0.25	1.15	4.10	6.22	10.9	2.42	7.25	6.95	28
Proposed	5	1.17	1.48	6.35	0.14	0.29	1.83	5.39	10.6	14.7	2.01	7.37	5.88	13
HEBF [6]	31	1.10	1.38	5.74	0.22	0.33	2.41	6.54	11.8	15.2	2.78	9.28	8.10	24
DTAggr-P [7]	39	1.75	2.10	7.09	0.24	0.45	2.59	5.70	11.5	13.9	2.49	7.82	7.30	33
CostFilter [5]	40	1.51	1.85	7.61	0.20	0.39	2.42	6.16	11.8	16.0	2.71	8.24	7.66	19
P-LinearS [9]	53	1.10	1.67	5.92	0.53	0.89	5.71	6.69	12.0	15.9	2.60	8.44	6.71	53
RecursiveBF [8]	68	1.85	2.51	7.45	0.35	0.88	3.01	6.28	12.1	14.3	2.80	8.91	7.79	36

the minimum average error percentage.

A. Middlebury Dataset

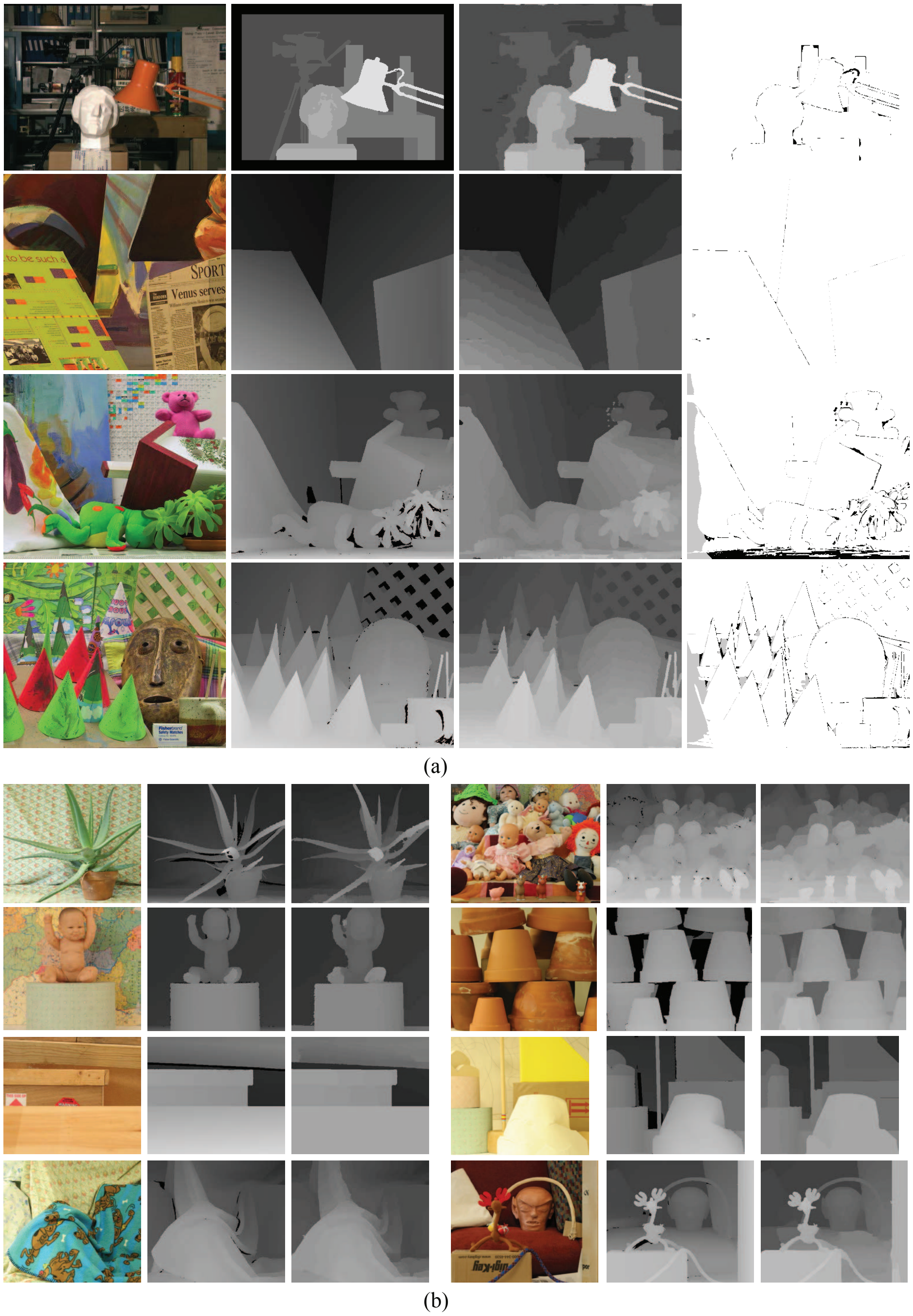
The experimental results on four test images (*Tsukuba*, *Venus*, *Teddy*, and *Cones*) from Middlebury online benchmark [11] of our proposed method are shown in Fig. 7(a). Our proposed method obtains competitive performance with the state-of-the-art methods, and ranks the 5^{th} out of 153 methods by the time we submit. To the best of our knowledge, our method is the top one of cost-volume filtering-based local methods.

