Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5556990, 20 pages
https://doi.org/10.1155/2021/5556990

Research Article

Local Stereo Matching Using Adaptive Cross-Region-Based Guided Image Filtering with Orthogonal Weights

Lingyin Kong, Jiangping Zhu, and Sancong Ying

College of Computer Science, Sichuan University, Chengdu 610065, China

Correspondence should be addressed to Sancong Ying; yingsancong@scu.edu.cn

Received 12 January 2021; Accepted 30 April 2021; Published 7 May 2021

Academic Editor: Luis J. Yebra

Adaptive cross-region-based guided image filtering (ACR-GIF) is a commonly used cost aggregation method. However, the weights of points in the adaptive cross-region (ACR) are generally not considered, which affects the accuracy of disparity results. In this study, we propose an improved cost aggregation method to address this issue. First, the orthogonal weight is proposed according to the structural feature of the ACR, and then the orthogonal weight of each point in the ACR is computed. Second, the matching cost volume is filtered using ACR-GIF with orthogonal weights (ACR-GIF-OW). In order to reduce the computing time of the proposed method, an efficient weighted aggregation computing method based on orthogonal weights is proposed. Additionally, by combining ACR-GIF-OW with our recently proposed matching cost computation method and disparity refinement method, a local stereo matching algorithm is proposed as well. The results of Middlebury evaluation platform show that, compared with ACR-GIF, the proposed cost aggregation method can significantly improve the disparity accuracy with less additional time overhead, and the performance of the proposed stereo matching algorithm outperforms other state-of-the-art local and nonlocal algorithms.

1. Introduction

Binocular stereo vision can acquire disparity information with required accuracy only by using image pairs of the same scene that are obtained from different angles. It is widely applied in three-dimensional reconstruction [1], three-dimensional measurement [2], robot vision [3], unmanned driving [4], and so on. The purpose of stereo matching is to find corresponding points in a pair of images. The accuracy tends to affect the precision of disparity results. So, it is a critical procedure in binocular stereo vision systems and is a topic of significant research interest in the field of computer vision.

Currently, stereo matching algorithms are mainly divided into two categories. The first category is based on deep learning. In particular, algorithms based on convolutional neural networks (CNNs) have developed rapidly in recent years. Žbontar and LeCun [5] proposed an algorithm that utilizes the Siamese network to compute the matching cost. Pairs of small image patches were used to train the network to determine the similarities among the patches. Pang et al. [6] proposed a cascade residual learning network divided into two stages, where each stage independently calculates disparity maps and multiscale residual signals. Chang et al. proposed a pyramid stereo matching network [7] comprising a spatial pyramid pooling module and a 3D CNN module. Kunal Swami et al. [8] proposed an end-to-end network model to utilize rich multiscale context information, which most existing methods cannot achieve. A large effective receiving domain is implemented to extract multiscale context information, while retaining the required spatial information in the entire network. Kim et al. [9] proposed a network architecture that uses both the matching cost volume and disparity map as inputs. Their architecture contains two subnetworks, namely, the matching probability construction network and the confidence estimation network. Such methods achieve high matching accuracy, but ground truth disparity maps are required in advance, especially for end-to-end network models.

The second category is conventional algorithms. These conventional algorithms can be classified as global, semiglobal, and local algorithms. Global algorithms usually obtain disparities by solving the minimum value of the global energy functions, including graph cuts [10] and belief propagation [11]. They are characterized by high matching accuracy and low computing efficiency. The first semiglobal algorithm was proposed by Hirschmüller [12], which mainly used the idea of dynamic programming. The matching accuracy and computing efficiency of semiglobal algorithms lie between those of global and local algorithms. Local algorithms are based on cost aggregation within the specified support region, and the matching accuracy is usually lower than those of the first two types. However, the computing efficiency is higher. Such algorithms generally employ the following four steps [13]: matching cost computation, cost aggregation, disparity computation, and disparity refinement.

Cost aggregation refers to summing or averaging the matching cost in the support region of each pixel, which directly influences the computing efficiency and accuracy of the algorithm. It is one of the most important steps and also the primary focus of many studies. Filtering-based cost aggregation methods are currently adopted by most local algorithms, in which cost aggregation is interpreted as the filtering of the matching cost volume. Compared with other filtering methods, guided image filtering (GIF) [14] is usually the preferred approach, owing to its superior filtering effect and computational efficiency.

Hosni et al. [15] were the first to apply GIF to cost aggregation, achieving good results. Various cost aggregation methods have been proposed on the basis of this approach. Based on weighting GIF [16], Hong et al. [17] proposed a cost volume filtering method, in which the adaptive weight based on the local variance is used to control linear coefficients according to local texture features; this yields better performance. Kordelas et al. [18] proposed a content-based GIF, in which two rectangular support regions with different sizes are employed. The support region was selected according to the texture homogeneity of the local area around the pixels. In order to improve the accuracy of object edges and areas with discontinuous disparities in images and to reduce the noise in the matching targets, Hamzah et al. [19] proposed an adaptive support weight based on iterative GIF. Moreover, Wu et al. [20] proposed a strategy to fuse GIF and MST (minimum spanning tree) filter, which embedded the local support region-based GIF into the MST filter based on the whole image; it improved robustness of textureless and highly textured regions. Zhu et al. [21] introduced the weight into the energy function of GIF, thus considering the entire image as a support region. This resulted in better matching accuracy.

Adaptive cross-region-based guided image filtering (ACR-GIF) is a cost aggregation method adopted by many local stereo matching algorithms currently [22–26]. However, the weights of points in the adaptive cross-region [27] (ACR) are generally not taken into account, which affects the accuracy of results. The main contributions of this paper to address this issue are summarized as follows:

(1) An improved cost aggregation method is proposed. Firstly, according to the structural characteristic of the ACR, the orthogonal weight is proposed, and then the matching cost volume is filtered using ACR-GIF with orthogonal weights (ACR-GIF-OW).

(2) In order to improve the computational efficiency of the proposed method, an efficient weighted aggregation computing method based on orthogonal weights is proposed.

(3) Combining ACR-GIF-OW with our recently proposed matching cost computation and disparity refinement methods, a local stereo matching algorithm is proposed.

The main contributions of this paper are different from our previous paper [28]. In the previous paper, our work focused on matching cost computation and disparity refinement, so a gradient calculation method and a multistep refinement method based on ACR were proposed. However, the main contributions of this paper are mainly related to cost aggregation as mentioned above.

The remainder of this paper is structured as follows. The related work using ACR-GIF is discussed in Section 2. The proposed stereo matching algorithm using ACR-GIF-OW is described in Section 3. Experimental results and discussions are presented in Section 4 and finally, Section 5 concludes this paper.

2. Related Work

In this section, we discuss cost aggregation methods using ACR-GIF as they are more relevant to our proposed approach. GIF is analyzed first since ACR-GIF is an improved version of GIF.

GIF utilizes support regions with fixed shapes and sizes. As a result, the matching accuracy of textureless areas or regions with discontinuous disparities in images is affected. Therefore, it has become imperative to acquire adaptive support regions according to different regions and structures in images.

Due to its simple implementation and high computing efficiency, many local stereo matching algorithms use the ACR as the support region. Yang et al. [22] established the rectangular ACR where the boundary is determined by the endpoint of the support arm, and the endpoint is the spot where the color difference exceeds the threshold and is closest to the center pixel in the given direction. This approach ensures that most pixels in the support region are similar to the center pixel. Zhu et al. [23] adopted both color difference and distance as conditions in constructing the rectangular ACR. The support arm extends when both conditions are met. This method can effectively reduce outliers from occluded regions or areas with discontinuous depths in the support region.

In addition to the rectangular ACR, the arbitrary-shaped ACR has also been commonly employed. Xu et al. [24] used

the arbitrary-shaped ACR-based GIF, where the length of the support arm is determined by the color similarity of RGB channels. Zhu et al. [25] added an exponential threshold to the decision rule for arm length in order to process textureless regions; subsequently, they proposed an adaptive threshold using image variance to address the issue of the support region not being constructed in the same image structure when the brightness changes. Furthermore, in order to improve the accuracy of textureless regions, Yan et al. [26] proposed a decision rule for arm length by using dual constraint linear variable thresholds to construct the arbitrary-shaped ACR.

The above works all put emphasis on how to construct the ACR; the weight of each point in the ACR is not considered. Different from them, our method takes the orthogonal weight of each point in the ACR into consideration.

3. The Proposed Algorithm

The stereo matching algorithm proposed in this paper mainly includes five steps: (1) input image preprocessing, (2) matching cost computation, (3) cost aggregation using ACR-GIF-OW, (4) disparity computation, and (5) disparity refinement. A flowchart of the proposed algorithm is shown in Figure 1. Each step is detailed in the following sections.

3.1. Input Image Preprocessing. Since guided images are required in matching cost computation, it is necessary to preprocess the rectified input image. GIF has edge-preserving feature and good smoothing effect, and the computation time is independent of the size of the support region. Hence, guided images are obtained by using GIF.

3.2. Matching Cost Computation. To render the model more robust and achieve more accurate results, we adopt a matching cost computation method that combines the absolute differences (AD), census transformation [29], and the gradient.

AD is computed by using information on RGB channels according to the following equation:

$$C_{\mathrm{AD}}(m,d) = \frac{1}{3}\sum_{c\in\{R,G,B\}}\left|I_{\mathrm{Left}}(m,c) - I_{\mathrm{Right}}(n,c)\right|, \tag{1}$$

where $I_{\mathrm{Left}}(m,c)$ is the value of point $m(x_m, y_m)$ on the channel c of the left image and $I_{\mathrm{Right}}(n,c)$ is the value of the corresponding point $n(x_m - d, y_m)$ on the channel c of the right image, when the disparity is d.

Census transformation firstly compares the gray value of the center point with those of other points in the window and utilizes 0 or 1 to represent the result; then, the results are linked to form a binary bit string. This can be formulated as

$$\begin{aligned} S_{\mathrm{cen}} &= \mathop{\otimes}_{i\in W_k}\xi[I(k), I(i)], \\ \xi[I(k), I(i)] &= \begin{cases} 1, & I(k) > I(i), \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

Here, $\otimes$ represents a bitwise connection operation, W_k is the window centered on the point k, i is an arbitrary point in W_k, and $I(i)$ and $I(k)$ are gray values of points i and k, respectively.

Subsequently, the Hamming distance of binary bit strings between corresponding points is computed to measure the similarity between them. We assume that $S_{\mathrm{cen}\,L}(m)$ is the binary bit string of point m in the left image and $S_{\mathrm{cen}\,R}(n)$ is the binary bit string of the corresponding point n in the right image when the disparity is d. The Hamming distance between m and n can be expressed as

$$C_{\mathrm{cen}}(m,d) = \mathrm{Hamming}\left[S_{\mathrm{cen}L}(m), S_{\mathrm{cen}R}(n)\right]. \tag{3}$$

The gradient is calculated by our recently proposed method [28]. It combines the RGB gradient of the input image and the guided image to compute the gradient in x and y directions, respectively. The expressions are as follows:

$$\begin{aligned} C_{gx}(m,d) &= \frac{1}{3}\sum_{c\in\{R,G,B\}}\left[\left|gx^{I}_{\mathrm{Left}}(m,c) - gx^{I}_{\mathrm{Right}}(n,c)\right| + \left|gx^{G}_{\mathrm{Left}}(m,c) - gx^{G}_{\mathrm{Right}}(n,c)\right|\right], \\ C_{gy}(m,d) &= \frac{1}{3}\sum_{c\in\{R,G,B\}}\left[\left|gy^{I}_{\mathrm{Left}}(m,c) - gy^{I}_{\mathrm{Right}}(n,c)\right| + \left|gy^{G}_{\mathrm{Left}}(m,c) - gy^{G}_{\mathrm{Right}}(n,c)\right|\right], \end{aligned} \tag{4}$$

where gx and gy represent the gradient in x and y directions, respectively. The superscripts I and G, respectively, represent the input image and the guided image.

