Hindawi
Journal of Sensors
Volume 2017, Article ID 9374870, 12 pages
https://doi.org/10.1155/2017/9374870

Research Article

A Nonlocal Method with Modified Initial Cost and Multiple Weight for Stereo Matching

Shenyong Gao,[1,2] Haohao Ge,[3] Hua Zhang,[3] and Ying Zhang[2]

[1] *College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*
[2] *School of Information Engineering, Zhejiang University of Water Resources and Electric Power, Hangzhou 310018, China*
[3] *School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China*

Correspondence should be addressed to Shenyong Gao; gaosy@hdu.edu.cn

Received 7 April 2017; Revised 5 June 2017; Accepted 20 June 2017; Published 6 August 2017

Academic Editor: Wendy Flores-Fuentes

This paper presents a new nonlocal cost aggregation method for stereo matching. The minimum spanning tree (MST) employs color difference as the sole component to build the weight function, which often leads to failure in achieving satisfactory results in some boundary regions with similar color distributions. In this paper, a modified initial cost is used. The erroneous pixels are often caused by two pixels from object and background, which have similar color distribution. And then inner color correlation is employed as a new component of the weight function, which is determined to effectively eliminate them. Besides, the segmentation method of the tree structure is also improved. Thus, a more robust and reasonable tree structure is developed. The proposed method was tested on Middlebury datasets. As can be expected, experimental results show that the proposed method outperforms the classical nonlocal methods.

1. Introduction

Dense two-frame stereo matching is one of the most extensively researched topics in machine vision. Finding corresponding points in two or more images is the most important progress. After their disparities are computed, the results are used to distinguish the objects and background. Moreover, the depth information arises from the obtained disparity map. Scharstein and Szeliski [1] performed the following four steps:

(1) Cost computation

(2) Cost aggregation

(3) Disparity computation

(4) Disparity refinement

Additionally, they separated stereo matching algorithms into local methods and global methods. On the one hand, in local methods, they require cost aggregation, which ensures that the disparity between pixels is more accurate and specific than making the calculation with only one pixel. Therefore, in local methods, the support windows of cost aggregation for each pixel are significant. On the other hand, global methods construct a global energy function, and then the matching problem can be replaced by optimization. In these methods, a global energy function always consists of data and a smoothness item. The former measures the matching degree of the guidance image and the disparity function. However, the latter is capable of embodying the constraint of the definition model. An important problem for these methods, however, is to find the balance. It is different to obtain the perfect matching result between both measures. A number of global methods have been developed such as dynamic programming [2], graph cut [3], and belief propagation [4].

The semiglobal matching (SGM) algorithm by Hirschmüller [5] plays a good trade between matching accuracy and speed. SGM performs energy minimization along several 1D paths across the image and, thus, approximates the otherwise two-dimensional NP-complete energy minimization problem. However, high computational complexity and memory demand are a challenge for fast implementations. SGM can be implemented relatively efficiently by parallelization schemes. Real-time designs are possible and have been reported for

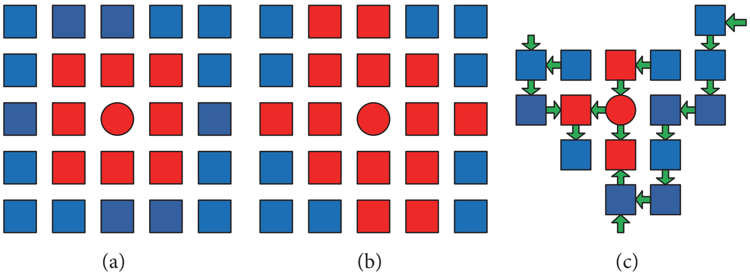


Figure 1: The support regions of cost aggregation. The red circle denotes the central pixel, and the red squares denote the pixels in the support region. The blue pixels are irrelevant. (a) Fixed support window; (b) cross-based support window; and (c) tree structure.

CPU and GPU systems [6]. There also exist some real-time embedded system designs, for example, on FPGA [7]. Schumacher and Greiner designed a higher data throughput FPGA architecture for SGM [8].

As for local methods, the problem in finding the correspondence of pixel p and pixel p' can be concluded as a similarity comparison of the two local patches, which exist around p and p', respectively [9]. Hence, the problem of finding the correspondence of two pixels is how to compute the cost value about two patches surrounded. Since then, it requires gathering the cost of each pixel during the cost aggregation procedure. Yoon and Kweon [10] proposed an adaptive support weight (ASW) method, which has higher matching accuracy but low efficiency. They use large support windows for robust cost aggregation which causes a huge computational burden [11] and fails to obtain satisfactory results on large planar surfaces.