Our method is also compared with some other filtering-based local methods and “ADCensus” [10] (the top local method) on Middlebury. The comparison results are listed in Table 1, and error percentages in different regions for the four images are presented. Error threshold is set to the default 1.0. Meanwhile, sub-pixel threshold 0.75 is also chosen, and the rank on it can be seen in the last column (Rank*) of Table I.

From Table I, when error threshold is 1.0, the method proposed in this paper is the best cost-volume filtering-based method, and the second best local method (poor than “ADCensus” [10]). However, when errors are evaluated at sub-pixel level (0.75), our method performs the best in the selected methods. Sub-pixel evaluation means the disparity can be a floating number, instead of being limited to integer, and it is useful in practical applications. But its worth noting that our method has no regard of sub-pixel, which means all of the disparities are estimated at integer level. In sub-pixel level, the performance of our method only has a slight decline (rank from the 5^{th} to the 13^{th}), which proves the robustness of our proposed method. Experimental results on other images in Middlebury dataset are shown in Fig. 7(b).

B. Real-World Dataset

In this experiment, we choose four real-world sequences as the test set: the *BookArrival* sequence from HHI 3D video database, the *Balloons* sequence from FTV, *Cafe* and *Newspaper* sequences obtained from GIST. For each test sequence, we randomly extract a frame with its corresponding view as test image pair. Besides, some competitive



(a)

(b)

Fig. 7. Experimental results on Middlebury dataset. (a) Top-to-bottom: *Tsukuba*, *Venus*, *Teddy*, *Cones*, respectively; left-to-right: color images, ground truth, results of our method, error maps with threshold equals 1.0. (b) Other results. Left-to-right: color images, ground truth, and results of our method.