By weighted fusion of the above-mentioned approaches, the matching cost computation function is acquired as follows:

$$\begin{aligned} C(m,d) = 4 &- \exp\left[-\frac{C_{\mathrm{AD}}(m,d)}{\lambda_{\mathrm{AD}}}\right] - \exp\left[-\frac{C_{\mathrm{cen}}(m,d)}{\lambda_{\mathrm{cen}}}\right] \\ &- \exp\left[-\frac{C_{gx}(m,d)}{\lambda_{gx}}\right] - \exp\left[-\frac{C_{gy}(m,d)}{\lambda_{gy}}\right]. \end{aligned} \tag{5}$$

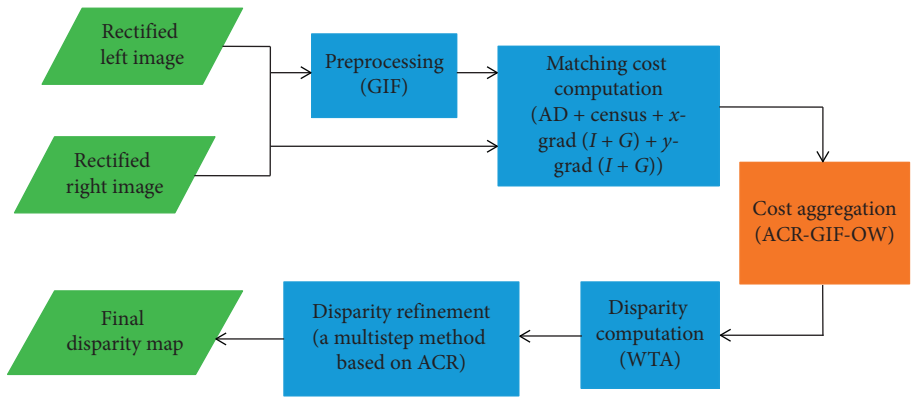


Figure 1: Flowchart of the proposed algorithm.

Here, λ_{AD}, λ_{cen}, λ_{gx}, and λ_{gy} are the weight of AD, census transformation, the gradient in x direction, and the gradient in y direction, respectively; C_{AD}, C_{cen}, C_{gx}, and C_{gy} are the matching cost of the corresponding approach.

3.3. Cost Aggregation Using ACR-GIF-OW

3.3.1. ACR Construction. Points with similar color in the support region may arise from the same structure of the image; thus, these points have similar disparity [27]. In order to ensure that only points with similar color are included in the support region, the ACR is adopted. Double thresholds of the distance and color difference are used to restrict the arm length [30]. The decision rules are as follows:

(1) $D_c(p, p_e) < C_1$ and $D_c(p_e, p_n) < C_1$

(2) $D_d(p, p_e) < L_1$

(3) If $L_2 < D_d(p, p_e) < L_1$, $D_c(p, p_e) < C_2$ and $D_c(p_e, p_n) < C_2$

p_e is an arbitrary point on the support arm with center point p, p_{n} is the point preceding p_e in the direction of the support arm, and $D_c(p, p_e) = \max_{c\in\{R,G,B\}} |I_c(p) - I_c(p_e)|$, $D_d(p, p_e) = |p - p_e|$. C_1, C_2, L_1, and L_2 are thresholds, and $C_2 < C_1$ and $L_2 < L_1$.

When the above rules are fulfilled, the center point is considered to be the starting point that expands in four directions, and the expansion stops when one of the decision rules is not satisfied. The ACR $R(p)$ can be expressed as the union of all horizontal support arms $H(q)$, whose center point q is on the vertical support arm of point p, as shown in Figure 2.

3.3.2. The Orthogonal Weight Calculation. According to the construction process and structural feature of the ACR, we detect a horizontal path at first and then a vertical one when observed from any point to the center point. Therefore, the weight of each point relative to the center point can be computed by multiplying the weight between the adjacent points on the path [31]. Since the weight can be decomposed into two parts (the horizontal weight and the vertical weight), we name it as the orthogonal weight, as shown in Figure 3.

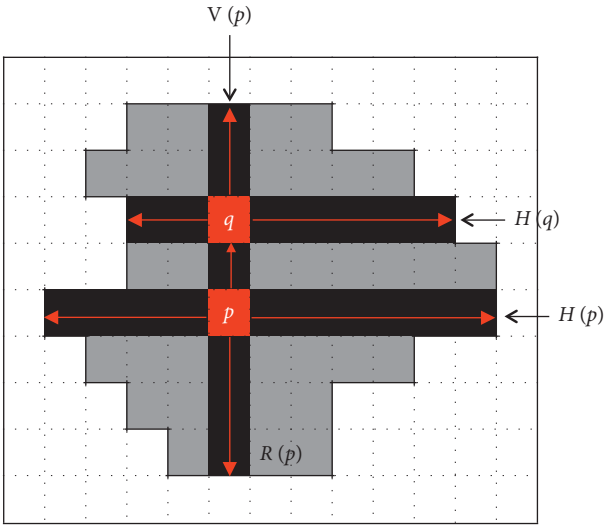


Figure 2: Schematic view of the ACR.

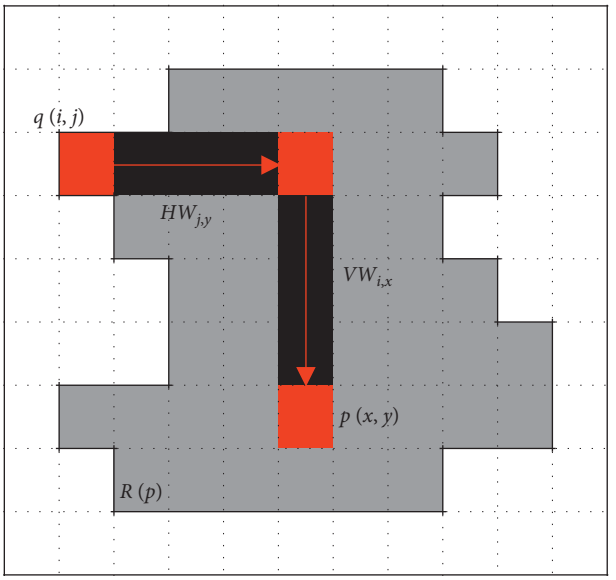


Figure 3: Schematic of the orthogonal weight.

Here, $q(i, j)$ is an arbitrary point in the $R(p)$, $p(x, y)$ is the center point, and $\mathrm{HW}_{j,y}$ and $\mathrm{VW}_{i,x}$ represent the horizontal and vertical weights of point q relative to point p, respectively.

The orthogonal weight can be computed by multiplying the horizontal and vertical weights, and thus, the orthogonal weight of q can be expressed as

$$W(q, p) = \mathrm{HW}_{j,y} \cdot \mathrm{VW}_{i,x}. \tag{6}$$

To obtain the calculations associated with $\mathrm{HW}_{j,y}$ and $\mathrm{VW}_{i,x}$ conveniently, we construct the weight matrix of horizontal adjacent points, WH, and the weight matrix of vertical adjacent points, WV. The information on RGB channels is used to compute the weights of adjacent points, as shown in the following formulae:

$$\mathrm{WH}(x, y) = f\left(\sum_{c\in\{R,G,B\}} \left|I_c(h, w) - I_c(h, w+1)\right|\right), \tag{7}$$

$$\mathrm{WV}(x, y) = f\left(\sum_{c\in\{R,G,B\}} \left|I_c(h, w) - I_c(h+1, w)\right|\right), \tag{8}$$

where h and w, respectively, represent the row and column indices of image I. The expression of function $f(t)$ in equations (7) and (8) is shown as follows:

$$f(t) = \begin{cases} 1, & t < \delta, \\ \exp\left(\dfrac{1}{\gamma}\right), & t \geq \delta, \end{cases} \tag{9}$$

where parameters δ and γ are constants. The purpose of introducing function $f(t)$ is to avoid the problem that when there is a significant local difference in the intensity between adjacent pixels on the path, information tends to get lost [21].

After calculating the matrices WH and WV, the horizontal and vertical weights of q can be computed as follows:

$$\mathrm{HW}_{j,y} = \begin{cases} \prod_{k=\min(j,y)}^{\max(j,y)-1} \mathrm{WH}(i, k), & j \neq y, \\ 1, & j = y, \end{cases} \tag{10}$$

$$\mathrm{VW}_{i,x} = \begin{cases} \prod_{k=\min(i,x)}^{\max(i,x)-1} \mathrm{WV}(k, y), & i \neq x, \\ 1, & i = x. \end{cases} \tag{11}$$

According to equation (10), another recursive form of computing the horizontal weight is formulated as

$$\mathrm{HW}_{j,y} = \begin{cases} \mathrm{WH}(i, j) \cdot \mathrm{HW}_{j+1,y}, & y - j > 1, \\ \mathrm{WH}(i, y-1), & y - j = 1. \end{cases} \tag{12}$$

Similarly, according to equation (11), another recursive form of computing the vertical weight is formulated as

$$\mathrm{VW}_{i,x} = \begin{cases} \mathrm{WV}(i, y) \cdot \mathrm{VW}_{i+1,x}, & x - i > 1, \\ \mathrm{WV}(x-1, y), & x - i = 1. \end{cases} \tag{13}$$

According to equations (12) and (13), $\mathrm{HW}_{j,y}$ and $\mathrm{VW}_{i,x}$ can be computed from the center point in both directions, such that the weight of the previous point can be utilized, and only one multiplication is required in each computation.

3.3.3. The Weighted Aggregation Computing Method. On the basis of the discussion in Section 3.3.2 and inspired by the orthogonal integral image [27], we propose a weighted aggregation computing method based on orthogonal weights. Based on the feature that the orthogonal weight can be decomposed; the process of weighted aggregation is decomposed into two orthogonal directions of one-dimensional weighted aggregation as follows:

(1) The weighted sum of the horizontal support arm for each point is computed. To improve the computing efficiency for any point $r(i, j)$ in the image, the weighted sum of the left and right support arms is separately computed as follows:

$$\begin{aligned} \mathrm{WSL}(i, j) &= \sum_{k=\mathrm{beg\_}h}^{j} \mathrm{HW}_{k,j} \cdot \mathrm{img}(i, k) = \sum_{k=\mathrm{beg\_}h}^{j-1} \mathrm{HW}_{k,j} \cdot \mathrm{img}(i, k) + \mathrm{img}(i, j), \\ \mathrm{WSR}(i, j) &= \sum_{k=j}^{\mathrm{end\_}h} \mathrm{HW}_{k,j} \cdot \mathrm{img}(i, k) = \sum_{k=j+1}^{\mathrm{end\_}h} \mathrm{HW}_{k,j} \cdot \mathrm{img}(i, k) + \mathrm{img}(i, j), \end{aligned} \tag{14}$$

where $\mathrm{WSL}(i, j)$ and $\mathrm{WSR}(i, j)$ are the weighted sum of the left and right support arms of point r, respectively; beg_h and end_h are the beginning and ending positions of the horizontal support arm for point r, respectively; img is a single-channel image. For RGB images, one of the channels is selected according to the need for calculation.