For this reason, to obtain accurate results, the matching windows with an appropriate size and shape should be selected. However, the fixed windows method (shown in Figure 1(a)) is restrictive. It may result in incorrect matching in low-texture areas if the support windows are not large enough, and the windows break the boundaries between the object and background to influence the validity of the depth discontinuity regions [12].

To this end, many methods to construct matching windows have been proposed recently. For instance, Qu et al. [13] presented an algorithm that filters the inapposite pixels around the matching point by using the color similarity of the pixels around a central matching point. This algorithm finally acquires the appropriate pixels that construct the adaptive support windows, which are helpful to the matching point. Zhang et al. [14] also proposed a cross-based structure (Figure 1(b)) and constructed it in the form of adaptive support windows by comparing the color similarity around the adjacent pixels. Both methods calculate the disparity of pixels with the assistance of adaptive support windows, which make the operations more specific and suitable than the approaches using a predefined fixed-size window. These computations, however, are dependent on the construction of each support window. And the time consumption caused by cost aggregation still does not satisfy the real-time requirement. Therefore, Mei et al. [11] designed an accurate stereo matching system by using an accelerated CUDA implementation on the basis of the previous proposed methods, which significantly improved the efficiency of the algorithm under the help of hardware.

Recently, Yang [15] proposed a nonlocal cost aggregation (NLCA) method and then relied on it to perform tree-based filtering [16]. The NLCA algorithm is a novel cost aggregation method on a tree structure instead of using support windows. It also has been demonstrated to outperform the tradition of cost aggregation methods on support windows in terms of both speed and accuracy. In the NLCA algorithm, the nodes of the tree are all the image pixels, and the edges are all the edges between the nearest neighboring pixels. The similarity between any two pixels is decided by their shortest distance on the tree. All the pixels are connected to make a tree as shown in Figure 1(c), each node is aggregated only with its parents and children directly, and then every node on the tree makes a contribution to the final results. Hence, both the accuracy and the efficiency have been improved in this method. Nevertheless, this method does not perform well when the scene is composed of boundaries between object and background areas with similar color distribution because it considers color correlation as the only component of the weight function.

Mei et al. [17] proposed segment-tree cost aggregation (STCA) that segments the guidance image into several independent trees and then independent segment graphs are linked to form the segment-tree structure. In addition, they selected initial depth as a new component when computing the weight function. This method involves a new process; it leads to consistent scene segmentation; and only one judgement condition is adopted during the three-step image segmentation process. More recently, a cross-scale framework which unified aggregated based algorithms was also proposed [18]. With the proposed color-depth weight, Peng et al. [19] further iteratively rebuilt the tree to improve the matching efficiency in textureless regions. Besides, based on a minimum spanning tree, Pham et al. [20] proposed a robust nonlocal stereo matching algorithm that improves the performance of nonlocal approaches for outdoor driving images.

In this paper, we propose an improved nonlocal cost aggregation algorithm that modifies the original algorithm in both computational cost and aggregation. The additional vertical gradient will be used as one of the components to calculate the initial cost of each pixel. We also employ a known function named *Gemen − McClure* [21] to deal with outliers. Furthermore, we add the inner correlations and mix them with color correlation. And then we compute the weight function with a mixture of both correlations together. Moreover, when segmenting the guidance image more reasonably is under consideration, we also try to provide a new segmentation method with brand.

We evaluate our proposed method on standard and extra Middlebury datasets and compare our method with ST and MST. Experimental results show that our method can achieve acceptable results when it is in the process of computing the accuracy of disparity, especially in some representative regions. The average number of erroneous pixels around discontinuous regions can be reduced efficiently while the disparities of flat regions become more stable. Compared with NLCA and STCA, a performance evaluation on Middlebury datasets shows that the proposed method has higher correct matching rate. In our method, the percentage of matching error declined to between 5% and 15%. Additionally, the computational cost of the new segmentation method can be ignored usually, while only the cost from the inner color correlation which was employed in our cost aggregation procedure also has a weak impact on the computational complexity. In this method, the computational complexity is the same as color correlation in terms of magnitude. Therefore, the total computational complexity retains the same magnitude as the STCA algorithm but slightly improves the result.

The main contribution of this paper is to improve the original nonlocal cost aggregation method with the following advantages:

(1) It has higher accuracy by adding the vertical gradient as one of the components in the process of cost computation. It is proved to be better in some discontinuous areas. Its initial value is more stable with the *Gemen − McClure* function.

(2) Inner color correlation is employed in the computation of the weight function to make constructing a tree structure more robust and reasonable.

(3) The segmentation method of STCA is improved and it achieves a better result. Moreover, irrelevant pixels contribute less to each other.

