

## Research Article

# A Line-Based Adaptive-Weight Matching Algorithm Using Loopy Belief Propagation

Hui Li,<sup>1</sup> Xiao-Guang Zhang,<sup>1</sup> and Zheng Sun<sup>2</sup>

<sup>1</sup>School of Mechanical and Electrical Engineering, China University of Mining & Technology, Xuzhou 221116, China

<sup>2</sup>College of Mechanical and Electrical Engineering, Zao Zhuang University, Zaozhuang 277160, China

Correspondence should be addressed to Xiao-Guang Zhang; doctorzgx@hotmail.com

Received 22 July 2014; Accepted 22 March 2015

Academic Editor: Hakim Naceur

Copyright © 2015 Hui Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In traditional adaptive-weight stereo matching, the rectangular shaped support region requires excess memory consumption and time. We propose a novel line-based stereo matching algorithm for obtaining a more accurate disparity map with low computation complexity. This algorithm can be divided into two steps: disparity map initialization and disparity map refinement. In the initialization step, a new adaptive-weight model based on the linear support region is put forward for cost aggregation. In this model, the neural network is used to evaluate the spatial proximity, and the mean-shift segmentation method is used to improve the accuracy of color similarity; the Birchfield pixel dissimilarity function and the census transform are adopted to establish the dissimilarity measurement function. Then the initial disparity map is obtained by loopy belief propagation. In the refinement step, the disparity map is optimized by iterative left-right consistency checking method and segmentation voting method. The parameter values involved in this algorithm are determined with many simulation experiments to further improve the matching effect. Simulation results indicate that this new matching method performs well on standard stereo benchmarks and running time of our algorithm is remarkably lower than that of algorithm with rectangle-shaped support region.

## 1. Introduction

Stereo vision is a fundamental technique for extracting 3D information of a scene from two or more 2D images. It is widely applied in robot navigation, remote sensing, and industrial automation. One of the key technologies of stereo vision is stereo matching, which produces a disparity map. The stereo matching algorithm can be classified into two broad categories: global-based and local-based algorithms.

Global-based matching algorithms follow the energy minimization principle. First, an energy function is established, consisting of a data term and a smoothness term. Next, this function is minimized with a global optimization method. Dynamic programming [1], loopy belief propagation (LBP) [2, 3], and graph cut [4, 5] are usually employed to identify the minimum energy required for a global-based algorithm. Comprehensive global constraint information can produce a more accurate disparity map in a global-based algorithm.

A local-based matching algorithm is a simple and effective method for stereo matching that is commonly used. An important underlying principle of local-based matching is that pixels in a support region have an approximately equal disparity. To satisfy this principle, it is very important to determine the support region size. The support region must contain enough pixels for intensity variation, and the support region must include only those pixels with the same disparity. Thus, the traditional, local-based matching method is prone to false matching for pixels from the depth discontinuities region, since those pixels are from different depths. To ensure that a local-based matching algorithm performs well in practical applications, various approaches have been proposed. For example, adaptive windows have been used to improve matching results. These methods search an appropriate support region for each pixel, greatly improve the performance of matching results, and outperform standard local-based methods [6–10]. However, it is difficult to search a support region with an arbitrary shape and size, and most of

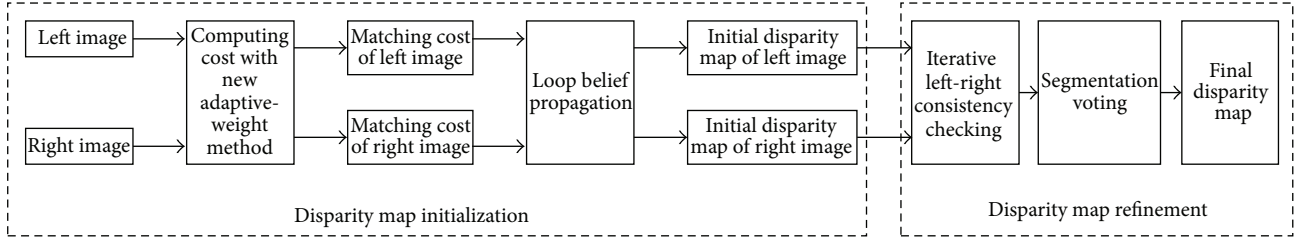


FIGURE 1: Block diagram of our algorithm.

these methods have a high computational complexity. Other researchers assign different support-weights to the pixels in a support region, keeping the size and shape of a support region constant [11–13].

In recent years, several methods for acquiring satisfactory effect of stereo matching have been adopted. Yang et al. [17] presented a stereo matching algorithm which integrates color-weight, belief propagation, left-right checking, color segmentation, plane fitting, and depth enhancement. Mei et al. [18] integrated the AD-census cost measurement function, the cost aggregation method based cross-based region, the scanline optimize method, the multistep refinement method, and the accelerative algorithm based on CUDA into their algorithm.

The algorithm presented in this paper is inspired from adaptive-weight matching algorithm. In this paper, the aim is to propose a low computation complexity and high accuracy stereo matching algorithm. So the rectangle-shaped support region is substituted for the line-shaped support region. Lacking of enough pixel information is a main weakness of the line-shaped support region, which is easy to cause error matching. Adaptive-weight can make full of limited pixel information, by analyzing the characteristic of the adaptive-weight model proposed in [13] on disparity accuracy, we use neural network (NN) to determine the spatial proximity and mean-shift based segmentation method to effectively describe the color similarity.

In addition, several approaches are applied to complete the algorithm. We develop a new pixel dissimilarity measurement function which combines Birchfield pixel dissimilarity measurement function and census transform to compute the matching cost. The loopy belief propagation method proposed in [2] is employed to estimate the initial disparity map, which is optimized with min convolution and image pyramid. There are several measurable improvements for the initial disparity map. To further improve the accuracy of the initial disparity map, iterative left-right consistency (LRC) checking and segmentation voting are used to refine the disparity map by analyzing the features of the initial disparity map.

## 2. Algorithm Description

The algorithm presented in this paper can be divided into two steps: a disparity map initialization step and a refinement step. The framework of the algorithm is shown in

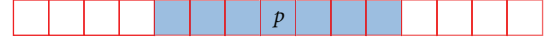


FIGURE 2: The assumption of segmentation-based stereo matching.

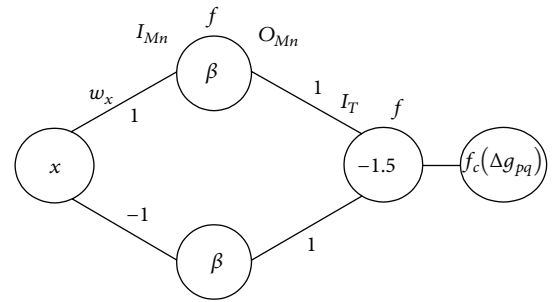


FIGURE 3: Spatial proximity described by neural network.

Figure 1. A detailed description of this algorithm is given in the following sections.