(2) The weighted sum of the horizontal support arm for point r is computed and stored in WSHas follows:

$$\mathrm{WSH}(i, j) = \mathrm{WSL}(i, j) + \mathrm{WSR}(i, j) - \mathrm{img}(i, j). \tag{15}$$

(3) Based on WSH, the weighted sum of the vertical support arm for each point is computed. Similarly, to improve the computing efficiency, the weighted sum

of the up and bottom support arms for the center point $p(x, y)$ is, respectively, computed as follows:

$$\begin{aligned} \mathrm{WSU}(x,y) &= \sum_{k=\mathrm{beg\_}v}^{x} \mathrm{VW}_{k,x} \cdot \mathrm{WSH}(k,y) = \sum_{k=\mathrm{beg\_}v}^{x-1} \mathrm{VW}_{k,x} \cdot \mathrm{WSH}(k,y) + \mathrm{WSH}(x,y), \\ \mathrm{WSD}(x,y) &= \sum_{k=x}^{\mathrm{end\_}v} \mathrm{VW}_{k,x} \cdot \mathrm{WSH}(k,y) = \sum_{k=x+1}^{\mathrm{end\_}v} \mathrm{VW}_{k,x} \cdot \mathrm{WSH}(k,y) + \mathrm{WSH}(x,y). \end{aligned} \tag{16}$$

Here, $\mathrm{WSU}(x, y)$ and $\mathrm{WSD}(x, y)$ are the weighted sum of the up and bottom support arms for point p, respectively; beg_v and end_v are the beginning and ending positions of the vertical support arm for point p, respectively.

(4) The weighted aggregation result of the ACR centered at point p is obtained as follows:

$$\mathrm{WACR}(x,y) = \mathrm{WSU}(x,y) + \mathrm{WSD}(x,y) - \mathrm{WSH}(x,y). \tag{17}$$

3.3.4. ACR-GIF-OW. On the basis of the computing method described in Section 3.3.3, we adopt ACR-GIF-OW as the cost aggregation method. Since the color image has more obvious edge protection effects [14], we select the color image as the guidance image. We denote the guidance image as I and the filtering input image as the matching cost volume C. The linear model coefficients a_p and b_p can be acquired by minimizing the weighted local energy function that is formulated as

$$E(a_p, b_p) = \sum_{q \in R_p} W(q,p) \cdot \left(\left(a_p \cdot I(q) + b_p - C(q,d) \right)^2 + \varepsilon \cdot a_p^2 \right), \tag{18}$$

where R_p is the ACR centered at pixel p, ε is a regularization parameter, and $W(q, p)$ is the orthogonal weight of point q defined in equation (6).

The solution to this equation is given as

$$\begin{aligned} a_p &= \left(\sum_p + \varepsilon U \right)^{-1} \left(\frac{1}{\sum_{q \in R_p} W(p,q)} \sum_{q \in R_p} W(p,q) \cdot I(q) \cdot C(q,d) - \mu_p \cdot \overline{C}_p \right) \\ &= \left(\sum_p + \varepsilon U \right)^{-1} \begin{pmatrix} \frac{1}{\sum_{q \in R_p} W(p,q)} \sum_{q \in R_p} W(p,q) \cdot I_R(q) \cdot C(q,d) - \mu_p^R \cdot \overline{C}_p \\ \frac{1}{\sum_{q \in R_p} W(p,q)} \sum_{q \in R_p} W(p,q) \cdot I_G(q) \cdot C(q,d) - \mu_p^G \cdot \overline{C}_p \\ \frac{1}{\sum_{q \in R_p} W(p,q)} \sum_{q \in R_p} W(p,q) \cdot I_B(q) \cdot C(q,d) - \mu_p^B \cdot \overline{C}_p \end{pmatrix}, \\ b_p &= \overline{C}_p - a_p^T \mu_p = \overline{C}_p - a_p^T \cdot \begin{pmatrix} \mu_p^R \\ \mu_p^G \\ \mu_p^B \end{pmatrix}. \end{aligned} \tag{19}$$

Here, $\mu_p^c = (\sum_{q \in R_p} (W(p,q) \cdot I_c(q)) / \sum_{q \in R_p} W(p,q))$ $(c \in \{R, G, B\})$, $\overline{C}_p = \sum_{q \in R_p} (W(p,q) \cdot C(q,d)) / \sum_{q \in R_p} W(p,q)$, $\sum_p$ is the 3×3 covariance matrix of I in R_p, and U is an 3×3 identity matrix.

The linear model is then used to compute the filtered result, which is also the result of cost aggregation, as shown as follows:

$$\mathrm{CA}(q,d) = \overline{a}_q^T I_q + \overline{b}_q, \tag{20}$$

where $\overline{a}_q = (1/\sum_{p\in R_q} W(p,q))\sum_{p\in R_q} a_p$, $\overline{b}_q = (1/\sum_{p\in R_q} W(p,q))\sum_{p\in R_q} b_p$, and CA is the matching cost volume after cost aggregation.

The comparison of results after cost aggregation using ACR-GIF [28] without/with orthogonal weights is shown in Figure 4.

3.4. Disparity Computation. We use the winner-take-all strategy [13] for the disparity computation, in which the disparity corresponding to the minimum matching cost of each point in CA is selected as the initial disparity. The expression is given by

$$d_{\text{ini}}(m) = \arg\min_{0\le d\le d_{\max}} [\text{CA}(m,d)]. \tag{21}$$

Here, $d_{\text{ini}}(m)$ is the initial disparity of point m.

3.5. Disparity Refinement. There are many outliers in the initial disparity map that need to be detected and corrected by disparity refinement. In this study, a multistep refinement method [28] proposed by us recently is adopted, and each step has been elucidated in the following sections.

3.5.1. Left-Right Consistency Check and Outlier's Classification. The left-right consistency check judges whether the disparities of two points satisfy the condition given in the following equation:

$$\left|d^L_{\text{ini}}(x_0,y_0) - d^R_{\text{ini}}\left(x_0 - d^L_{\text{ini}}(x_0,y_0), y_0\right)\right| \le 1, \tag{22}$$

where d^L_{ini} and d^R_{ini} represent the initial disparity map of the left and right images, respectively; x_0 and y_0 are the point indices.

Subsequently, the detected outliers are divided into two classes: one has the corresponding point in the right image, and the other does not exist the corresponding point in the right image. The first class is called the corresponding outlier, and the second is called the no-corresponding outlier. The steps described below correct the two classes separately.

3.5.2. ACR Voting. To replace the disparities of outliers with that of reliable points, we first use ACR voting, in which the total number of votes of reliable points and the highest number of votes among different disparities are counted. We then consider the following conditions:

$$N_T > N, \tag{23}$$

$$\frac{N_{\max}}{N_T} > P. \tag{24}$$

Here, N_T is the total number of votes, $N_{\max}$ is the highest number of votes among different disparities, and N and P are thresholds. If both equations (23) and (24) are satisfied, the disparity corresponding to the highest number of votes is used to replace the outlier's disparity. Meanwhile, the outlier is marked as reliable. In order to deal with as many outliers as possible, this step is iterated five times.

3.5.3. ACR Four-Direction Propagation Interpolation. For corresponding outliers, the nearest reliable points are found in their own ACR along the directions of the four support arms. The corresponding disparities are separately marked as d_{RL}, d_{RR}, d_{RU}, and d_{RD}. Then, the disparities of these outliers are replaced by equation (25), and they are marked as reliable points.

$$d_R = \begin{cases} d_{h\min}, & \text{if } d_{\text{RL}} \text{ and } d_{\text{RR}} \text{ existing,} \\ d_{v\min}, & \text{if } d_{\text{RU}} \text{ and } d_{\text{RD}} \text{ existing,} \\ \dfrac{(d_{h\min} + d_{v\min})}{2}, & \text{if } \left|d_{h\min} - d_{v\min}\right| \le 2, \\ \text{not changed}, & \text{otherwise.} \end{cases} \tag{25}$$

Here, $d_{h\min} = \min(d_{\text{RL}}, d_{\text{RR}})$ and $d_{v\min} = \min(d_{\text{RU}}, d_{\text{RD}})$. For dealing with as many outliers as possible, this step is iterated three times.

3.5.4. Two-Direction Propagation Interpolation. For remaining corresponding outliers, the nearest reliable points are found along the left and right directions, and the corresponding disparities are recorded as d_l and d_r, respectively. Then, the disparities of these outliers are replaced by equation (26), and they are marked as reliable points.

$$d_R = \begin{cases} d_{h\min}, & \text{if } d_l \text{ and } d_r \text{ existing,} \\ \text{not changed}, & \text{otherwise.} \end{cases} \tag{26}$$

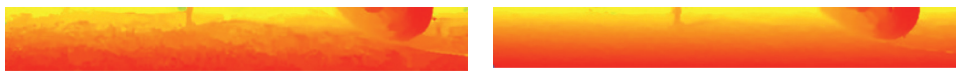
Here, $d_{h\min} = \min(d_l, d_r)$.

3.5.5. No-Corresponding Outliers Interpolation. After the above-mentioned steps, the remaining outliers are mainly no-corresponding outliers. Since such outliers usually appear in the leftmost area of the image, we use one-direction propagation interpolation; that is, the nearest reliable point is found along the right side of the outlier. The disparity of the outlier is then replaced and the outlier is marked as reliable.

3.5.6. Subpixel Refinement. To reduce the error caused by disparity discontinuity, an approach based on quadratic polynomial interpolation is used for subpixel refinement as follows:

$$d_{\text{sub}} = d_R - \frac{\text{CA}(m,d_R+1) - \text{CA}(m,d_R-1)}{2[\text{CA}(m,d_R+1) + \text{CA}(m,d_R-1) - 2\text{CA}(m,d_R)]}, \tag{27}$$

where d_R is the disparity of point m after previous steps; $\text{CA}(m,d_R+1)$ and $\text{CA}(m,d_R-1)$ are cost aggregation results of point m when the disparity is d_R+1 and d_R-1,



(a) (b)

Figure 4: Comparison of results after cost aggregation. (a) Using ACR-GIF without orthogonal weights. (b) Using ACR-GIF with orthogonal weights.

respectively. At last, the 3×3 median filter is used to smooth the disparity result.

4. Experimental Results and Discussions

We carried out our experiments on Middlebury evaluation platform [32], whose dataset includes two parts: training sets and test sets. Every part has fifteen image pairs with different resolution at least 1300×1100 pixels. Owing to the high resolution, complicated scene structure, and different lighting or exposure conditions, the results of the dataset can actually reflect the robustness and accuracy of the algorithm.

The parameters and thresholds in the proposed stereo matching algorithm are set as follows: $\delta = 1/510$, $\gamma = -3$, $\lambda_{\mathrm{AD}} = 30/255$, $\lambda_{\mathrm{cen}} = 45/255$, $\lambda_{gx} = 5/255$, $\lambda_{gy} = 15/255$, $\varepsilon = 0.01^2$, $C_1 = 15/255$, $C_2 = 12/255$, $L_1 = \max(H, W)/30$, $L_2 = \max(H, W)/60$, $N = 40$, and $P = 0.5$, where H and W represent the height and width of the input image, respectively. Among them, the values of λ_{AD}, λ_{cen}, λ_{gx}, and λ_{gy} are referenced from [23], the values of C_1, C_2, L_1, L_2, N, and P are referenced from [28], the value of ε is the same as in [15].

4.1. Efficiency of the Proposed Weighted Aggregation Computing Method. In order to verify the effectiveness of the proposed weighted aggregation computing method, the computation time of straightforward computing (the weight of each point in the ACR is computed according to equations (12) and (13) and then summed by traversal) and the computing method described in Section 3.3.3 are compared. The experimental environment used is Matlab 2018b, and the computer configuration is Intel Core i7-8750H CPU and 16G memory. The results of training sets are shown in Figure 5.

The chart illustrates that the computation time of the proposed computing method is obviously less than that required for straightforward computing. Among them, the computation time for Shelves is reduced by a maximum of 80.2%; the computation time for Recycle, Vintage, Jadeplant, and Adirondack is, respectively, reduced by 79.9%, 79.1%, 78.9%, and 78%. Owing to the relatively low resolution and disparity level, the percentage reduction of Teddy and ArtL is comparatively low, that is, 64.6% and 55.6%, respectively. The computation time is reduced by 72.7% on average. The above data indicate that, compared with straightforward computing, the proposed computing method can effectively reduce computing time and improve computing efficiency.