The rest of this paper is organized as follows. In Section 2, we briefly introduce related work on local methods. Then, our proposed improved method is described in Section 3. Section 4 describes and analyzes the experimental results, and Section 5 discusses setting the parameters. Finally, we provide conclusion in Section 6.

2. Related Work

Cost aggregation, which consists of constructing support regions and aggregating the disparity for each pixel within those support regions, is one of the important processes in stereo matching. The efficiency and effectiveness rely on the used aggregation method; therefore, they are different from each other. In this section, we review the related work on cost aggregation, especially on the traditional local methods and nonlocal cost aggregation methods based on tree structure.

2.1. The Traditional Local Methods. The stationary support windows with a stationary weight for each pixel are used by the simplest local method of cost aggregation. However, note that this method fails in many specific regions, including occlusion regions and low-textured areas. Furthermore, this method is unable to achieve decent robustness and its matching accuracy falls well short of the ideal result. To resolve this dilemma, there are usually two approaches: (1) make the fixed support window alterable using shiftable windows, multiple windows [22], or variable windows [23, 24] or (2) concentrate on varying the weights to achieve excellent matching accuracy.

The algorithms based on adaptive weight consider every pixel in the support windows as a unique unit and calculate weight for the central point by themselves. The pixel will have a dramatic effect on the final result only if there is a cost value which is similar to the central point. Hence, every pixel is able to receive proper contributions from all the other neighboring pixels. This approach blurs the boundaries between local methods and global methods due to its remarkable accuracy and the obvious increase of computational cost.

Yoon and Kweon [10] first proposed an adaptive weight method and Gu et al. [25] further enhanced their method by introducing rank transform and disparity refinement. Tombari et al. [26] obtained the cost value after using the Meanshift [27] algorithm to segment the image, which revises ASW algorithm performance calamitously in repetitive texture regions and discontinuous regions. Hosni et al. [28] performed connectivity by using the geodesic distance transform; nevertheless, the computational efficiency of their strategy still has similar efficiency to others.

2.2. Nonlocal Cost Aggregation Based on Tree Structure. Even though great progress has been made in local algorithms, they still aggregate pixels into local regions. As mentioned above, a nonlocal cost aggregation (NLCA) method has been proposed that breaks through the boundaries of local and global methods. This method transforms the guidance images into a graph and constructs a tree structure so that all the image pixels become the nodes of the tree. Before aggregating, a minimum spanning tree (MST) must be constructed. The nodes attached to edges with the lowest weights (calculated by differences in color distribution process) are connected to one another until all the pixels are finally included in the tree. It is an important step, that is, to convert the guidance image into a cost tree after all the pixels have been connected. Then, the whole process is separated into three steps:

(1) Traversing the cost tree

(2) Assigning an appropriate value to each node

(3) Calculating each node's disparity level with its relatives

After constructing the tree structure, the aggregation costs can be efficiently computed by executing a tree filter, which traces the MST from the leaf nodes to the root nodes and from the root nodes to the leaf nodes. Hence, the aggregation is complete after only two trees traverse, and then any pixel receives proper contributions from every node in the constructed tree (more or less). Based on the tree structure, some effective disparity refinement methods are proposed as follows.

Chen et al. [29] improved the NLCA by adding depth information in the weight function, which enhances the effect of regions around the border. Mei et al. [17] proposed a new segment-tree (ST) method that divides the construction of the tree structure into two rounds. In the first round, it combines subtrees in the homogeneous regions, and it also keeps those subtrees that belong to different regions separate from each other if they break the predefined equation. In the second round, to ensure that the different regions have little impact on each other, it combines the remaining subtrees with a penalty value. However, the segmentation performance is not robust because the segmentation equation is extremely ordinary. Therefore, the performance of this method falls short of expectations.

3. Our Proposed Method

Our work is directly motivated by the above two nonlocal cost aggregation methods. We further improve these methods during cost computation and tree construction process, respectively. We include the vertical gradient as a new component in the cost computation. On the other hand, due to its stability and versatility, inner color correlation is employed instead of using a single color component. Moreover, we modify the structure of the segment tree, which improves its validity and robustness. In this section, we divide our methods into five parts as follows:

(1) Cost computation

(2) Tree construction

(3) Cost aggregation

(4) Disparity computation and refinement

(5) Computation complexity

More details can be found in the following subsections.