**2.1. Adaptive-Weight Based Cost Aggregation Method.** Assuming the two pixels  $p$  and  $q$ , the disparity of center pixel  $p$  wants to be computed.  $S(p)$  is the support region of pixel  $p$ , while  $q$  is a neighboring pixel of  $p$  in the support region. The support-weight of  $q$  is assigned by the following according to [12]:

$$w(p, q) = f_c(\Delta c_{pq}) f_c(\Delta g_{pq}), \quad (1)$$

where  $f_c(\Delta g_{pq})$  represents the spatial proximity,  $f_c(\Delta c_{pq})$  represents the color similarity, and  $w(p, q)$  is the support-weight. Our algorithm is designed on the basis of this framework. The list of variables used in this paper is shown in the end of the paper.

**2.1.1. The Model of Line-Based Adaptive-Weight.** Wang et al. [13] noted that when the support region is large enough, color similarity plays a major role in computing the center pixel disparity within a certain range. As shown in Figure 2, red represents the support region  $S(p)$ . We used the pixels in  $S(p)$  to compute the disparity of  $p$ . The effects of spatial proximity can be neglected in the pale blue region according to [13]. We call this region the transition area, represented by  $T(p)$ . To satisfy this principle, the neural network can be applied in the design of this spatial proximity model.

Figure 3 shows the spatial proximity model established by neural network. The position of a pixel is the input, the spatial

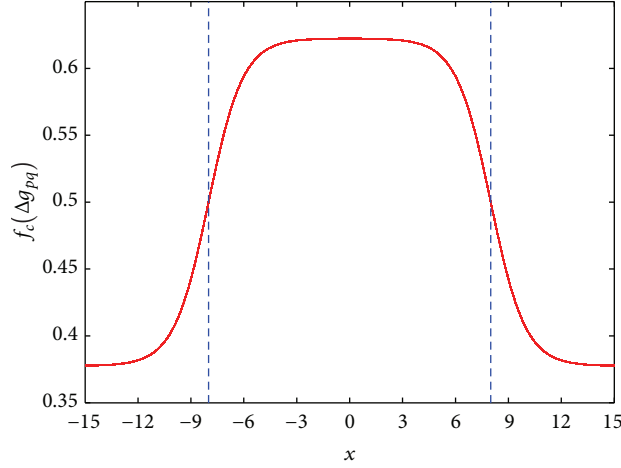


FIGURE 4: Spatial proximity proposed in this paper when  $\alpha = 1$ ,  $\beta = 8$ , and  $L_s = 15$ .

proximity is the output, and the connect weights are shown as in the figure. In fact, the distance  $x - x_p$  is the input of neural network. To simplify the notations, suppose that the center point  $p$  is at  $x_p = 0$ , the distance  $x - x_p$  can be simplified into  $x$  which represents the position of a pixel. The concrete form of the spatial proximity is expressed by

$$\begin{aligned} I_{M1} &= x + \beta, \\ I_{M2} &= -x + \beta, \\ O_{Mi} &= f(I_{Mi}), \\ I_T &= \sum_{i=1}^2 O_{Mi} - 1.5, \\ f_c(\Delta g_{pq}) &= f(I_T), \end{aligned} \quad (2)$$

where  $f(x) = 1/(1 + e^{-\alpha x})$  is the sigmoid function. Figure 4 demonstrates the varied trend of the spatial proximity according to the position of a pixel.

In Figure 4, the space between the two blue lines is the transition area and the support region is represented by the whole  $x$ -axis. It can be seen from Figure 4 that (1) the spatial proximity of pixel in the transition area is significantly greater than that of pixel in other area, which accords with the spatial proximity model of traditional adaptive-weight theory; (2) there is not much difference between these pixels in the transition area for the spatial proximity, which means that the influence of spatial proximity can be neglected.

According to the segmentation-based stereo matching principle [14, 19], a new model of color similarity is established by the following in [20]:

$$f_c(\Delta c_{pq}) = \begin{cases} 1 & \text{if } \text{seg}(q) = \text{seg}(p) \\ \exp\left(-\frac{\Delta^2 c_{pq}}{\gamma_c}\right) & \text{otherwise.} \end{cases} \quad (3)$$

Equation (3) shows that color similarity contributes enormously to measure the dissimilarity between center pixel

$p$  and its neighbor pixel  $q$  when  $p$  and  $q$  belong to the same segmentation.

That color similarity model based on image segmentation can achieve good performance, as introduced in [13]. Mean-shift is a nonparametric estimation iterative technique, and its application domains include computer vision, clustering, and image processing [21]. In this work, we use mean-shift as the segmentation method.

**2.1.2. Cost Aggregation.** The matching cost of pixel  $p$  with disparity  $d$  is represented by

$$E(p, d) = \frac{\sum_{q \in L_p, \bar{q}_d \in L_{\bar{p}_d}} w(p, q) w(\bar{p}_d, \bar{q}_d) e(q, \bar{q}_d)}{\sum_{q \in L_p, \bar{q}_d \in L_{\bar{p}_d}} w(p, q) w(\bar{p}_d, \bar{q}_d)}, \quad (4)$$

where  $\bar{p}_d$  and  $\bar{q}_d$  are the corresponding pixels of  $p$  and  $q$ , respectively, when the disparity of the center pixel is  $d$ .

The pixel dissimilarity measurement function  $e(q, \bar{q}_d)$  in (4) is very important for cost aggregation. The absolute difference and Birchfield function [22] are widely used in cost aggregation. To improve the matching accuracy of the textureless and repetitive regions, the pixel dissimilarity measurement function is described by combining Birchfield function and census transform:

$$\begin{aligned} e(q, \bar{q}_d) &= \min(C_{\text{Birch}}(q, \bar{q}_d, I_R, I_L), T_{\text{Birch}}) \\ &\quad + \min(C_{\text{Census}}(q, \bar{q}_d), T_{\text{Census}}). \end{aligned} \quad (5)$$

To validate the effect of this cost aggregation method, simulation results on Teddy and Cones with Birchfield method and our method are shown in Figure 5.

**2.2. Initial Disparity Determination.** The winner-take-all (WTA) searching strategy is a common method for determining disparity, which can be expressed by