4.2. Comparison of ACR-GIF and ACR-GIF-OW. To verify the effect of the proposed cost aggregation method, two stereo matching algorithms that, respectively, adopt ACR-GIF used in [28] and ACR-GIF-OW for cost aggregation are compared in terms of their disparity result and time overhead. Except for cost aggregation, all other steps of two algorithms are identical.

4.2.1. Comparison of Disparity Results. The metric bad 2.0 is used to quantitatively evaluate the accuracy of disparity results. It is the default metric of Middlebury evaluation platform that represents the percentage of bad pixels with disparity errors greater than 2.0 pixels. The results of the training sets are shown in Figure 6.

As observed in Figure 6(a), in nonoccluded regions, except Shelves and Teddy, the values of bad 2.0 for the remaining images are reduced to varying degrees. Among them, the value of Motorcycle and MotorcycleE can be reduced by more than 40%; the value of Adirondack, PlaytableP, and Recycle can be reduced by more than 30%; the value of Piano, Pipes, and Vintage can be reduced by more than 20%. From Figure 6(b), we can see that, in all regions, except ArtL, Shelves, and Teddy, the values of bad 2.0 for the remaining images are also reduced to varying degrees. Among them, the values of Motorcycle and MotorcycleE are reduced by more than 30%; the values of Adirondack, PlaytableP, and Recycle are reduced by more than 20%; the values of Piano, Playtable, and Vintage are reduced by more than 15%.

Figure 7 shows the comparison result of the bad 2.0 weighted average on training sets, which is obtained from Middlebury evaluation platform (the weight of each image is given by the platform). It can be seen from Figure 7 that the value of ACR-GIF-OW is evidently lower than that of ACR-GIF. The weighted average can be reduced by 24.1% and 16.3% in nonoccluded regions and all regions, respectively.

According to the above results, we can conclude that, compared to that of ACR-GIF, the accuracy of disparity results obtained by using ACR-GIF-OW is significantly superior in both nonoccluded regions and all regions.

Next, to compare the results of two algorithms more intuitively, we select three images from training sets and make comparisons of disparity maps and the corresponding error maps acquired by two algorithms, as shown in Figure 8.

We find that, in error maps of ACR-GIF-OW, black regions in red boxes are apparently smaller in area (the black color implies that the disparity errors are greater than 2.0 pixels), and these red boxes mainly correspond to weakly textured and textureless regions in the image. The above result indicates that ACR-GIF-OW can improve the disparity accuracy of these regions, thereby improving the overall disparity accuracy.

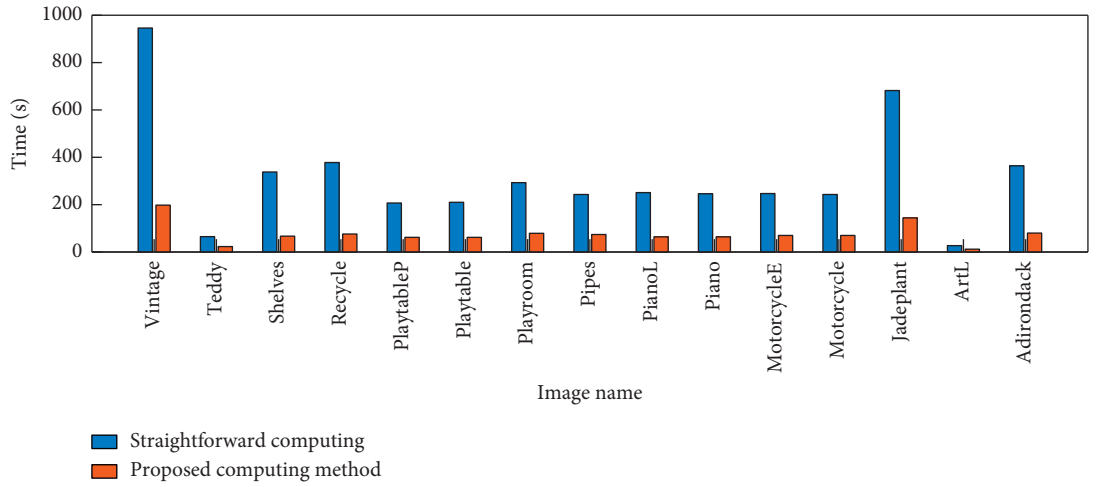


Figure 5: Comparison of the computation time between straightforward computing and the proposed computing method.

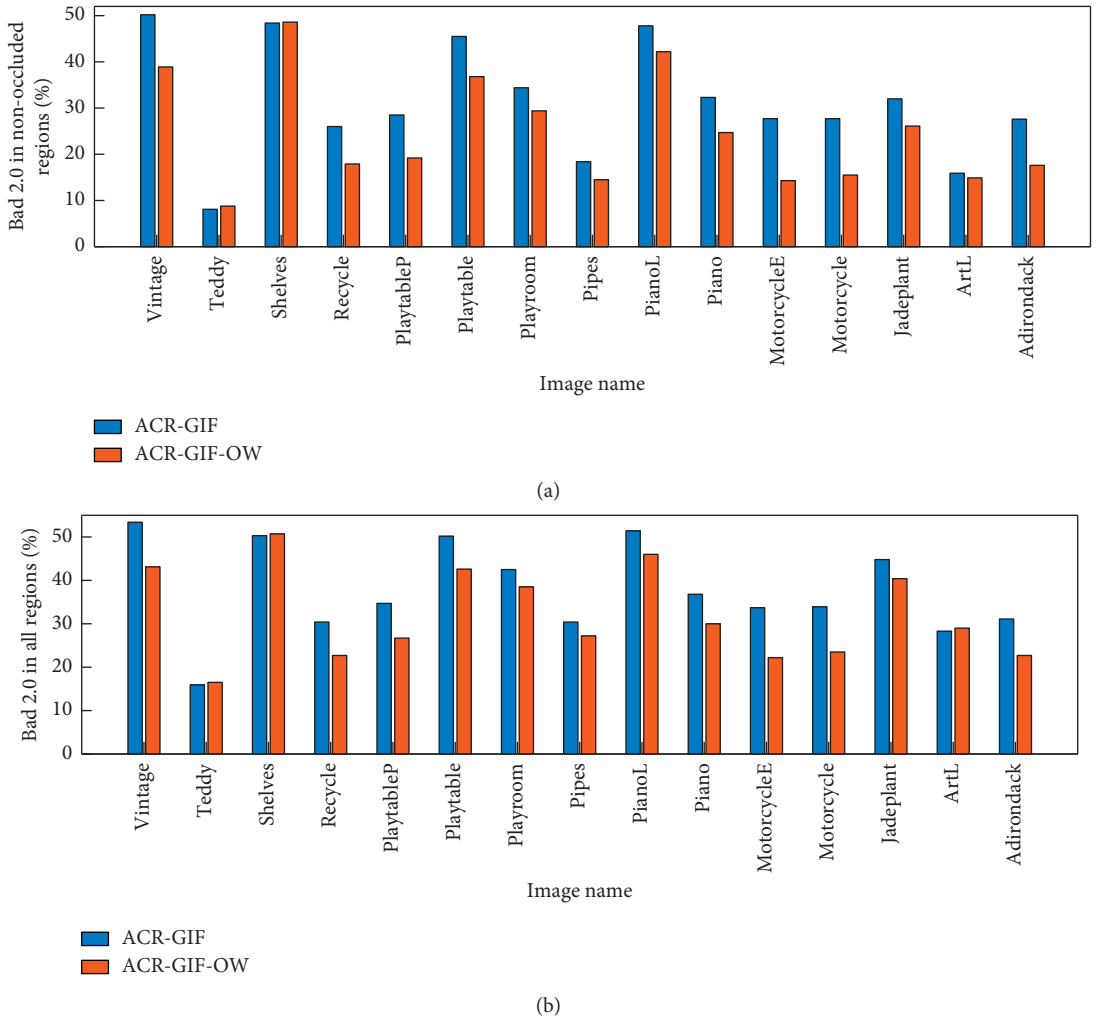


Figure 6: Comparison of bad 2.0 between ACR-GIF and ACR-GIF-OW: (a) bad 2.0 in nonoccluded regions; (b) bad 2.0 in all regions.

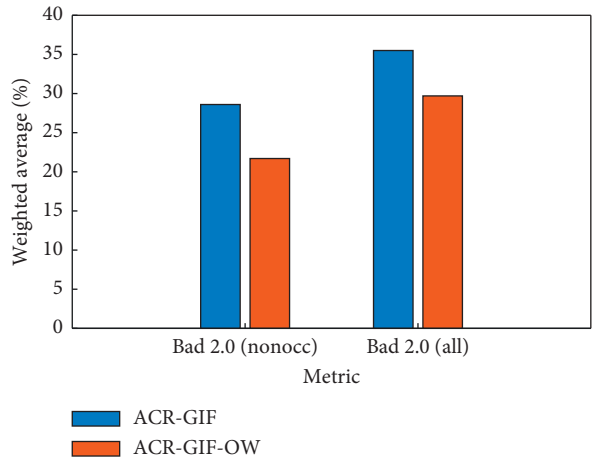


Figure 7: Comparison of the bad 2.0 weighted average between ACR-GIF and ACR-GIF-OW. nonocc: nonoccluded regions; all: all regions.

Furthermore, the performance on the low texture, repetitive pattern, plain color, and discontinue regions is compared, as shown in Figure 9.

It can be found in Figure 9 that the performance of ACR-GIF-OW is obviously better than ACR-GIF (the area of black region in the red box), especially in low texture, plain color, and discontinue regions.

4.2.2. Comparison of Time Overhead. Using the same experimental environment and computer as described in Section 4.1, the time overhead of ACR-GIF and ACR-GIF-OW is compared. The result is shown in Figure 10.

The chart illustrates that the time overhead of ACR-GIF-OW is more than that of ACR-GIF, owing to the weights' computation. Among them, the time overhead of Teddy has the lowest growth rate of 4.5%; the time overhead of Adirondack has the highest growth rate of 17.6%. The average growth rate of time overhead on training sets is 12.7%.

Combining the results acquired in Sections 4.2.1 and 4.2.2, we can conclude that comparing the increase in time overhead shows that ACR-GIF-OW exhibits an obvious improvement with regard to the disparity accuracy. Thus, considering both the accuracy and the time overhead, the proposed method is advantageous over ACR-GIF.

4.3. Analysis of Parameter Setting. Parameters δ and γ are two key parameters in the process of the orthogonal weight calculation. Figure 11 shows the effect of parameters δ and γ with different settings.

Figure 11(a) indicates that when $\delta > 1/510$, the disparity accuracy of both nonoccluded and all regions becomes worse; on the contrary, the accuracy remains unchanged. Figure 11(b) reveals that when $\gamma = -3$, the disparity accuracy can achieve its best in both nonoccluded and all regions. According to the above conclusions, the best setting of δ and γ is 1/510 and -3, respectively.

4.4. Effect of Each Step in the Proposed Algorithm. The proposed algorithm is composed of several steps. For analyzing how does each step affect the final result, except for bad 2.0, the weighted average of avgerr on training sets is also used. Avgerr is another metric that means average absolute error in pixels. The results after performing each step are shown in Figure 12.

Figure 12 presents the contribution of each step to the reduction of disparity errors in both nonoccluded and all regions. After performing CA, the value of bad 2.0 in nonoccluded and all regions can be decreased by 39.4% and 31.1%, respectively; the value of avgerr in nonoccluded and all regions can be decreased by 56.4% and 34.4%, respectively. After performing DR, the value of bad 2.0 can be, respectively, reduced by 27.7% and 22.9% in nonoccluded and all regions; the value of avgerr can be, respectively, reduced by 29.5% and 47.6% in nonoccluded and all regions.