3.1. Cost Computation. Traditional nonlocal methods are considered to employ the truncated absolute difference of the color and the horizontal gradient as the initial cost. However, the performance of this cost measurement is unstable in marginal areas. Hence, we decided to employ the vertical gradient to make the cost measurement reveal more detailed description of the reference images. We compute the individual cost values $C_{AD}(p,d)$, $C_{GD_x}(p,d)$, and $C_{GD_y}(p,d)$ primarily for a pixel $p = (x,y)$ in the guidance image with a disparity level d. Let I_i denote RGB color component. $C_{AD}(p,d)$ is defined as the average absolute difference of p and its relevant pixel pd in the *RGB* channel (as shown in (1)):

$$C_{AD}(p,d) = \frac{1}{3}\sum_{i=R,G,B}\left|I_i^{\text{Left}}(p) - I_i^{\text{Right}}(pd)\right|. \tag{1}$$

Then, we compute the gradient cost values $C_{GD_x}(p,d)$ and $C_{GD_y}(p,d)$ using (2) and (3), respectively. The equations can be designed as follows:

$$C_{GD_x}(p,d) = \left|\bigtriangledown_x I^{\text{Left}}(p) - \bigtriangledown_x I^{\text{Right}}(pd)\right|, \tag{2}$$

$$C_{GD_y}(p,d) = \left|\bigtriangledown_y I^{\text{Left}}(p) - \bigtriangledown_y I^{\text{Right}}(pd)\right|. \tag{3}$$

In addition, our proposed method works pretty well when truncated values are used for discarding the extremum of the initial cost. However, the improvement this method yields is not obvious. Therefore, we employ the *Gemen − McClure* function to handle the exception values as shown in

$$\begin{aligned} C_{\text{tran}_c} &= \frac{C_{\text{init}_c}^2}{C_{\text{init}_c}^2 + \varepsilon_c^2}, \\ C_{\text{tran}_g} &= \frac{C_{\text{init}_g}^2}{C_{\text{init}_g}^2 + \varepsilon_g^2}, \end{aligned} \tag{4}$$

where C_{tran_c} and C_{init_c} denote the final and initial cost values of the color, respectively. And then let C_{tran_g} and C_{init_g} denote the final and initial cost values of the gradient, respectively. In addition, ε_c and ε_g are user-specified parameters for adjustment. The former is related to the color adjustment and the latter is related to adjustments on behalf of the gradient. ε_c is set to 7, and ε_g is set to 2 in our experiments. The effect of this function declines smoothly when the initial cost reaches a certain value and the final cost value converges to 1 under the control of ε. So, by using three cost components as mentioned above together, the final initial cost value can be expressed as the following equation:

$$\begin{aligned} C(p,d) = {} & \alpha \cdot C_{AD}(p,d) + \beta \cdot C_{GD_y}(p,d) \\ & + (1 - \alpha - \beta) \cdot C_{GD_x}(p,d), \end{aligned} \tag{5}$$

where α and β are the weights for each component. Figure 2 shows a comparison between the traditional cost computation and our method, which demonstrates the improvement after adding the discontinuous regions.

3.2. Tree Construction. According to Yang's contribution [15], we treat the guidance image I as a graph $G = (V,E)$ in this paper, where each node denotes the corresponding pixel in I and each edge represents the weight that connects two neighboring nodes. Accordingly, a flow chart shows how to construct our tree structure in Figure 3.

The weight W_e of an edge e is determined with its conjoint nodes p and q; this process can be described as follows:

$$\begin{aligned} W_e = {} & \theta_{\text{In}} \cdot \left|I_{\text{In}}(p) - I_{\text{In}}(q)\right| + (1 - \theta_{\text{In}}) \\ & \cdot \left|I(p) - I(q)\right|, \end{aligned} \tag{6}$$

where θ_{In} is the predefined weight and is set to 0.2 in this paper. I_{In} denotes the inner color correlation, which is shown in