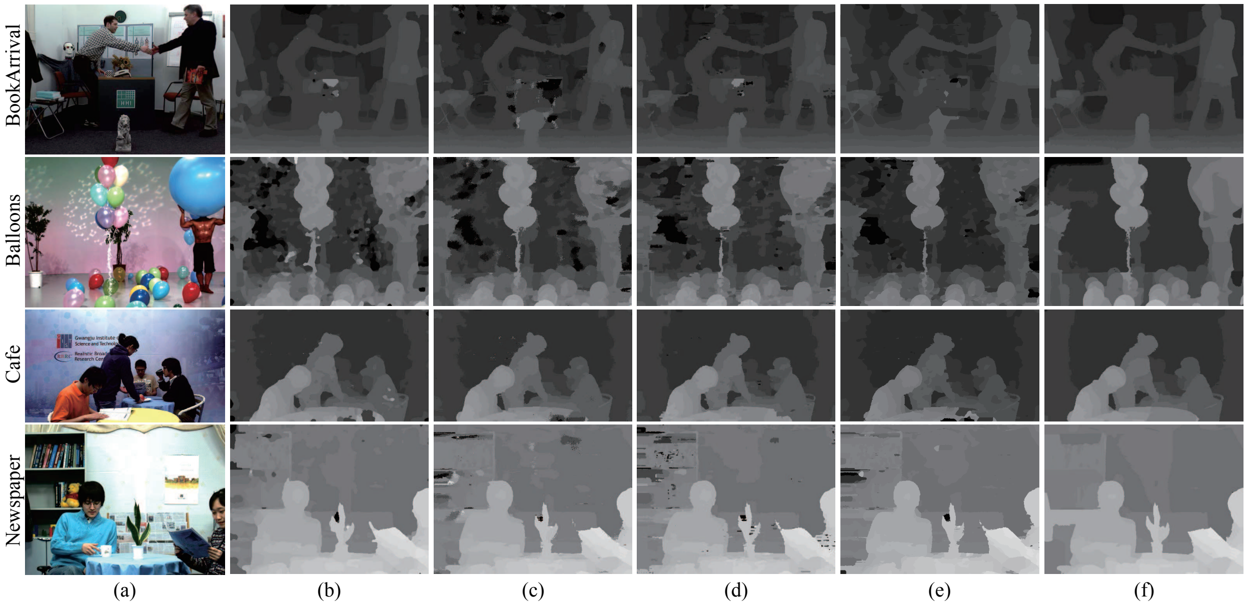


Fig. 8. Experimental results of our method compared with competitive algorithms on real-world sequences. (a) Left frames. From (b) to (e): Results of RecursiveBF, CostFilter, DTAggr-P, and HEBF. (f) Results of proposed method.

filtering-based methods mentioned above are selected for the comparison, which are RecursiveBF [8], CostFilter [5], DTAggr-P [7], and HEBF [6]. The parameter settings of these methods are the same as recommended in these papers. The visual results are shown in Fig. 8.

From the visual results we can see that, our method has the best edge-preserving performance, such as the lion in *BookArrival* sequence and objects in *Balloons* sequence. In addition, our disparity results perform well on image borders, e.g., the coat on the left border of *BookArrival* and Newspaper sequences, which is important in practical applications, such as virtual view synthesis and 3D reconstruction. The experimental results on real-word sequences again prove the effectiveness of our proposed method.

C. 3D Reconstruction and View Synthesis

In order to verify the effectiveness of our proposed method, the three dimensional reconstruction and virtual view synthesis results are generated based on the disparity map calculated by our method. In Fig.9(a), the 3D views of *Tsukuba* and *Teddy* reconstructed by the disparity maps generated by our method, CostFilter [5] and ADCensus [10] are shown. For the red rectangle regions, our method performs better than the other two methods.

Another application is the depth-based virtual view synthesis. Here, the common used reference software VSRS from MPEG is utilized to synthesize virtual view at the middle of the two reference views. Sequences *Balloons* and