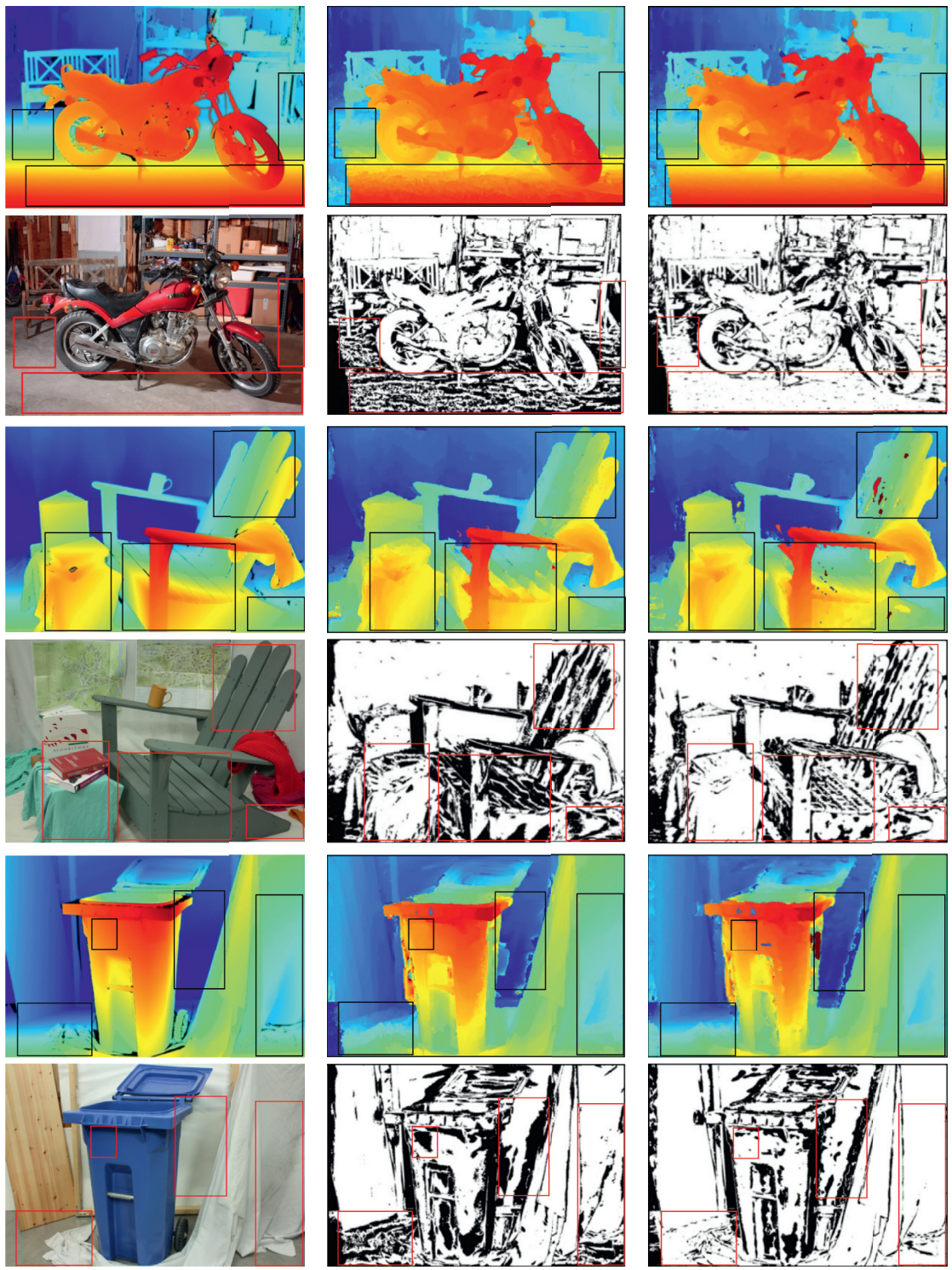
Moreover, Figure 13 shows the effect of each step in DR. The charts indicate that the contribution of each step is distinct for different metrics and regions. For the bad 2.0, Step 5 is the most effective. But for the avgerr, the errors are significantly reduced by Step 1. In all regions, Step 2 is more effective than in nonoccluded regions, and so are Steps 3 and 4. Thus, the combination of these steps guarantees a better result.

4.5. Comparison with State-of-the-Art Stereo Matching Algorithms

4.5.1. Comparison with Other Local Stereo Matching Algorithms. To verify the performance of the local stereo matching algorithm proposed in this paper, we select seven state-of-the-art local algorithms for comparison, namely, DAWA-F [33], FASW [20], IEBIMst [34], ADSM [35], DoGGuided [36], IGF [37], and ISM [38]. The disparity map comparison of five stereo images in Middlebury datasets is shown in Figure 14.

To make a quantitative comparison of disparity results, the metric bad 2.0 is employed again. The comparison results of whole datasets are shown in Tables 1 and 2, where the bold fonts indicate the best results. The weighted average shown in the last row is the weighted average of training sets and test sets. The weights are given by the Middlebury evaluation platform.

The results of Tables 1 and 2 indicate that, whether in nonoccluded regions or all regions, the number of the best results obtained by the proposed method is higher than other local algorithms, and the rest of the results are also relatively good. Besides, both of the weighted average values are the best as well. Furthermore, for image pairs with different illuminations like ArtL, PianoL, and DjembeL and with different exposures like MotorcycleE and Classroom2E, better results are acquired by the proposed algorithm. This demonstrates that the proposed algorithm has better robustness when the illumination or exposure changes for a pair of images. To summarize, the performance of the proposed algorithm is evidently better than those of the other seven state-of-the-art local algorithms.



(a) (b) (c)

Figure 8: Comparison of MotorcycleE, Adirondack, and Recycle. (a) Ground truth disparity maps and rectified left images. (b) Disparity maps and the corresponding error maps of ACR-GIF and (c) ACR-GIF-OW.

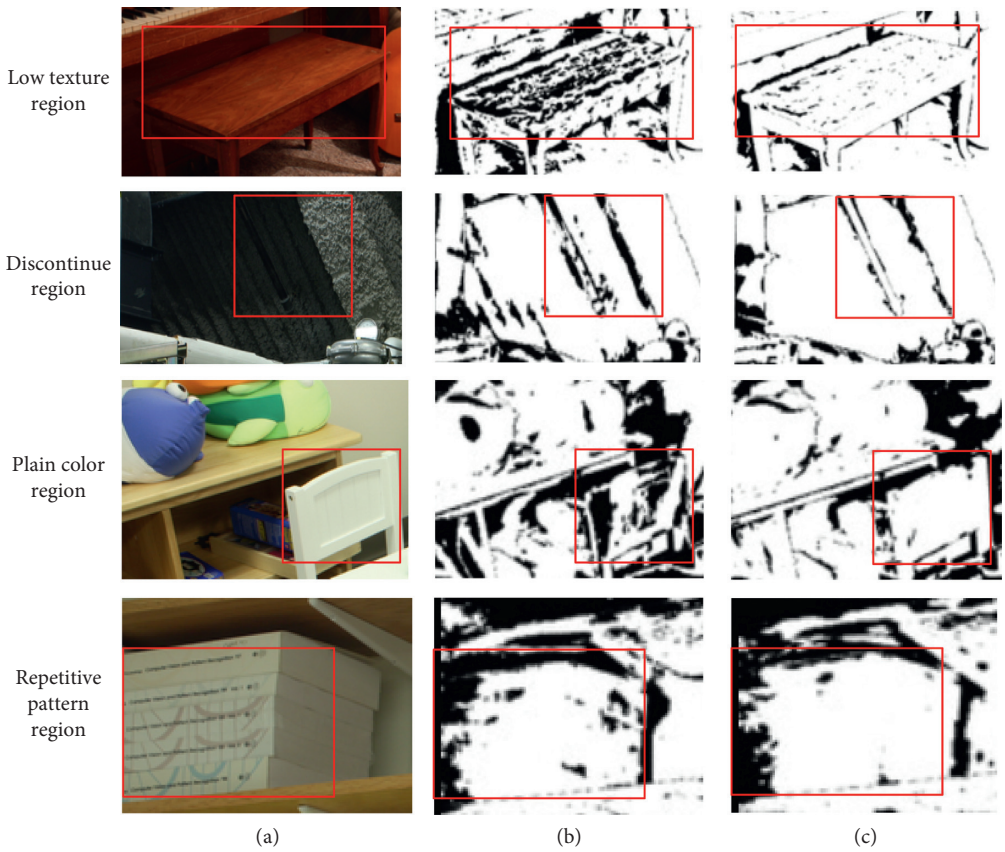


Figure 9: Comparison of low texture, repetitive pattern, plain color, and discontinue regions. (a) Left images. (b) Error maps of ACR-GIF. (c) Error maps of ACR-GIF-OW.

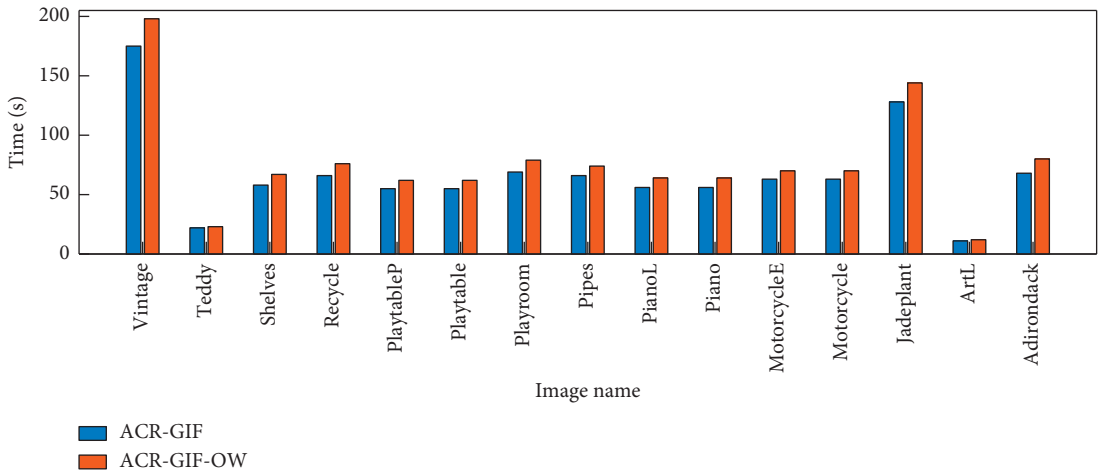


Figure 10: Comparison of time overhead between ACR-GIF and ACR-GIF-OW.