$$I_{\text{In}} = \left[I_{RtG}, I_{GtB}, I_{BtR}\right]. \tag{7}$$

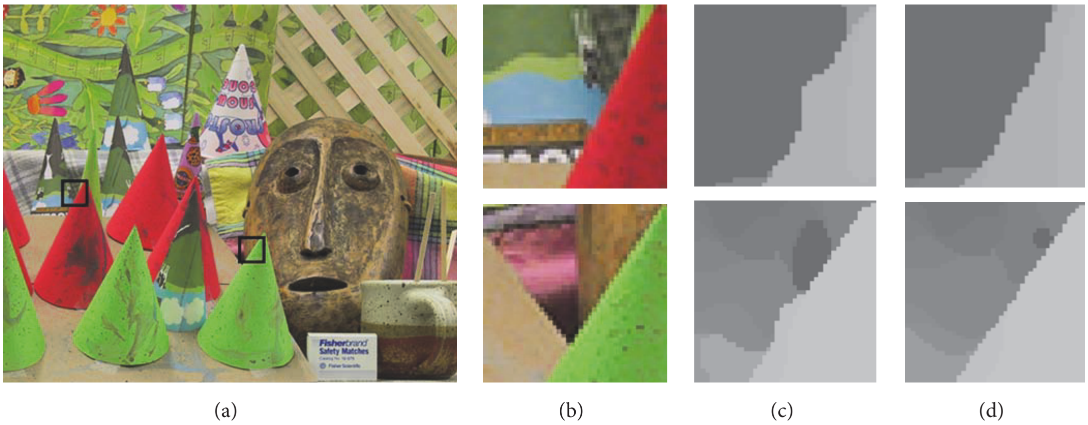


Figure 2: Cost measure comparison. (a) The input image; the black boxes express the target areas. (b) Insets of target area; (c) and (d) denote the results of the traditional cost measure and our method, respectively.

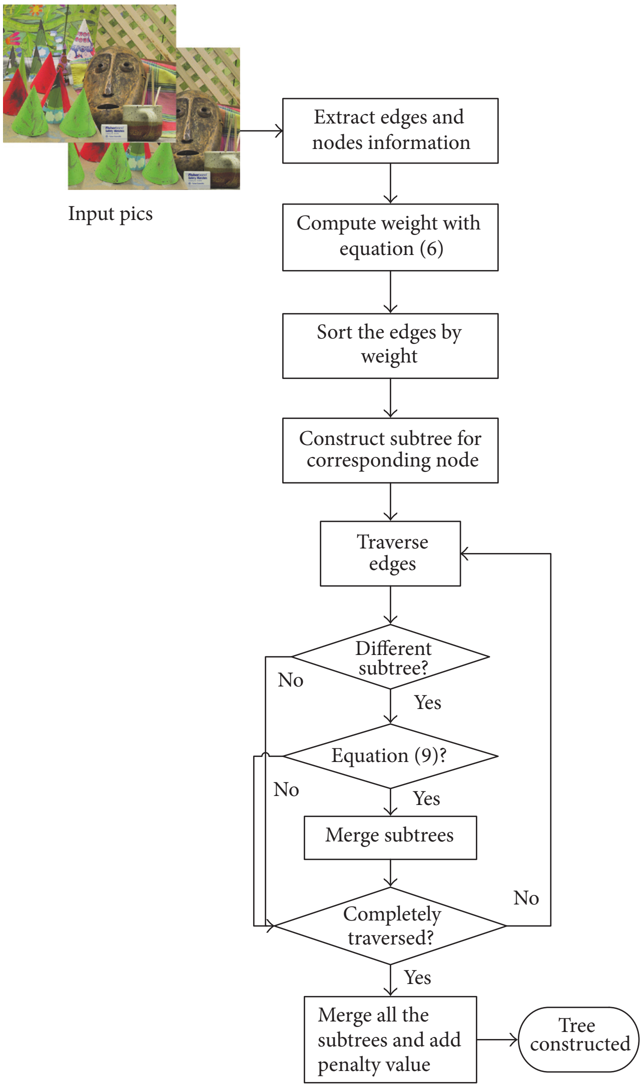

Input pics
Extract edges and nodes information
Compute weight with equation (6)
Sort the edges by weight
Construct subtree for corresponding node
Traverse edges
Different subtree?
No
Yes
Equation (9)?
No
Yes
Merge subtrees
No
Completely traversed?
Yes
Merge all the subtrees and add penalty value
Tree constructed

Figure 3: A flow chart showing the tree construction steps.

The components with a pixel $p(x, y)$ of I_{In} are specifically expressed as follows:

$$\begin{aligned} I_{RtG} &= I_R(x, y) - I_G(x, y), \\ I_{GtB} &= I_G(x, y) - I_B(x, y), \\ I_{BtR} &= I_B(x, y) - I_R(x, y). \end{aligned} \tag{8}$$

Then, the edges in E are sorted in an ascending order according to their weights. And then the subtrees are created for each node in V. Every node p has one subtree T_p. Finally, we traverse the sequence of edges, and then the subtrees T_p and T_q are merged into bigger groups only if the edge weight should satisfy

$$\begin{aligned} & w_{e_i} \le \min\left(\left(\max\left(w_{e_{T_p}}\right) + \frac{\tau}{|T_p|}\right), \left(\max\left(w_{e_{T_q}}\right) + \frac{\tau}{|T_q|}\right)\right), \quad w_{e_i} < w_{e_{\text{Avg}}}, \\ & w_{e_i} \le \min\left(\left(w_{e_{\text{Avg}}} + \frac{\tau}{|T_p|}\right), \left(w_{e_{\text{Avg}}} + \frac{\tau}{|T_q|}\right)\right), \quad w_{e_i} \ge w_{e_{\text{Avg}}}, \end{aligned} \tag{9}$$

where w_{e_i} denotes the weight of edge e_i that connects the two nodes p and q. $w_{e_{T_p}}$ and $w_{e_{T_q}}$ denote the weight sequence of edges in subtrees T_p and T_q, respectively. $w_{e_{\text{Avg}}}$ denotes the average weight of all the edges. τ is a predefined parameter. We employ $w_{e_{\text{Avg}}}$ and divide the equation into two cases, which guarantees that the constraint condition will not be lost in those boundary regions with high weights and makes the segmentation of the tree more precise and robust.