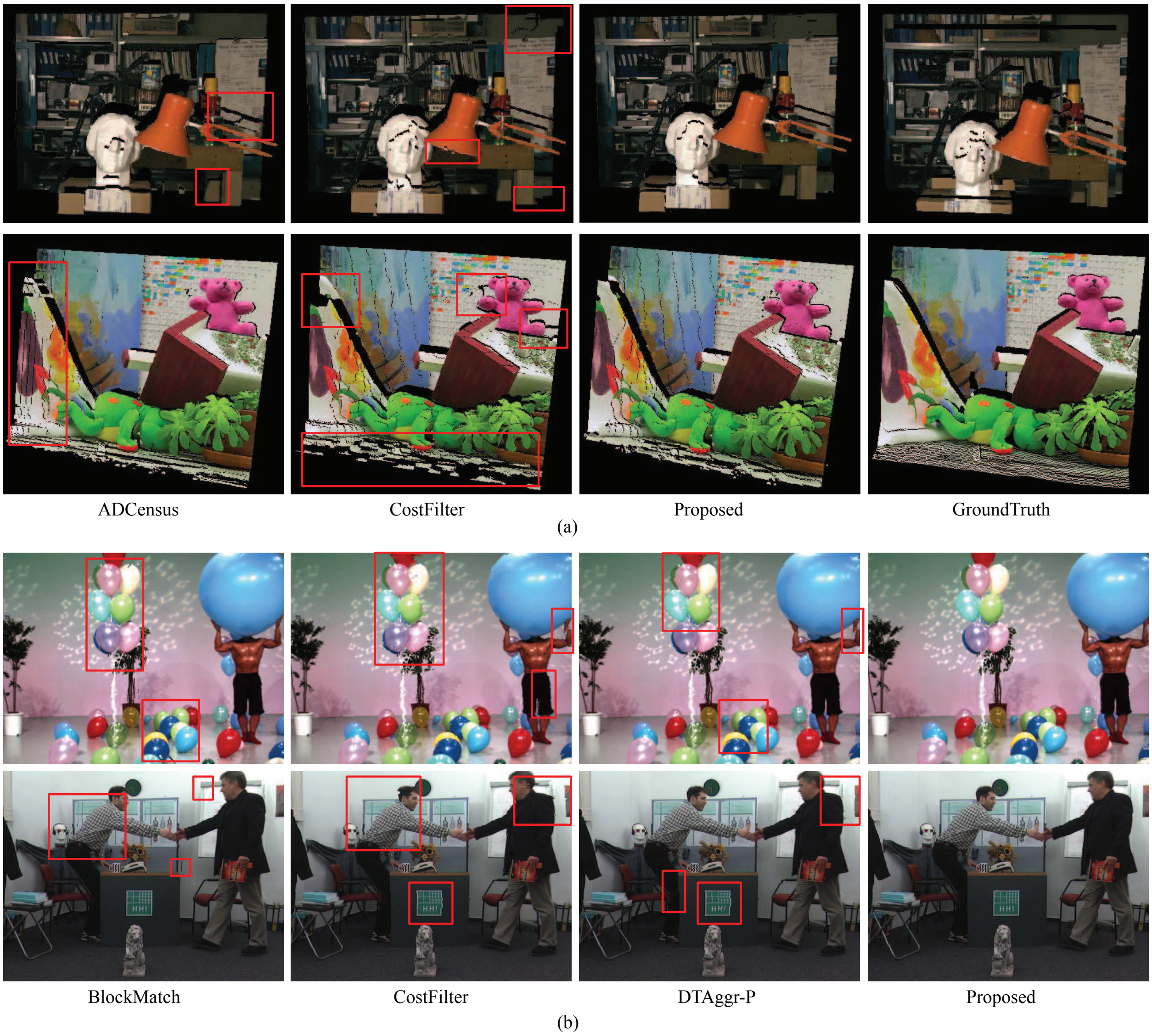


Fig. 9. (a) 3D reconstruction of *Tsukuba* and *Teddy*. (b) Virtual view synthesis on *Balloons* and *BookArrival*.

BookArrival are chosen for the test. Results based on the disparity maps generated by Block Matching, CostFilter [5], DTAggr-P [7], and our proposed method are shown in Fig. 9(b). As shown, there are many artifacts in the virtual view synthesized by BlockMatch, CostFilter, and DTAggr-P, while there is no artifact in the synthesized view by our method, which validate the effectiveness of our proposed stereo matching method.

D. Computational Complexity

The computational complexity of our proposed stereo matching method is determined by the following four steps: 1) combined cost computation, 2) guided filter aggregation, 3) initial refinement, and 4) the RADAR. Let N be the number of image pixels, and D be the number of disparity levels. The complexity of the combined cost computation is $\mathcal{O}(ND)$, the complexity of the cost aggregation is $\mathcal{O}(1)$ [9] given the integral image and the complexity of computing integral image is $\mathcal{O}(N)$. The complexity of initial disparity refinement is $\mathcal{O}(N)$. In RADAR, "small hole" filling and the MOW can be done in $\mathcal{O}(N)$ and $\mathcal{O}(\varepsilon)$, where ε is the number of error pixels. For Canny edge detector, the complexity is $\mathcal{O}(NlogN)$. The color segmentation in RADAR can be implemented in $\mathcal{O}(NlogN)$ or even $\mathcal{O}(N)$ with the help of KD-tree or integral image. Therefore, the overall complexity of the proposed method is $\mathcal{O}(N + ND + NlogN + \varepsilon)$. As D and ε are small constants for practical applications, the complexity can be approximately equal to $\mathcal{O}(NlogN)$. In addition, most parts of the method can be implemented in parallel. Thus the proposed method is applicable for high-resolution images or video application.