4.5.2. Comparison with Other Nonlocal Stereo Matching Algorithms. In order to make a more comprehensive comparison, except for local algorithms, we also select six state-of-the-art nonlocal algorithms for comparison, namely, DDL [39], LS_ELAS [40], TSGO [41], DSGCA [42], SIGMRF [43], and SPPSMNet [44]. Among them, DSGCA,

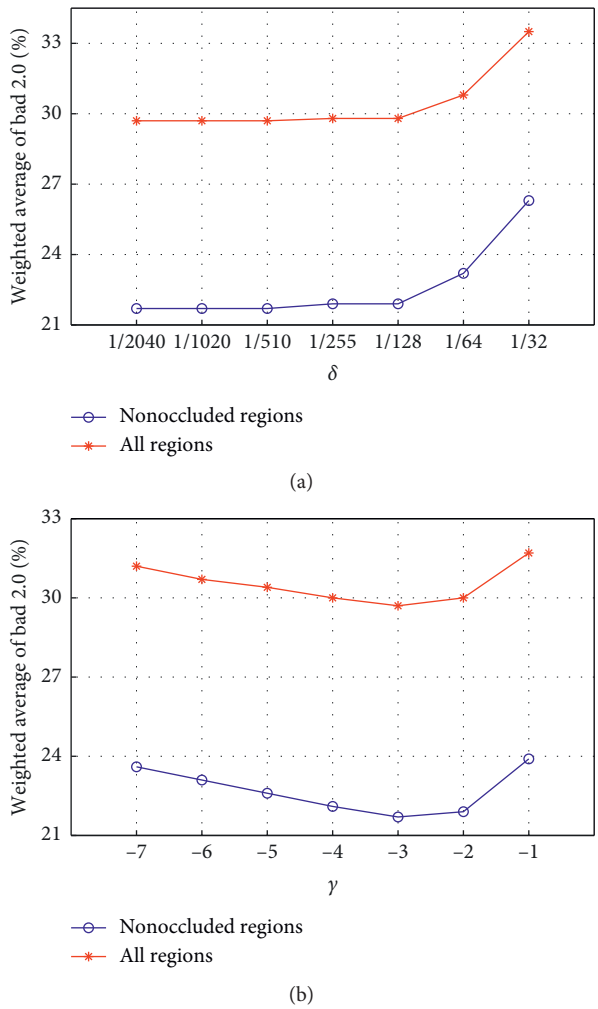


Figure 11: Effect of parameters (a) δ and (b) γ. The weighted average of bad 2.0 on training sets in both nonoccluded and all regions is listed.

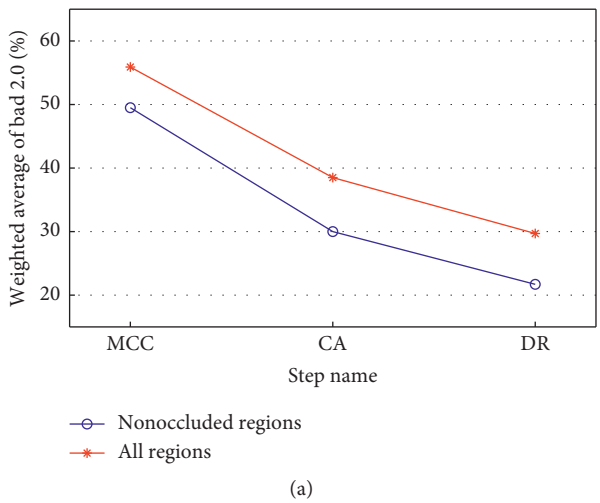
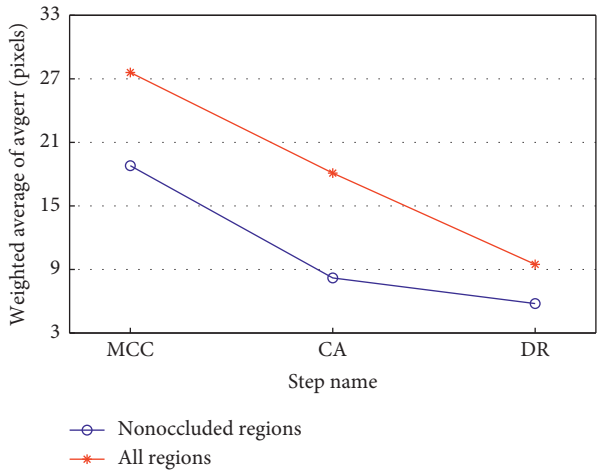
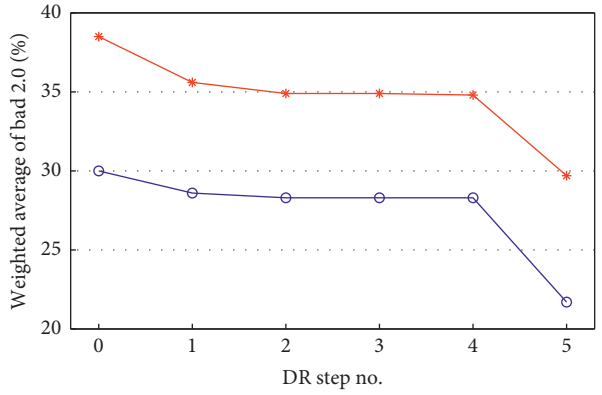


Figure 12: Continued.

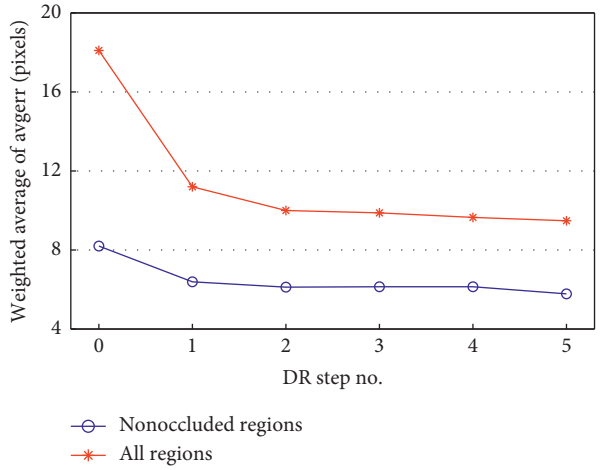


(b)

Figure 12: Weighted average on training sets in nonoccluded and all regions after performing each step: (a) bad 2.0 and (b) avgerr. MCC: matching cost computation, CA: cost aggregation, and DR: disparity refinement.



(a)



(b)

Figure 13: Weighted average on training sets in nonoccluded and all regions after performing each step of DR: (a) bad 2.0; (b) avgerr. 1: ACR voting, 2: ACR four-direction propagation interpolation, 3: two-direction propagation interpolation, 4: no-corresponding outliers interpolation, and 5: subpixel refinement.

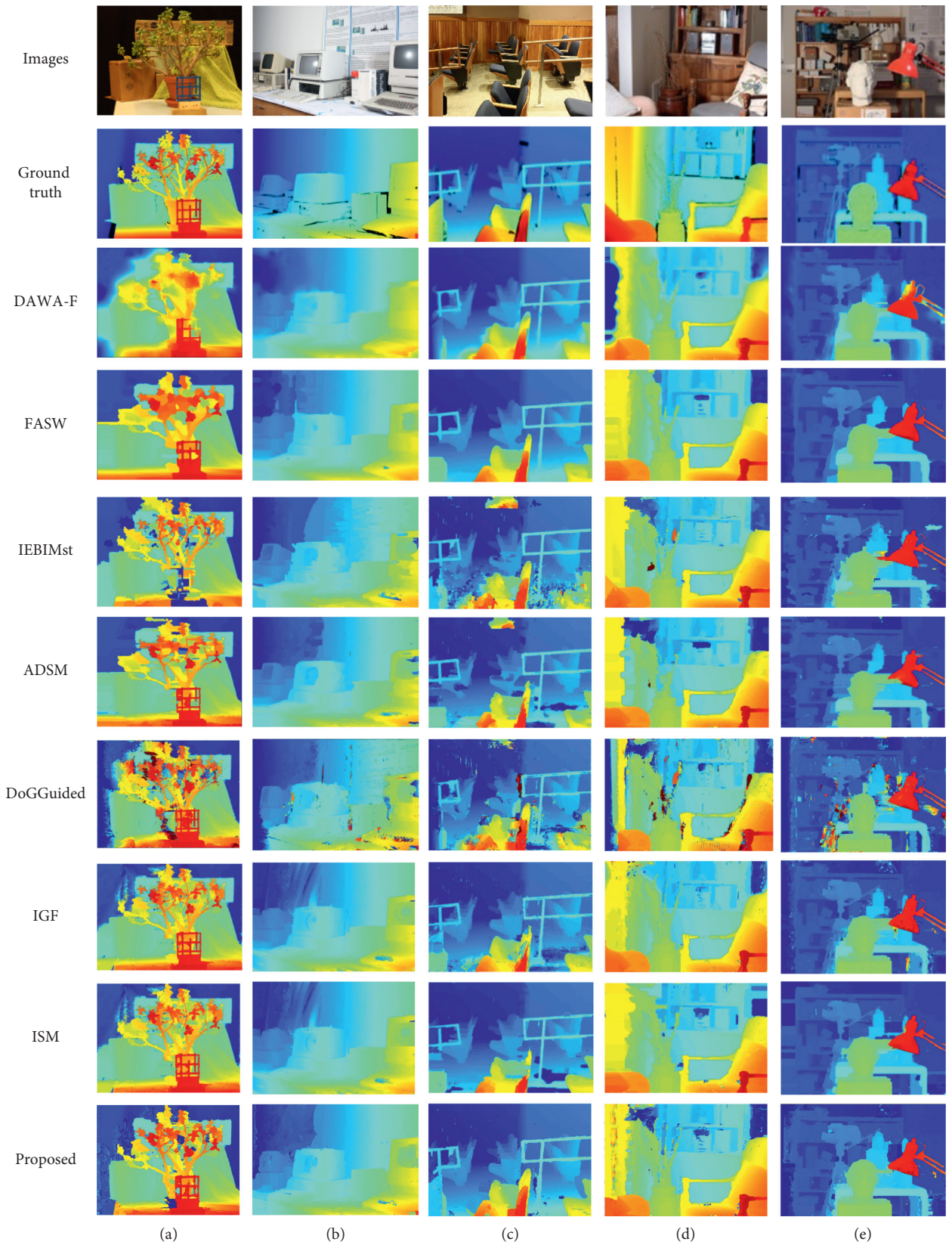


Figure 14: Comparison of disparity maps with other local stereo matching algorithms. (a) Jadeplant, (b) Vintage, (c) Classroom2, (d) Livingroom, and (e) Newkuba.

Table 1: Comparison of bad 2.0 in nonoccluded regions with other local stereo matching algorithms.

Image name	DAWA-F (%)	FASW (%)	IEBIMst (%)	ADSM (%)	IGF (%)	ISM (%)	DoGGuided (%)	Proposed (%)
Adirondack	20.9	26.0	41.3	36.5	36.7	37.3	40.5	**17.6**
ArtL	20.4	17.2	16.0	20.6	28.0	28.8	23.9	**14.9**
Jadeplant	44.7	34.3	31.5	35.7	44.6	45.5	42.5	**26.1**
Motorcycle	16.7	23.0	17.0	27.6	31.5	32.1	30.8	**15.5**
MotorcycleE	17.1	22.7	20.8	30.5	33.7	34.3	31.6	**14.3**
Piano	**21.6**	32.0	30.4	38.8	45.3	44.7	41.8	24.7
PianoL	49.5	43.6	53.9	59.0	52.5	53.1	58.0	**42.2**
Pipes	21.0	23.7	15.4	26.6	33.5	34.2	28.6	**14.5**
Playroom	30.5	37.5	42.0	46.8	44.4	44.6	44.6	**29.4**
Playtable	**27.0**	43.5	55.0	56.9	59.8	59.8	61.5	36.8
PlaytableP	**19.2**	28.1	25.6	31.8	44.4	44.5	36.4	**19.2**
Recycle	19.8	23.2	24.2	29.6	37.8	38.1	33.1	**17.9**
Shelves	51.3	49.9	52.5	53.3	52.5	51.8	54.6	**48.6**
Teddy	10.8	10.5	**8.57**	12.2	21.6	22.0	14.7	8.78
Vintage	39.1	44.1	64.7	52.8	54.9	55.0	57.5	**38.9**
Australia	47.2	41.7	**36.7**	40.4	42.7	42.5	45.4	37.5
AustraliaP	13.6	18.1	12.1	20.3	20.1	26.4	23.6	**10.8**
Bicycle2	**13.1**	23.1	16.9	27.3	23.7	34.8	30.6	16.3
Classroom2	19.2	27.2	32.5	35.1	32.2	36.1	34.6	**17.4**
Classroom2E	66.4	**40.6**	51.0	55.9	45.6	44.5	52.5	44.9
Computer	20.4	19.1	25.3	22.3	28.6	34.4	28.3	**17.2**
Crusade	**30.3**	34.9	58.1	56.1	43.0	56.2	59.1	33.5
CrusadeP	33.9	28.1	49.8	50.9	37.2	52.7	53.8	**25.2**
Djembe	**8.73**	18.5	11.2	24.2	21.4	25.2	26.4	11.4
DjembeL	48.9	**40.8**	48.6	58.0	50.9	51.0	60.6	45.4
Hoops	37.8	36.4	56.9	56.3	44.7	52.4	54.7	**35.7**
Livingroom	26.7	29.3	30.2	36.5	34.7	39.7	38.3	**26.6**
Newkuba	29.9	28.4	26.8	32.1	31.9	33.3	35.5	**23.3**
Plants	28.0	31.1	26.9	38.7	37.4	38.8	44.5	**23.6**
Staircase	**36.5**	41.0	71.7	69.7	47.1	75.3	72.0	38.4
Weighted average (%)	26.2	28.3	31.5	36.3	36.6	40.2	39.2	**23.1**

Table 2: Comparison of bad 2.0 in all regions with other local stereo matching algorithms.