$$D(p) = \arg \min_d E(p, d). \quad (6)$$

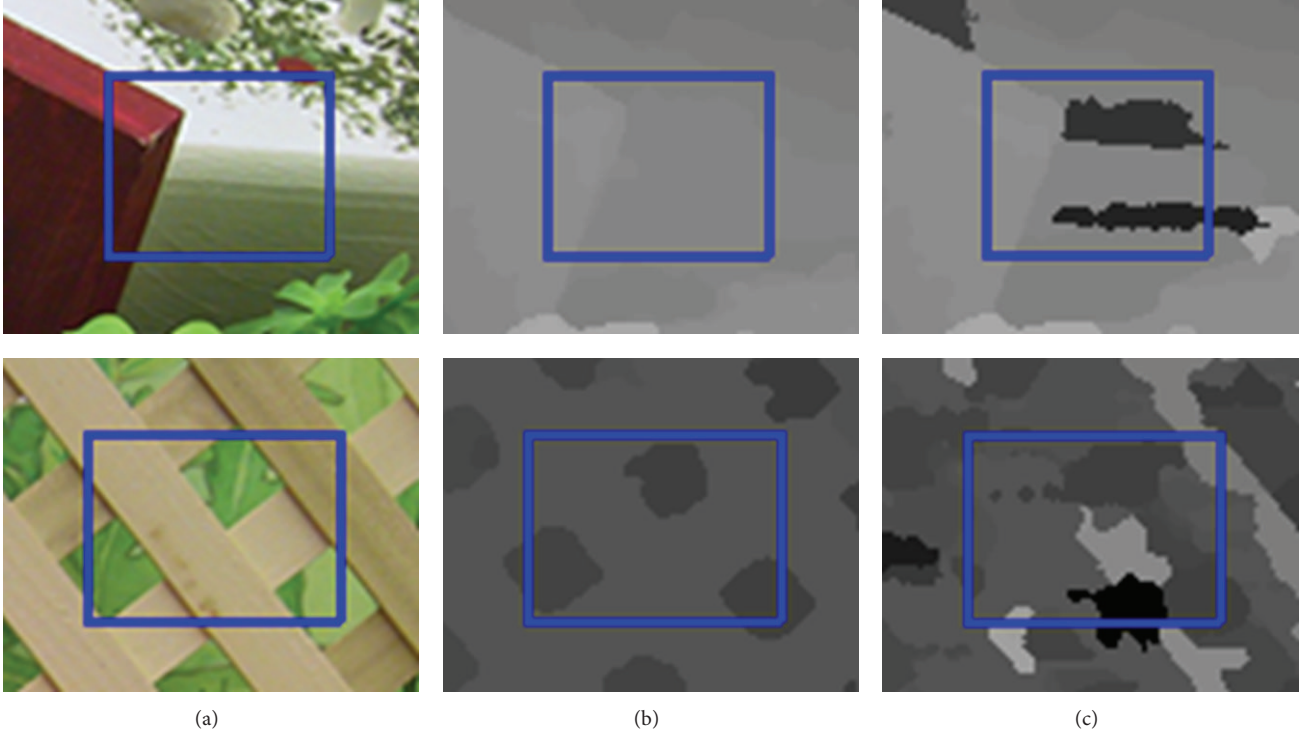


FIGURE 5: (a) Part of the origin image. (b) Disparity results which are computed with our cost aggregation method. (c) Disparity results which are computed with Birchfield pixel dissimilarity measurement function.

- (1) Computing matching cost  $E$  by (4)  
    Initializing pyramid level  $l$  and down sampling factor  $\eta = 0.5$
- (2) Pyramid initialization  
    Pyramid(1).cost =  $E$
- (3) Establishing pyramid structure with  $l$  and  $\eta$
- (4) For each level  $i$
- (5)   For each iteration  $j$
- (6)     Updating left, right, up and down direction message for each pixel
- (7)   End
- (8) End
- (9) Determining disparity for each pixel by max-product principle

ALGORITHM 1: The procedure of initial determination procedure.

WTA tends to produce a low accuracy disparity map. Therefore, we adopt an efficient LBP algorithm proposed in [2] in this paper. In this LBP algorithm, FFT convolution and image pyramid are integrated into LBP, which can effectively decrease the complexity of LBP and increase the matching effect. The flowchart of the initial determination procedure is shown in Algorithm 1.

**2.3. Disparity Refinement.** It is inevitable that initial disparity maps will contain many error-matched pixels. To refine the disparity map, a two-step postprocessing method is put forward in this section.

**2.3.1. Left-Right Consistency Check.** The disparity map  $D_L$  for the left image is computed by previous steps. The disparity map  $D_R$  for the right image is computed in a similar manner. In [15, 17], pixels are classified into several types according to  $D_L$ ,  $D_R$ , and  $E_R$  to remove outliers. Then different strategies are designed for different type of pixels to determine its disparity.

In this work, we analyzed the property of initial disparity map. Figure 6 shows the distribution of bad pixels after executing the disparity initialization step of our algorithm for Tsukuba. It can be seen from this figure that most of the bad pixels concentrate in the occluded region. According to this result, most of the pixels match correctly and the initial

- (1) Establishing the table of pixel type  $S_{ii}$  according to (7)-(8)
- (2)  $S_t \leftarrow S_{ii}$
- (3) For each iteration  $k$
- (4) For each undependable pixel  $p_i$  which is determined by  $S_t$
- (5) Searching the nearest dependable pixel  $p_i^L$  and  $p_i^R$
- (6)  $d(p_i) \leftarrow \min(d_o(p_i^L), d_o(p_i^R))$  //where  $d_o(\cdot)$  represents initial disparity of pixel
- (7) If  $d(p_i) \neq d_o(p_i)$
- (8) Set  $p_i$  as dependable pixel in  $S_t$
- (9) End
- (10) End
- (11) End
- (12) End

ALGORITHM 2: The disparity map after execution of left-right consistency checking.

TABLE 1: The error percentages in different region for disparity refinement step.

	Nonocc	All	Disc
Initial disparity map	1.21%	3.20%	6.42%
Left-right consistency	1.17%	2.40%	6.30%
Segmentation voting	0.95%	1.62%	5.07%

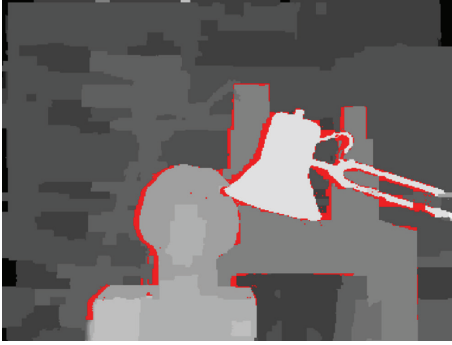


FIGURE 6: The distribution of bad pixels for initial disparity map of Tsukuba. Bad pixels are marked with red points.

disparity in the nonocc region should be trusted. Therefore an iterative left-right consistency check is proposed for handling this.

Pixels can be divided into two types: undependable pixels and dependable pixels. Pixel  $p$  is classified as dependable when it meets the following condition:

$$D_L(p) = D_R(p - D_L(p)). \quad (7)$$

Pixel  $p$  is considered to be undependable if it fits the following condition:

$$D_L(p) \neq D_R(p - D_L(p)). \quad (8)$$

The new disparity of the undependable pixel can be computed as in Algorithm 2.

Figure 7 shows the result after the execution of iterative left-right consistency checking. Table 1 shows the detailed data for the role of iterative left-right consistency checking.



FIGURE 7: The disparity map after execution of left-right consistency checking.

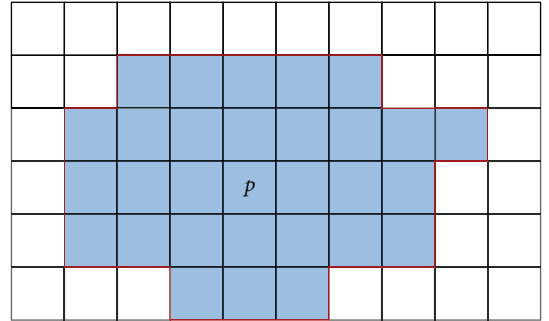


FIGURE 8: Pixelwise region for segmentation voting.