For Middlebury dataset, the dimensions of images *Tsukuba, Venus, Teddy*, and *Cones* are 384×288, 434×383, 450×375, and 450×375 with the disparity range of 15, 19, 59, and 59, respectively. Our algorithm is applied on each pair of stereo images to calculate the disparity results. In each case, the computational time is recorded. Our experiment is implemented on a PC equipped with a 3.40 GHz Intel core i7 CPU and a 4GB memory. In our proposed method, the mainly time-consuming parts are the aggregation step and the disparity refinement step, accounting for 60.61% and 24.49% of the total time respectively. However, both of them as well as the cost computation can be performed in parallel on the GPU. With the help of OpenMP, parts of our algorithm are parallelized on the CPU, the acceleration ratio is about $3.2\times$. The computational time for *Tsukuba*, *Venus*, *Teddy*, and *Cones* are 3.11s, 5.70s, 12.34s, and 12.48s, respectively. As the algorithm has not been fully optimized, and it is only implemented on CPU instead of the GPU, it is expected be accelerated greatly in our future work.

IV. Conclusions

In this paper, we propose a secondary refinement scheme and a combined cost to improve the performance of local stereo matching. The secondary refinement scheme, namely RADAR, mainly focuses on handling remaining artifacts after traditional disparity refinement. In the combined cost, a modified color census transform (MCCT) is proposed combined with truncated AD and gradients. Experimental results show that our proposed method achieves the state-of-the-art performance and is one of the best local stereo matching methods on Middlebury benchmark. In addition, experimental results on four representative real-world sequences and some depth-based applications show the effectiveness of our method as well. A parallel implementation on CPU also demonstrates the parallelizability of our method.

ACKNOWLEDGEMENT

This work was partly supported by the grant of National Science Foundation of China 61370115, and Shenzhen Peacock Plan.

REFERENCES

[1] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vis.*, vol. 47, no. 1-3, pp. 7–42, 2002.

[2] K. Yoon and I. Kweon, “Adaptive support-weight approach for correspondence search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, 2006.

[3] V. Kolmogorov and R. Zabih, “Computing visual correspondence with occlusions using graph cuts,” in *Proc. IEEE Intl Conf. Computer Vision*, 2001, pp. 508–515.

[4] J. Sun, N.-N. Zheng and H.-Y. Shum, “Stereo matching using belief propagation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, 2003.

[5] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, “Fast cost-volume filtering for visual correspondence and beyond,” in *Proc. IEEE Intl Conf. Computer Vision and Pattern Recognition*, 2011, pp. 3017–3024.

[6] Q. Yang, “Hardware-efficient bilateral filtering for stereo matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–8, 2013.

[7] C. Pham and J. Jeon, “Domain transformation-based efficient cost aggregation for local stereo matching,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1119–1130, 2013.

[8] Q. Yang, “Recursive bilateral filtering,” in *European Conference on Computer Vision*, 2012, pp. 399–413.

[9] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza, “Linear stereo matching,” in *Proc. IEEE Intl Conf. Computer Vision*, 2011, pp. 1708–1715.

[10] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, “On building an accurate stereo matching system on graphics hardware,” in *Proc. IEEE Intl Conf. Computer Vision Workshops*, 2011, pp. 467–474.

[11] D. Scharstein and R. Szeliski, “Middlebury Stereo Website[Online],” Available: http://vision.middlebury.edu/stereo/.

[12] J. Jiao, R. Wang, W. Wang, S. Dong, Z. Wang, and W. Gao, “Cost-volume filtering-based stereo matching with improved matching cost and secondary refinement,” in *Proc. IEEE Intl Conf. Multimedia and Expo*, 2014, to be published.

[13] S.C. Pei and Y.Y. Wang, “Color invariant census transform for stereo matching algorithm,” in *IEEE International Symposium on Consumer Electronics*, 2013, pp. 209–210.

[14] J. Geusebroek, R. Boomgaard, and A.W.M. Smeulders, “Color invariance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 12, pp. 1338–1350, 2001.

[15] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 679–698, 1986.

[16] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[17] W. Wang and C. Zhang, “Local disparity refinement with disparity inheritance,” in *Symposium on Photonics and Optoelectronics*, 2012, pp. 1–4.

[18] Y.-C. Wang, C.-P. Tung, and P.-C. Chung, “Efficient disparity estimation using hierarchical bilateral disparity structure based graph cut algorithm with foreground boundary refinement mechanism,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 5, pp. 784–801, 2013.