Image name	DAWA-F (%)	FASW (%)	IEBIMst (%)	ADSM (%)	IGF (%)	ISM (%)	DoGGuided (%)	Proposed (%)
Adirondack	25.7	28.0	43.6	38.6	39.3	40.0	40.5	**22.7**
ArtL	34.9	27.9	27.0	30.2	36.2	37.1	**23.9**	29.0
Jadeplant	56.4	45.5	41.6	46.7	53.5	54.4	42.5	**40.4**
Motorcycle	24.1	27.2	**22.6**	31.2	35.6	36.3	30.8	23.5
MotorcycleE	24.6	26.9	26.6	34.3	37.9	38.5	31.6	**22.2**
Piano	**26.6**	36.1	34.6	42.1	48.6	48.1	41.8	30.0
PianoL	52.8	47.0	56.8	61.2	55.4	56.1	58.0	**46.0**
Pipes	33.4	33.7	**26.6**	36.5	42.5	43.1	28.6	27.2
Playroom	39.3	44.9	49.3	53.2	50.6	50.9	44.6	**38.5**
Playtable	**34.1**	47.0	58.3	59.8	62.2	62.3	61.5	42.6
PlaytableP	**25.9**	31.3	31.0	36.2	48.4	48.5	36.4	26.7
Recycle	24.6	25.8	26.0	31.7	40.2	40.5	33.1	**22.7**
Shelves	52.8	51.3	54.0	54.4	53.6	53.0	54.6	**50.7**
Teddy	19.4	17.3	15.6	18.6	26.9	27.3	**14.7**	16.5
Vintage	43.4	48.1	66.6	56.0	57.6	57.7	57.5	**43.1**
Australia	49.2	42.9	**39.2**	41.0	44.2	43.5	45.4	41.5
AustraliaP	19.1	20.4	**15.6**	22.2	23.3	28.2	23.6	16.6
Bicycle2	**19.3**	27.7	22.2	32.1	29.3	38.8	30.6	22.0
Classroom2	29.8	33.9	38.9	40.8	39.2	41.9	34.6	**27.7**
Classroom2E	70.6	**46.3**	57.3	61.1	51.6	49.6	52.5	52.0
Computer	34.2	31.3	35.3	34.1	40.3	43.6	**28.3**	31.4
Crusade	43.1	**39.0**	63.9	62.7	50.6	60.9	59.1	44.8
CrusadeP	46.2	**35.4**	57.3	58.1	45.3	58.4	53.8	38.0
Djembe	**13.6**	21.1	13.7	26.4	24.5	27.4	26.4	15.9
DjembeL	52.2	**42.5**	49.5	58.7	52.6	52.0	60.6	47.5
Hoops	49.6	47.8	64.4	64.1	54.8	60.8	54.7	**47.4**
Livingroom	34.9	35.7	35.6	41.3	40.0	45.3	38.3	**33.7**
Newkuba	38.3	33.9	33.2	37.0	37.8	38.9	35.5	**31.3**

Table 2: Continued.

Image name	DAWA-F (%)	FASW (%)	IEBIMst (%)	ADSM (%)	IGF (%)	ISM (%)	DoGGuided (%)	Proposed (%)
Plants	38.2	37.1	**32.4**	44.5	43.4	43.8	44.5	34.2
Staircase	**45.7**	48.8	75.2	74.0	54.0	78.2	72.0	47.0
Weighted average (%)	34.3	34.0	37.2	41.5	42.1	44.9	46.2	**31.4**

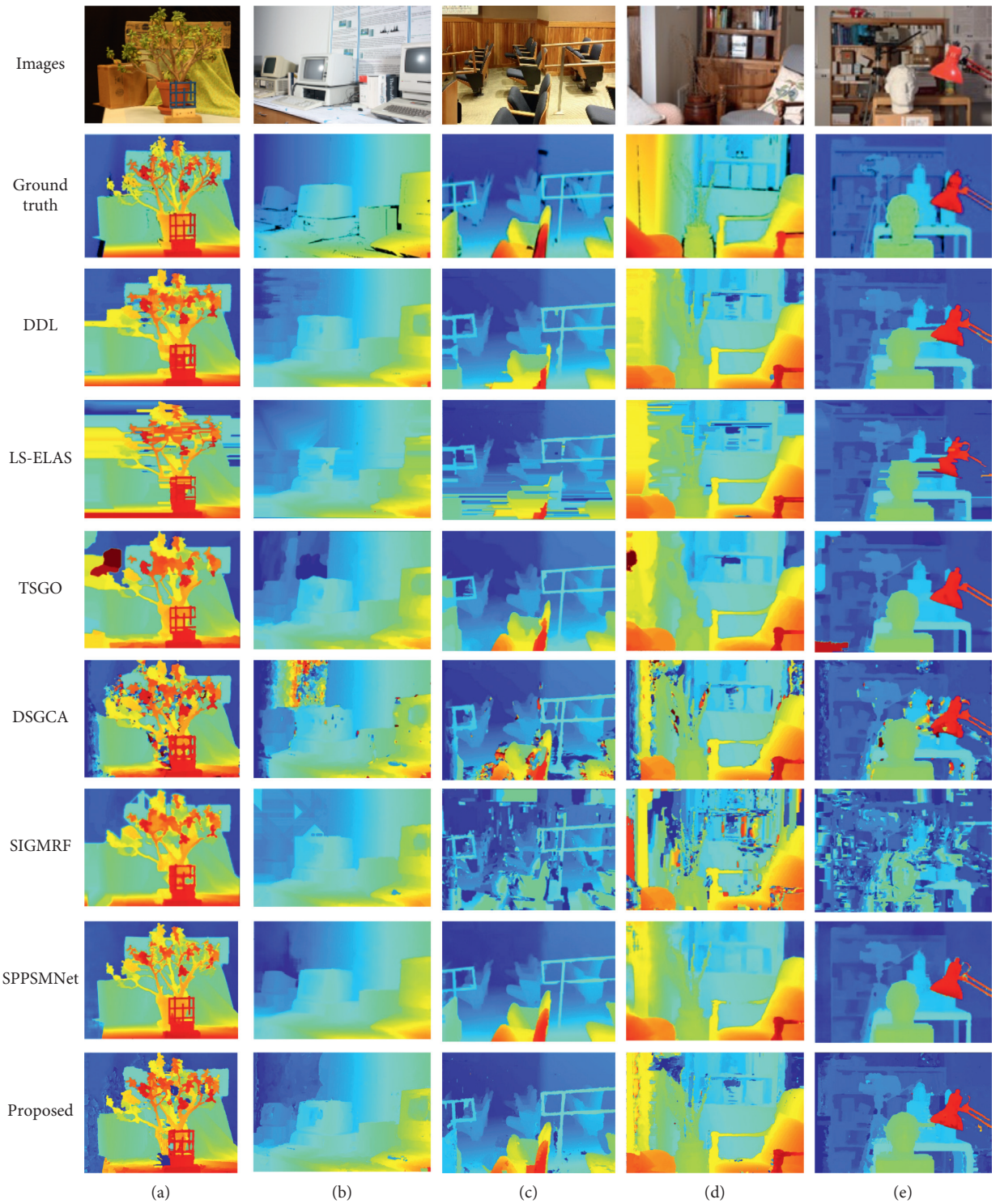


Figure 15: Comparison of disparity maps with other nonlocal stereo matching algorithms. (a) Jadeplant. (b) Vintage. (c) Classroom2. (d) Livingroom. (e) Newkuba.

Table 3: Comparison of bad 2.0 in nonoccluded regions with other nonlocal stereo matching algorithms.

Image name	DDL (%)	LS-ELAS (%)	TSGO (%)	DSGCA (%)	SPPSMNet (%)	SIGMRF (%)	Proposed (%)
Adirondack	24.2	34.5	27.3	27.0	25.6	29.2	**17.6**
ArtL	20.2	**10.6**	12.3	20.4	39.8	15.0	14.9
Jadeplant	34.5	40.6	53.1	35.0	45.5	41.5	**26.1**
Motorcycle	26.1	21.4	23.5	26.1	32.3	28.7	**15.5**
MotorcycleE	24.3	23.0	25.7	25.1	33.7	34.5	**14.3**
Piano	31.2	33.1	33.4	32.4	37.2	36.7	**24.7**
PianoL	43.6	48.8	54.5	45.7	54.1	47.2	**42.2**
Pipes	26.1	23.0	22.5	26.0	34.8	15.6	**14.5**
Playroom	36.2	38.5	49.6	41.8	43.1	39.9	**29.4**
Playtable	38.8	51.7	45.0	53.5	38.1	54.5	**36.8**
PlaytableP	25.6	21.1	27.0	35.1	32.8	35.5	**19.2**
Recycle	24.3	36.2	24.2	26.7	30.4	34.2	**17.9**
Shelves	54.7	54.6	52.2	54.9	**47.8**	58.2	48.6
Teddy	12.4	14.4	13.3	13.8	21.4	14.1	**8.78**
Vintage	44.1	50.6	57.5	50.8	46.7	41.1	**38.9**
Australia	44.3	53.5	34.1	42.9	**32.5**	60.0	37.5
AustraliaP	19.4	**10.3**	16.9	20.9	20.6	33.0	10.8
Bicycle2	25.8	**15.8**	20.0	23.6	34.0	67.9	16.3
Classroom2	28.3	37.0	43.3	30.2	33.0	63.2	**17.4**
Classroom2E	**42.1**	83.6	55.4	45.5	55.9	99.5	44.9
Computer	21.1	24.5	**14.3**	27.6	36.3	39.8	17.2
Crusade	37.1	49.1	54.1	42.0	53.8	84.8	**33.5**
CrusadeP	28.7	34.6	49.2	36.0	48.6	82.0	**25.2**
Djembe	21.7	13.9	33.9	21.0	35.0	35.2	**11.4**
DjembeL	46.8	**44.9**	66.2	50.2	71.6	95.2	45.4
Hoops	36.0	45.7	45.9	44.2	52.8	91.5	**35.7**
Livingroom	30.3	34.9	39.8	33.3	37.1	58.1	**26.6**
Newkuba	28.4	29.1	42.6	34.6	38.1	65.8	**23.3**
Plants	32.7	64.4	47.2	38.4	46.7	55.0	**23.6**
Staircase	**37.5**	62.7	52.6	46.8	56.9	88.6	38.4
Weighted average (%)	29.4	33.6	35.2	32.6	38.7	48.3	**23.1**

Table 4: Comparison of bad 2.0 in all regions with other nonlocal stereo matching algorithms.