After traversing all the edges, a large number of subtrees are merged with each other and changed into some new

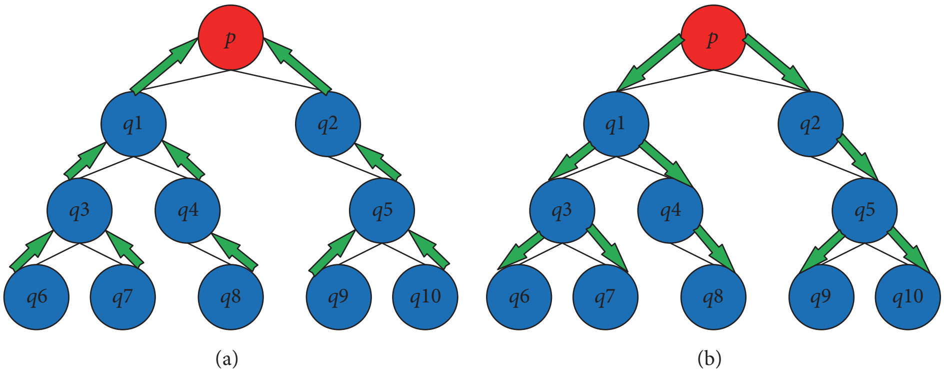


FIGURE 4: The tree filter for cost aggregation; p denotes the matching pixel. (a) Leaves to root pass; (b) root to leaves pass.

subtrees that have a bigger structure but are small in quantity. Note that the integrated graph I has been segmented into several smaller pieces. We then traverse the edges once again and merge the rest of the subtrees. Meanwhile, we add a penalty value to the weight of edges to ensure that boundary regions do not interact with each other. Finally, all the nodes are constructed into a segment tree T, and there is only one path between any two nodes in T. The segment tree T is used in aggregating the final cost value.

3.3. Cost Aggregation. The nonlocal cost aggregation method is a linear-time method in which the computational complexity is extremely low. We employ a weighting function $S(p,q)$ to compute the contribution from pixel q to p; its function is decided as follows:

$$S(p,q) = \exp\left(-\frac{D(p,q)}{\sigma}\right), \quad (10)$$

where $D(p,q)$ denotes the distance from p to q in the tree structure that relates to (6) and σ is a predefined parameter for adjustment. Because of the otherness of our initial matching cost, σ is set to 0.08 in our experiments, and the setting of σ will be discussed in Section 5. Let $C_d(p)$ denote the cost value for pixel p at disparity level d; the aggregated cost value $C_d^A(p)$ is computed as follows:

$$C_d^A(p) = \sum_{q\in I} S(p,q)\cdot C_d(q), \quad (11)$$

where I denotes the whole graph and therefore $C_d^A(p)$ is aggregated with all the nodes in the graph I. Yang employs a tree filter to compute the cost aggregation that traverses the tree structure from leaves to root and root to leaves [15], as shown in Figure 4. A node is affected by all the other nodes in the segment tree T but aggregates with only its children and parents. For a pixel p, the aggregated value is calculated as follows:

$$C_d^{A\uparrow}(p) = \sum_{q\in \mathrm{Child}(p)} S(p,q)\cdot C_d^{A\uparrow}(q), \quad (12)$$

where the set Child(p) contains the children of node p, and the computation for the node will be complete only if its child nodes have already been computed. Therefore, all the nodes have been aggregated by their low-grade nodes. Then, the tree structure is traversed from root to leaves, and the final aggregated cost value of pixel p is computed as follows:

$$\begin{aligned} C_d^A(p) &= S(\mathrm{Parent}(p), p)\cdot C_d^A(\mathrm{Parent}(p)) \\ &\quad + \left(1 - S^2(\mathrm{Parent}(p), p)\cdot C_d^{A\uparrow}(p)\right), \end{aligned} \quad (13)$$

where Parent(p) denotes the parent node of pixel p. After that, all the pixels eventually obtain a reliable aggregated cost. The complexity of computation is $O(n\cdot d)$, where n denotes the number of pixels in the guidance image and d denotes the disparity level.