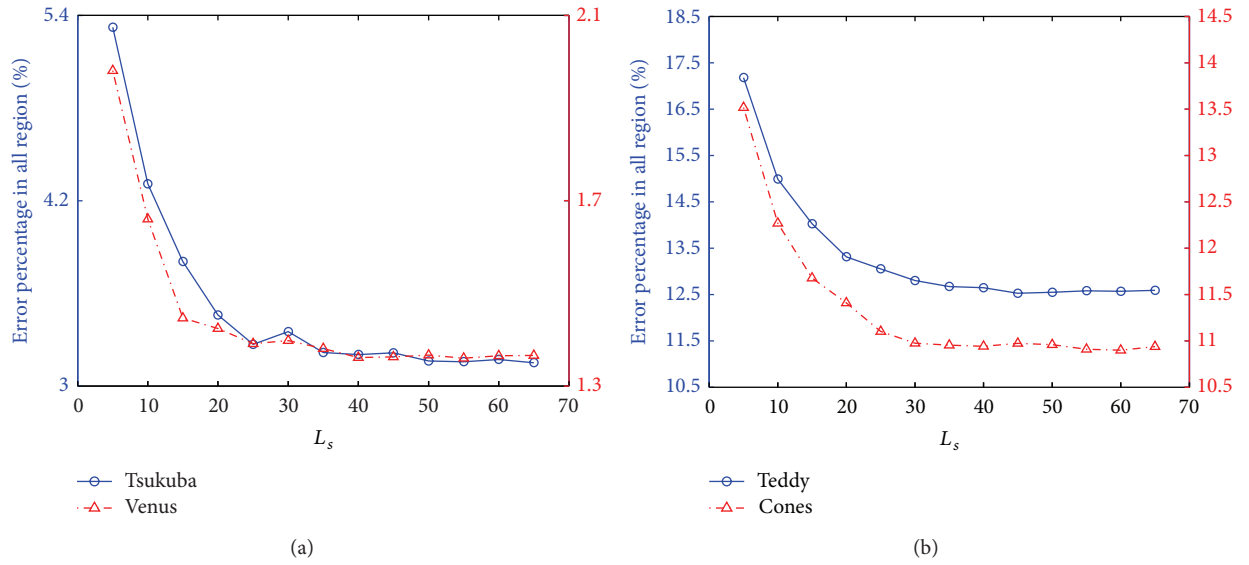
**2.3.2. Segmentation Voting.** The pixelwise region shown in Figure 8 is established according to color consistency. This region can be represented by  $U(p)$ . Pixels in  $U(p)$  satisfy the following condition:

$$\begin{aligned} |I_q - I_p| &< \tau, \quad q \in U(p), \\ |x_q - x_p| &< L_{sv}, \\ |y_q - y_p| &< L_{sv}. \end{aligned} \quad (9)$$

Let  $H(d)$  represent the frequency distribution of disparity  $d$  in  $U(p)$ . The new disparity is updated by Algorithm 3.



FIGURE 9: The disparity map of Tsukuba obtained by segmentation voting.

FIGURE 10: (a) Performance evaluation of the proposed method when varying  $L_s$  (Tsukuba and Venus). (b) Performance evaluation of the proposed method when varying  $L_s$  (Teddy and Cones).

```

(1)  $H(d) \leftarrow 0$ 
(2) For each  $q \in U(p)$ 
(3)   For each  $d \in [d_{\min}, d_{\max}]$ 
(4)     If  $d == D_L(q)$ 
(5)        $H(D_L(q)) \leftarrow H(D_L(q)) + 1$ 
(6)     End
(7)   End
(8) End
(9)  $D_L(p) = \arg \max_d H(d)$ 

```

ALGORITHM 3: Disparity determination of segmentation voting.

The  $3 \times 3$  median filter is applied to the left disparity map. Figure 9 shows the effect of segmentation voting on the accuracy of the disparity map.

### 3. Results and Discussion

**3.1. Parameters Determining.** The parameters involved in our algorithm greatly affect the performance of the algorithm. In this section, we present the parameter settings.

We considered eight main parameters:  $L_s$ ,  $\alpha$ ,  $\beta$ ,  $\gamma_c$ ,  $h_s$ ,  $h_l$ ,  $L_{sv}$ , and  $\tau$ , which are kept constant for all benchmarks.

Figure 10 shows the influence of  $L_s$  on the accuracy of disparity map obtained by our algorithm. When  $L_s$  varies from 35 to 65, our algorithm is insensitive to  $L_s$ .

Figure 11 demonstrates that the performance of our algorithm varies with  $\alpha$  and  $\beta$ . From these figures, it can be noted that (1) generally speaking when  $\beta$  is larger than 15, the influence of  $\alpha$  over algorithm performance is small for Tsukuba, Venus, and Teddy. (2) Our algorithm shows good performance for Tsukuba and Teddy when  $\beta \in [0, 45]$  and  $\alpha \in [0.3, 1]$ . (3) The error percentage tends to decrease with a decreasing  $\alpha$  when  $\beta$  is smaller than 15. (4) In regard to Cones,