Image name	DDL (%)	LS-ELAS (%)	TSGO (%)	DSGCA (%)	SPPSMNet (%)	SIGMRF (%)	Proposed (%)
Adirondack	26.4	37.1	29.4	32.5	28.6	32.5	**22.7**
ArtL	30.3	**19.7**	20.8	36.5	48.4	26.0	29.0
Jadeplant	46.0	52.2	61.0	49.4	54.8	53.2	**40.4**
Motorcycle	30.2	25.1	28.5	33.5	37.1	34.9	**23.5**
MotorcycleE	28.2	26.4	30.3	32.5	38.4	39.8	**22.2**
Piano	35.4	36.9	37.1	37.5	41.3	41.0	**30.0**
PianoL	47.1	51.5	57.1	49.7	56.8	50.5	**46.0**
Pipes	36.3	33.0	31.6	38.0	42.3	**27.1**	27.2
Playroom	43.7	45.4	55.0	49.6	48.7	46.3	**38.5**
Playtable	43.1	55.2	48.8	58.1	**41.9**	58.6	42.6
PlaytableP	29.4	**24.4**	31.9	41.7	35.8	41.3	26.7
Recycle	27.2	38.7	26.8	31.9	32.5	38.1	**22.7**
Shelves	55.9	55.2	53.6	57.6	**49.8**	60.1	50.7
Teddy	19.0	20.4	18.2	22.6	25.9	21.2	**16.5**
Vintage	48.0	53.7	60.5	54.6	50.0	44.9	**43.1**
Australia	45.5	54.6	35.6	47.3	**35.3**	63.6	41.5
AustraliaP	22.2	**13.5**	18.7	27.0	24.3	39.9	16.6
Bicycle2	30.5	**20.9**	24.3	30.2	37.0	70.7	22.0
Classroom2	34.8	43.7	49.0	40.1	39.8	68.4	**27.7**
Classroom2E	**47.3**	85.3	60.2	53.2	59.2	99.4	52.0
Computer	33.5	35.5	**25.3**	40.3	46.5	50.4	31.4
Crusade	**42.2**	55.9	60.2	52.6	60.2	87.5	44.8
CrusadeP	**35.5**	44.4	55.4	47.8	55.2	85.2	38.0
Djembe	24.3	17.2	36.2	26.3	38.6	39.6	**15.9**

Table 4: Continued.

Image name	DDL (%)	LS-ELAS (%)	TSGO (%)	DSGCA (%)	SPPSMNet (%)	SIGMRF (%)	Proposed (%)
DjembeL	48.3	**46.7**	67.1	53.4	73.0	95.4	47.5
Hoops	**47.1**	55.2	55.1	55.3	60.3	93.1	47.4
Livingroom	35.9	40.6	43.3	40.7	41.6	62.9	**33.7**
Newkuba	34.3	35.5	46.3	43.0	43.6	70.2	**31.3**
Plants	39.3	67.7	50.7	47.7	53.0	61.8	**34.2**
Staircase	**45.6**	66.5	58.3	55.1	62.5	90.2	47.0
Weighted average (%)	35.1	38.9	40.0	40.8	43.8	53.7	**31.4**

SIGMRF, and SPPSMNet are algorithms based on deep learning. The disparity map comparison of the same five stereo images is shown in Figure 15.

The comparison results of bad 2.0 are shown in Tables 3 and 4, where the bold fonts also indicate the best results.

Similar to Tables 1 and 2, the results of Tables 3 and 4 can also indicate that the robustness and performance of the proposed algorithm are obviously better than those of the other six state-of-the-art nonlocal algorithms.

5. Conclusions

In this study, an improved cost aggregation method is proposed, in which the matching cost volume is filtered by ACR-GIF-OW. Different from other methods adopted ACR-GIF, the proposed method takes the orthogonal weight of each point in the ACR into consideration. For improving the computational efficiency of the proposed method, a weighted aggregation computing method based on orthogonal weights is proposed. Moreover, a local stereo matching algorithm using ACR-GIF-OW is proposed as well. Experimental results demonstrate that, compared with that of ACR-GIF, the disparity accuracy of ACR-GIF-OW is significantly improved at the cost of a smaller growth of the time overhead, and the stereo matching algorithm proposed in this paper exhibits superior performance than those of other state-of-the-art local and nonlocal algorithms. In the future work, we will introduce the orthogonal weight in the disparity refinement to further improve the disparity accuracy.

Data Availability

The dataset used to support the findings of this study are included in the article, which are cited at relevant places within the text as [32].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61901287), the Key Research and Development Project of Sichuan Province (Grant no. 2020YFG0112; Grant no. 2020YFG0306), the Major Science and Technology Project of Sichuan Province (Grant nos. 2018GZDZX0024, 2019ZDZX0039, and 2018GZDZX0029), and the Science and Technology Planning Project of Sichuan Province (Grant no. 2020YFG0288). The authors would like to thank Associate Professor Yue Wu for reviewing the manuscript.

References

[1] R. A. Hamzah, H. Ibrahim, and A. H. A. Hassan, "Stereo matching algorithm for 3D surface reconstruction based on triangulation principle," in *Proceedings of the 2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 119–124, Yogyakarta, Indonesia, July 2016.

[2] X. Feng, J. Liu, Y. Deng, and S. Xu, "The measurement of three-dimensional points based on the single camera stereo vision sensor," in *Proceedings of the 2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pp. 278–281, Changsha, China, August 2017.

[3] G. Zhuo, Y. Zhu, and G. Liang, "A 3D terrain reconstruction method of stereo vision based quadruped robot navigation system," in *Proceedings of the Seventh International Conference on Electronics and Information Engineering Proceedings of the SPIE*, Dalian, China, September 2017.

[4] X. Xiongwu, "Multi-view stereo matching based on self-adaptive patch and image grouping for multiple unmanned aerial vehicle imagery," *Remote Sensing*, vol. 8, no. 2, p. 89, 2016.

[5] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, no. 65, pp. 1–32, 2016.

[6] J. Pang, W. Sun, J. S. Ren et al., "Cascade residual learning: a two-stage convolutional neural network for stereo matching," in *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 878–886, Venice, Italy, June 2017.

[7] J. Chang and Y. Chen, "Pyramid stereo matching network," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5410–5418, Salt Lake City, UT, USA, December 2018.

[8] K. Swami, K. Raghavan, N. Pelluri et al., "DISCO: depth inference from stereo using context," in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 502–507, Shanghai, China, May 2019.

[9] S. Kim, D. Min, S. Kim et al., "Unified confidence estimation networks for robust stereo matching," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1299–1313, 2019.

[10] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proceedings of the Eighth IEEE International Conference on Computer*

Vision. ICCV 2001, pp. 508–515, Vancouver, Canada, April 2001.

[11] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.

[12] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[13] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pp. 131–140, Kauai, HI, USA, August 2001.

[14] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[15] A. Hosni, C. Rhemann, M. Bleyer et al., "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013.

[16] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, "Weighted guided image filtering," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 120–129, 2015.

[17] G. Hong and B. Kim, "A local stereo matching algorithm based on weighted guided image filtering for improving the generation of depth range images," *Displays*, vol. 49, pp. 80–87, 2017.

[18] G. A. Kordelas, D. S. Alexiadis, P. Daras et al., "Content-based guided image filtering, weighted semi-global optimization, and efficient disparity refinement for fast and accurate disparity estimation," *IEEE Transactions on Multimedia*, vol. 18, no. 2, pp. 155–170, 2016.

[19] R. A. Hamzah, A. F. Kadmin, M. S. Hamid et al., "Improvement of stereo matching algorithm for 3D surface reconstruction," *Signal Processing: Image Communication*, vol. 65, pp. 165–172, 2018.

[20] W. Wu, H. Zhu, S. Yu et al., "Stereo matching with fusing adaptive support weights," *IEEE Access*, vol. 7, pp. 61960–61974, 2019.

[21] C. T. Zhu and Y. Z. Chang, "Efficient stereo matching based on pervasive guided image filtering," *Mathematical Problems in Engineering*, vol. 2019, Article ID 3128172, 11 pages, 2019.

[22] Q. Q. Yang, "Fast stereo matching using adaptive guided filtering," *Image & Vision Computing*, vol. 32, pp. 202–211, 2014.

[23] S. P. Zhu and L. N. Yan, "Local stereo matching algorithm with efficient matching cost and adaptive guided image filter," *Visual Computer*, vol. 33, no. 9, pp. 1087–1102, 2017.

[24] Y. Xu, Y. Zhao, and M. Ji, "Local stereo matching with adaptive shape support window based cost aggregation," *Applied Optics*, vol. 53, no. 29, pp. 6885–6892, 2014.

[25] S. Zhu, Z. Wang, X. Zhang et al., "Edge-preserving guided filtering based cost aggregation for stereo matching," *Journal of Visual Communication and Image*, vol. 39, pp. 107–119, 2016.

[26] L. Yan, R. Wang, H. Liu et al., "Stereo mactching method based on improved cost computation and adaptive guided filter," *Acta Optica Sinica*, vol. 38, no. 11, Article ID 1115007, 2018.

[27] 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.

[28] L. Y. Kong, J. P. Zhu, and S. C. Ying, "Stereo matching based on guidance image and adaptive support region," *Acta Optica Sinica*, vol. 40, no. 9, Article ID 0915001, 2020.

[29] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proceedings of the European Conference on Computer Vision*, pp. 151–158, Stockholm, Sweden, June 1994.

[30] X. Mei, X. Sun, M. Zhou et al., "On building an accurate stereo matching system on graphics hardware," in *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 467–474, Barcelona, Spain, May 2011.

[31] Q. Yang, D. Li, L. Wang et al., "Full-image guided filtering for fast stereo matching," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 237–240, 2013.

[32] D. Scharstein, R. Szeliski, and H. Hirschmuller, "Middlebury stereo evaluation version 3," 2014, https://vision.middlebury.edu/stereo/eval3/.

[33] J. Navarro and A. Buades, "Semi-dense and robust image registration by shift adapted weighted aggregation and variational completion," *Image and Vision Computing*, vol. 89, pp. 258–275, 2019.

[34] C. He, C. Zhang, Z. Chen et al., "Minimum spanning tree based stereo matching using image edge and brightness information," in *Proceedings of the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, Shanghai, China, September 2017.

[35] M. Ning, "Accurate dense stereo matching based on image segmentation using an adaptive multi-cost approach," *Symmetry*, vol. 8, p. 159, 2016.

[36] M. Kitagawa, I. Shimizu, and R. Sara, "High accuracy local stereo matching using DoG scale map," in *Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pp. 258–261, Nagoya, Japan, May 2017.

[37] R. A. Hamzah, H. Ibrahim, and A. H. A. Hassan, "Stereo matching algorithm based on per pixel difference adjustment, iterative guided filter and graph segmentation," *Journal of Visual Communication and Image Representation*, vol. 42, pp. 145–160, 2017.

[38] R. A. Hamzah, A. F. Kadmin, M. S. Hamid, S. F. A. Ghani, and H. Ibrahim, "Improvement of stereo matching algorithm for 3D surface reconstruction," *Signal Processing: Image Communication*, vol. 65, pp. 165–172, 2018.

[39] J. Yin, "Sparse representation over discriminative dictionary for stereo matching," *Pattern Recognition*, vol. 17, 2017.

[40] R. A. Jellal, M. Lange, B. Wassermann et al., "Line segment based efficient large scale stereo matching," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 146–152, Singapore, August 2017.

[41] M. G. Mozerov and J. Van de Weijer, "Accurate stereo matching by two-step energy minimization," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1153–1163, 2015.

[42] I. K. Park, "Deep self-guided cost aggregation for stereo matching," *Pattern Recognition Letters 112*, vol. 41, pp. 168–175, 2018.

[43] S. Nahar and M. V. Joshi, "A learned sparseness and IGMRF-based regularization framework for dense disparity estimation using unsupervised feature learning," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, p. 1, 2017.

[44] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13961–13970, Seattle, WA, USA, November 2020.