3.4. Disparity Computation and Refinement. This subsection describes the universal winner-takes-all strategy, which is employed to seek the appropriate disparity level. And it carries the lowest matching cost, as shown in

$$D(p) = \arg\min_{d\in \mathrm{dislevel}} \left(C_d^A(p)\right), \quad (14)$$

where set dislevel denotes the disparity level.

We employ a tree structure to refine the coarse disparity map. First, we use the left and right images as guidance images, respectively. And the tree filter is executed twice, receiving two corresponding disparity maps. Then, we employ left and right consistency checks to mark the mismatched pixels and store them in set P_{mis}. For the left disparity map D, the cost value $C_{\mathrm{new}}(p,d)$ for each pixel p at each disparity d is recalculated as follows:

$$C_{\mathrm{new}}(p,d) = \begin{cases} 0, & p\in P_{\mathrm{mis}} \\ |d - D(p)|, & \text{else}, \end{cases} \quad (15)$$

where $D(p)$ denotes the initial disparity of pixel p. This method uses the tree structure mentioned above to execute the tree filter, and the process of creating a new mathematical model has no extra computation cost. The total running time is taken by recalculating the cost value and executing the tree filter. Furthermore, all the pixels with unstable disparity are marked as mismatch pixels, and the cost value of each

Table 1: Comparison of computational complexity.

Process	Complexity of computation	
	Tree construction	Cost aggregation
MST	$O(e + n + 2e \cdot \log_2 n)$	$O(n \cdot d)$
ST-1	$O(2 \cdot (e + e\alpha(e)))$	$O(n \cdot d)$
ST-2	$O(4 \cdot (e + e\alpha(e)))$	$O(2n \cdot d)$
Our proposed method	$O(2 \cdot (2e + e\alpha(e)))$	$O(n \cdot d)$

disparity level is set to zero. Only pixels with stable and precise disparity participate in aggregating the new cost value. The mismatched pixels achieve their final disparity value through the propagation of stable pixels afterwards.

This postprocessing technique has two advantages. A great advantage is that it is a nonlocal method and the whole stable and precise pixels contribute to the mismatched pixels. Another great advantage is that the tree structure is ready-made and the additional computational cost is negligible. The computation of the tree filter has an extremely low cost as well.

Moreover, we can further refine the disparity by means of (9) as mentioned above. Here, this equation can be regarded as a standard method for image segmentation. By comparing the boundaries of the disparity map with those of other segmented maps to mark the blurry regions, we can execute the tree filter again to obtain a disparity map with higher precision and more elaborate boundaries.

3.5. Complexity of Computation. We mainly analyze the computational complexity of tree construction and the cost aggregation in this section. Let n denote the number of pixels in image I and e denote the number of edges. The computation of tree construction in MST concentrates on the calculation of edge weights and node connections. The calculation of edge weight is $O(e)$. The pixels connections are divided into *find* and *connect* operations. The *find* operation requires $O(2e \cdot \log_2 n)$, and the complexity of the *connect* operation is determined only by n, so the total computation of tree construction in MST is $O(e + n + 2e \cdot \log_2 n)$.

As shown in Table 1, compared with MST, ST-1 must execute more *find* operations due to the constraint condition. So, the complexity of tree construction in ST-1 is $O(2 \cdot (e + e\alpha(e)))$, but in ST-2, it is $O(4 \cdot (e + e\alpha(e))$ according to [17]. Therefore, the computational complexity of tree construction in our proposed method is $O(2 \cdot (2e + e\alpha(e)))$, which is slightly larger than ST-1 due to the multiple components of weight function. As for cost aggregation, let d denote the disparity level. Therefore, it is ordinary to deduce the computational complexity of aggregation. The cost aggregation computation complexity of MST, ST-1, and our proposed method is $O(n \cdot d)$ while ST-2 is 2 times slower. Our proposed method requires more computations than some nonlocal cost aggregation methods but only on an extremely small scale.