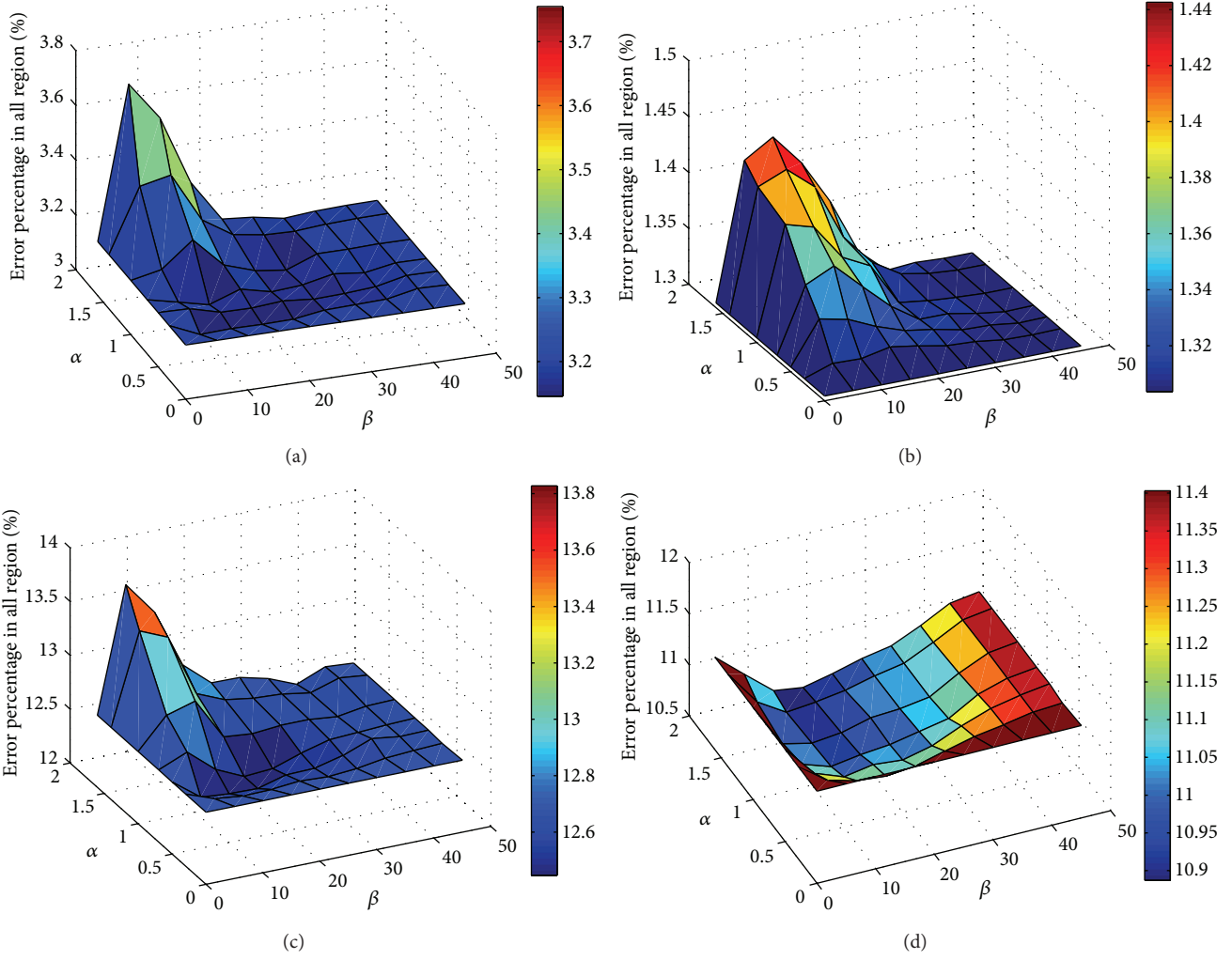


FIGURE 11: (a) Performance evaluation of the proposed method according to  $\alpha$  and  $\beta$  for Tsukuba. (b) Performance evaluation of the proposed method according to  $\alpha$  and  $\beta$  for Venus. (c) Performance evaluation of the proposed method according to  $\alpha$  and  $\beta$  for Teddy. (d) Performance evaluation of the proposed method according to  $\alpha$  and  $\beta$  for Cones.

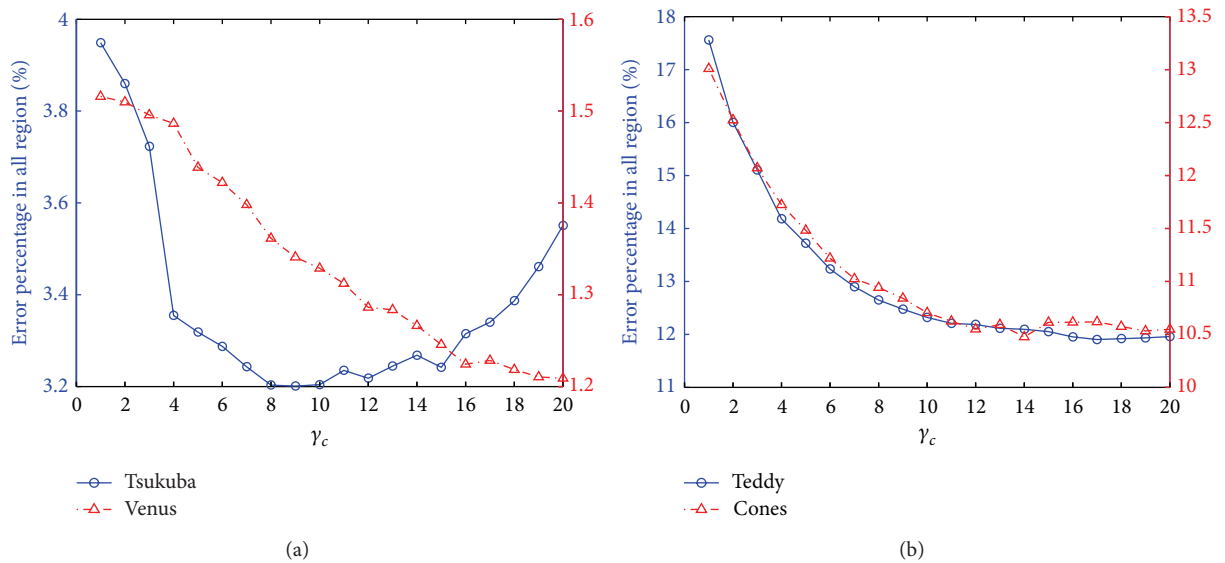


FIGURE 12: (a) Performance evaluation of the proposed method when varying  $\gamma_c$  (Tsukuba and Venus). (b) Performance evaluation of the proposed method when varying  $\gamma_c$  (Teddy and Cones).

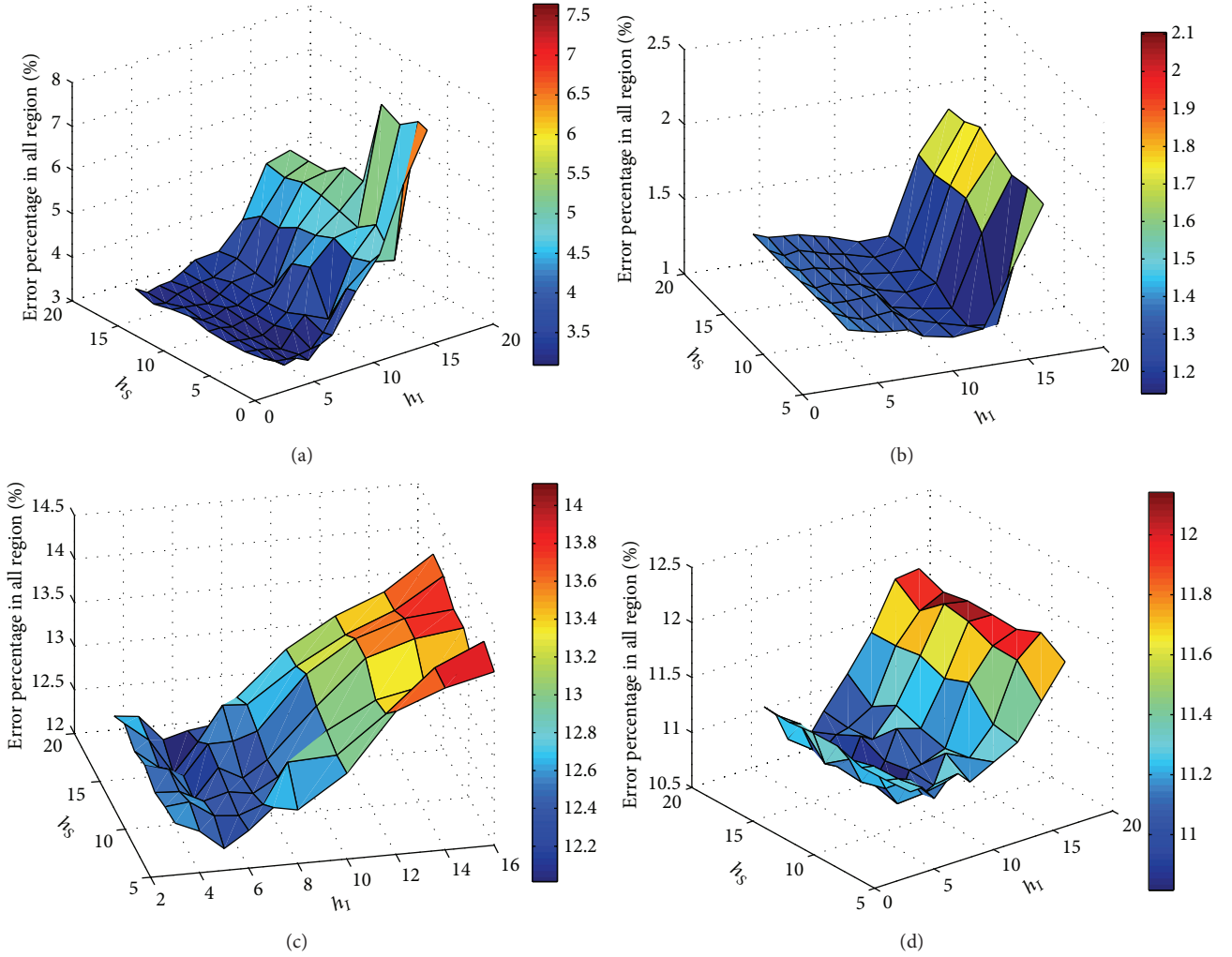


FIGURE 13: (a) Performance evaluation of the proposed method according to  $h_s$  and  $h_l$  for Tsukuba. (b) Performance evaluation of the proposed method according to  $h_s$  and  $h_l$  for Venus. (c) Performance evaluation of the proposed method according to  $h_s$  and  $h_l$  for Teddy. (d) Performance evaluation of the proposed method according to  $h_s$  and  $h_l$  for Cones.

the performances of this algorithm improve as  $\alpha$  increases, and when  $\beta$  lies between 15 and 30.

Figure 12 shows the influence of  $\gamma_c$  on performance. The trend of the error percentage is U-shaped for Tsukuba when  $\gamma_c$  is smaller than 20. The error percentage bottom out when  $\gamma_c$  is between 7 and 13. The error percentage for Venus follows a downward trend. For Teddy and Cones, the error percentage is inversely proportional to  $\gamma_c$ . When  $\gamma_c$  is larger than 12, the error percentage is insensitive to  $\gamma_c$ .

Figure 13 shows the performance of our algorithm according to  $h_s$  and  $h_l$ . From these figures, our algorithm has a good ability of robustness with different values of  $h_s$ . When  $h_l \in [3, 8]$ , the error percentage of the disparity map obtained by our algorithm is still fairly low.

For the disparity map refinement step, two main parameters must be set.  $L_{sv}$  and  $\tau$  are previously introduced. Figure 14 shows the influence of  $L_{sv}$  and  $\tau$  on performance. When  $L_{sv} \in [5, 20]$  and  $\tau \in [7, 10]$ , all data sets show good performance.

TABLE 2: Parameter setting for Middlebury benchmark.

$L_s$	$\alpha$	$\beta$	$\gamma_c$	$h_s$	$h_l$	$L_{sv}$	$\tau$
40	1	15	8	15	5	7	12

**3.2. Experimental Results.** We evaluate our algorithm on Middlebury benchmarks [23] with error threshold 1. The test platform hardware consists of T9600 CPU and 5 GB memory. Software consists of MATLAB 2014a and VS2012. Parameters are shown in Table 2.

Simulation results on Middlebury data sets are presented in Figure 15. The quantitative performance of our algorithm is shown in Tables 3 and 4. Our algorithm ranks 5th in the Middlebury data set (July 1, 2014).

These results demonstrate our algorithm has a good performance. However, it is difficult to decrease the error percentages in the three regions (nonocc, all, and disc) at the same time. This is because many pixels with correct disparity



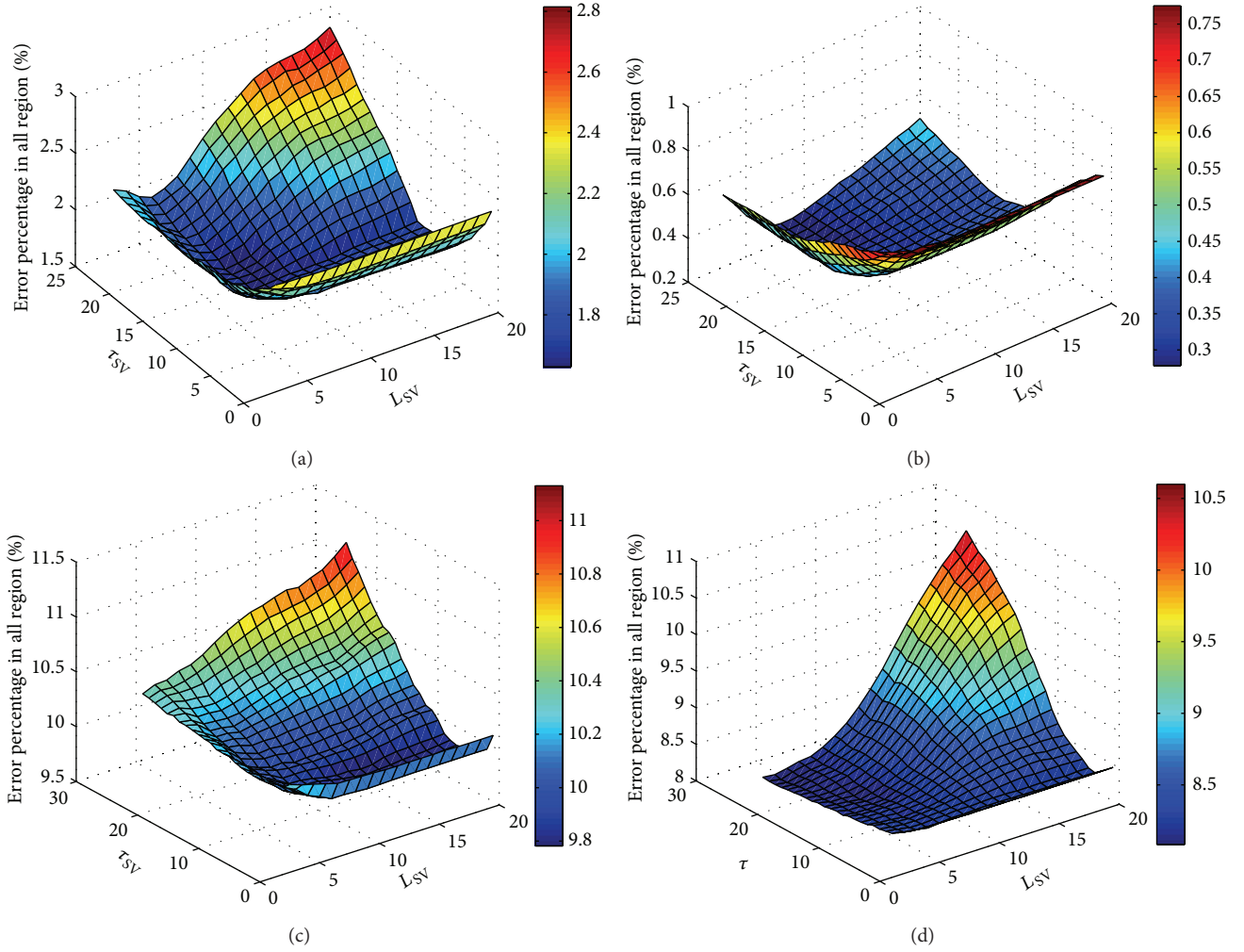


FIGURE 14: (a) Performance evaluation of the proposed method according to  $L_{sv}$  and  $\tau$  for Tsukuba. (b) Performance evaluation of the proposed method according to  $L_{sv}$  and  $\tau$  for Venus. (c) Performance evaluation of the proposed method according to  $L_{sv}$  and  $\tau$  for Teddy. (d) Performance evaluation of the proposed method according to  $L_{sv}$  and  $\tau$  for Cones.

TABLE 3: Quantitative evaluation results for Middlebury benchmark.

Algorithm	Tsukuba			Venus			Teddy			Cones		
	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc
AdaptingBP	1.11 <sub>22</sub>	1.37 <sub>9</sub>	5.79 <sub>24</sub>	0.10 <sub>4</sub>	0.21 <sub>7</sub>	1.44 <sub>8</sub>	4.22 <sub>17</sub>	7.06 <sub>15</sub>	11.8 <sub>18</sub>	2.48 <sub>22</sub>	7.92 <sub>28</sub>	7.32 <sub>27</sub>
CoopRegion [14]	0.87 <sub>5</sub>	1.16 <sub>2</sub>	4.61 <sub>4</sub>	0.11 <sub>5</sub>	0.21 <sub>5</sub>	1.54 <sub>14</sub>	5.16 <sub>30</sub>	8.31 <sub>20</sub>	13.0 <sub>25</sub>	2.79 <sub>39</sub>	7.18 <sub>14</sub>	8.01 <sub>47</sub>
Our algorithm	0.89 <sub>8</sub>	1.59 <sub>23</sub>	4.78 <sub>9</sub>	0.11 <sub>5</sub>	0.36 <sub>24</sub>	1.49 <sub>11</sub>	4.33 <sub>18</sub>	9.93 <sub>32</sub>	11.2 <sub>17</sub>	2.81 <sub>41</sub>	8.46 <sub>45</sub>	7.85 <sub>40</sub>
RDP [15]	0.97 <sub>11</sub>	1.39 <sub>11</sub>	5.00 <sub>11</sub>	0.21 <sub>39</sub>	0.38 <sub>29</sub>	1.89 <sub>24</sub>	4.84 <sub>22</sub>	9.94 <sub>33</sub>	12.6 <sub>22</sub>	2.53 <sub>25</sub>	7.69 <sub>20</sub>	7.38 <sub>28</sub>
MultiRBF [16]	1.33 <sub>47</sub>	1.56 <sub>20</sub>	6.02 <sub>32</sub>	0.13 <sub>11</sub>	0.17 <sub>2</sub>	1.84 <sub>21</sub>	5.09 <sub>28</sub>	6.36 <sub>8</sub>	13.4 <sub>29</sub>	2.90 <sub>48</sub>	6.76 <sub>8</sub>	7.10 <sub>24</sub>

TABLE 4: The results of whole performance evaluation for Middlebury benchmark.

Algorithm	Average rank	Average percent bad pixels
AdaptingBP	16.5	4.23
CoopRegion	12.2	4.41
Our algorithm	22.8	4.48
RDP	22.9	4.57
MultiRBF	23.2	4.39

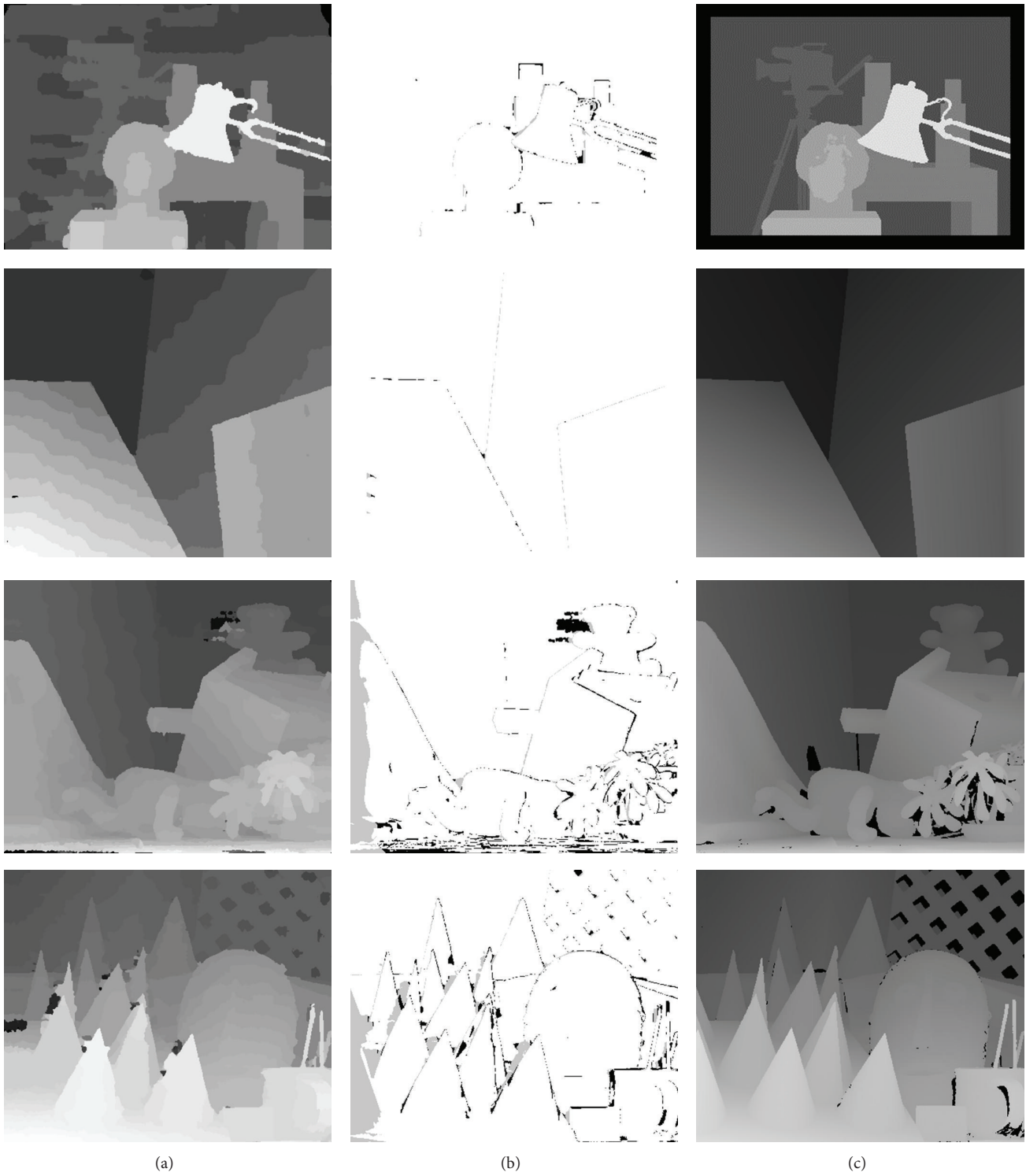


FIGURE 15: (a) The final disparity map obtained by our algorithm. (b) Error maps. (c) Ground truth.

TABLE 5: Comparison between standard LRC checking and iterative LRC checking presented in Section 2.3.1.

Algorithm	Nonocc	All	Disc
Initial disparity	1.21%	3.20%	6.42%
Standard LRC checking	1.43%	2.10%	7.54%
Iterative LRC checking	1.17%	2.40%	6.30%

TABLE 6: The average running time of each part of our algorithm for Tsukuba.

Mean-shift	Cost aggregation	LBP	Iterative LRC	Segmentation voting
7.37	19.23	0.78	0.24	0.30

TABLE 7: The variation of average running time of cost aggregation of our rectangle-based algorithm under different size of support region.

The size of support region	$11 \times 11$	$15 \times 15$	$19 \times 19$	$23 \times 23$	$27 \times 27$
Running time	37.02	69.81	106.53	154.41	213.94

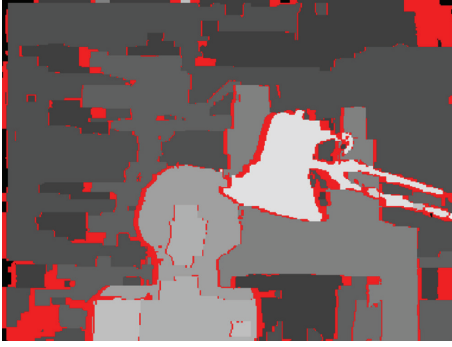


FIGURE 16: Erroneous disparity distribution determined by (7)-(8) for the Tsukuba image.



FIGURE 17: Erroneous disparity distribution after executing standard LRC checking for Tsukuba.

are classified as undependable, according to (8). Tsukuba can be used as an example. Figure 16 represents the bad pixels detected by (7)-(8).

Figure 6 shows the bad pixels obtained by comparing the initial disparity map and ground truth. Figure 6 indicates that true bad pixels are distributed in the occlusion region. However, it can be inferred from Figure 16 that many pixels in nonocc and disc regions are mistakenly classified as undependable. When standard LRC checking is used, misclassified pixels in the disc region may be assigned a wrong disparity, as shown in Figure 17. Table 5 shows quantitative comparison results for standard LRC checking and iterative LRC checking.

Thus, it can be seen that the error percentage in the disc region greatly increases after applying standard LRC checking. Because the disc region is part of the nonocc region, the error percentage in the nonocc region also increases. Our iterative LRC checking method can effectively improve the performance of our stereo matching algorithm.

**3.3. Running Time of Our Algorithm.** In this section, we investigate the computation running time of our algorithm. The running time directly reflects computational complexity. Without a loss of generality, our algorithm runs 50 times and

average running time was calculated. Results are shown in Table 6.

The new rectangle-based adaptive-weight method can be obtained by imposing the  $y$ -direction constraint on the new weight model. Table 7 displays the running time of cost aggregation of our algorithm with the rectangle-shaped support region under different sizes of support region. The rectangle-based algorithm also runs 50 times under each size of support windows.

Our algorithm with the rectangle-shaped support region performed best when the size of the support region is  $27 \times 27$ ; the error percentages of the initial disparity map in nonocc, all, and disc regions are 1.45%, 3.42%, and 7.08%, respectively. Tables 5–7 show that our algorithm produced notable results in decreasing computation complexity and improving performance.

## 4. Conclusions

In this work, we proposed a new line-based adaptive-weight stereo matching algorithm that integrates several methods. The main conclusions that can be drawn from our results are as follows.

- (1) Cost aggregation is the most time-consuming part of stereo matching algorithm. Using a line-shaped support region can dramatically reduce the elapsed time of cost aggregation.
- (2) The adaptive-weight model proposed in this paper can produce a rather satisfactory initial disparity map in the absence of enough pixel information.
- (3) Experimental results show that the algorithm proposed in this paper can attain a better matching effect with less running time.

Although our algorithm has a good performance on Middlebury data sets, there is still much room for improvement.

- (1) There are too many parameters in our algorithm to accommodate different image pairs. In further research, we will analyze the intrinsic relationship among the parameters and reduce the number of parameters.
- (2) Figure 13 indicates that image characteristics have a significant impact on the optimum values of  $\alpha$  and  $\beta$ . In future studies, we will explore how the optimum values of  $\alpha$  and  $\beta$  vary according to the texture of image.

## List of Variables

- $L_s$ : The half-length of support region  
 $\alpha$ : Shape controlling parameter  
 $\beta$ : The half-length of transition region  
 $h_s$ : Spatial radius of mean-shift  
 $h_t$ : Color radius of mean-shift  
 $L_{sv}$ : The half-length of segment for segmentation voting  
 $\tau$ : Confidence level of color similarity for segmentation voting.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors express their appreciation for the financial support of the Shandong Natural Science Foundation, Grant no. ZR2013FL033. They also extend their sincere gratitude to reviewers for their constructive suggestions.

## References

- [1] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 1075–1082, San Diego, Calif, USA, June 2005.
- [2] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 41–54, 2006.
- [3] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 399–406, San Diego, Calif, USA, June 2005.
- [4] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.
- [5] H. Li and G. Chen, "Segment-based stereo matching using graph cuts," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 74–81, Washington, DC, USA, June 2004.
- [6] T. Kanade and M. A. Okutomi, "Stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.
- [7] X.-Z. Zhou, G.-J. Wen, and R.-S. Wang, "Fast stereo matching using adaptive window," *Chinese Journal of Computers*, vol. 29, no. 3, pp. 473–479, 2006.
- [8] Y. Boykov, O. Veksler, and R. I. Zabih, "A variable window approach to early vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1283–1294, 1998.
- [9] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 556–561, Madison, Wis, USA, June 2003.
- [10] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1073–1079, 2009.
- [11] Y. Xu, D. Wang, T. Feng, and H.-Y. Shum, "Stereo computation using radial adaptive windows," in *Proceedings of 16th International Conference on Pattern Recognition (ICPR '02)*, vol. 3, pp. 595–598, Quebec, Canada, August 2002.
- [12] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650–656, 2006.
- [13] F. Wang and D. Huang, "Improved Yoon stereo matching algorithm based on adaptive weight," *Journal Electronic Measurement and Instrument*, vol. 24, no. 7, pp. 632–637, 2010.
- [14] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 15–18, Hong Kong, China, August 2006.
- [15] X. Sun, X. Mei, S.-H. Jiao, M.-C. Zhou, and H. Wang, "Stereo matching with reliable disparity propagation," in *Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT '11)*, pp. 132–139, Hangzhou, China, May 2011.
- [16] X.-Z. Zhou and P. Boulanger, "New eye contact correction using radial basis function for wide baseline videoconference system," in *Proceedings of 13th Pacific-Rim Conference on Multimedia (PCM '12)*, pp. 68–79, Singapore, December 2012.
- [17] Q. X. Yang, L. Wang, R. Yang, H. Stewénus, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 492–504, 2009.
- [18] X. Mei, X. Sun, M.-C. Zhou, S.-H. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on

- graphics hardware,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 467–474, Barcelona, Spain, November 2011.
- [19] Z.-F. Wang and Z.-G. Zheng, “A region based stereo matching algorithm using cooperative optimization,” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [20] F. Tombari, S. Mattoccia, and L. D. Stefano, “Segmentation-based adaptive support for accurate stereo correspondence,” in *Proceedings of 2nd Pacific-Rim Symposium on Image and Video Technology (PSIVT '07)*, pp. 427–438, Santiago, Chile, December 2007.
- [21] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [22] S. Birchfield and C. Tomasi, “Depth discontinuities by pixel-to-pixel stereo,” *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, 1999.
- [23] “Middlebury Stereo Vision Page,” <http://vision.middlebury.edu/stereo/>.