4. Experimental Results

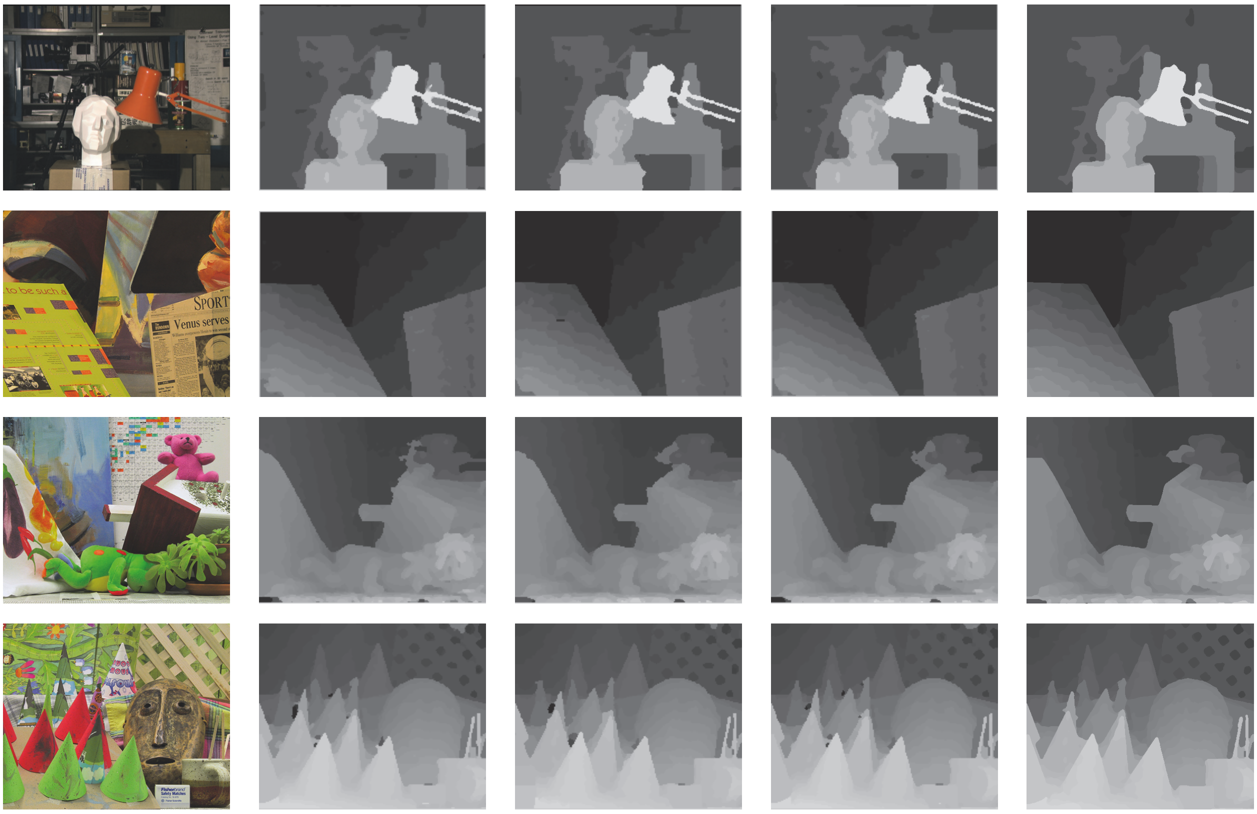
This section compares three mature nonlocal cost aggregation methods (MST [15], ST-1, and ST-2 [17]) with our proposed method. We tested our method using four standard Middlebury datasets [30] (Tsukuba, Venus, Teddy, and Cones). The MST and ST methods use an AD-Gradient measure [31] as the matching cost, while our proposed method employs the improved AD-Gradient method mentioned in Section 3. Moreover, the initial disparity for all the methods is computed by a WTA strategy. Finally, the postprocessing for each method involves nonlocal disparity refinement using their own tree structures. The parameters for our proposed method are defined as follows: $\alpha = \beta = 0.2$, $\varepsilon_c = 7$, $\varepsilon_g = 2$, $\theta_{\text{In}} = 0.2$, $\sigma = 0.08$, and $\tau = 1200$, and the parameters of MST and ST methods follow the relevant cited papers. The performance is tested on a PC with a 3.40 GHz CPU and 4 GB of memory.

Figure 5 shows the results of the four standard Middlebury datasets with these methods described above. The performance of ST-2 is better than that of ST-1 and MST in most typical regions when the boundaries of ST-2 are quite expressive. Our proposed methods' performance on the areas around the eaves near Teddy (the occluded regions) is particularly excellent. On Tsukuba, the angle of the table, where the foreground objects and the background have similar color contributions, is resolved faultlessly. In addition, the results of our proposed method are more satisfactory than the results of ST-2; the boundaries of the disparity maps are extremely smooth and precise. The typically tough regions such as the discontinuity regions and low-texture areas both achieve a good performance. However, our proposed method also fails in some regions, especially in the areas around the cones in the Cones datasets. The inner pixels of the cones contribute too much to the mismatch of the pixels outside, and the areas between any two cones do not achieve desirable results. The regions between the lamp and the table in Tsukuba are affected by various regions and, finally, obtain incorrect results.

More intuitive results are shown in Table 2. ST-1 is slightly better than MST, while the performance of ST-2 is better than both. Moreover, our proposed method obtains the best performance among these four algorithms. Compared with three classical methods, the number of erroneous pixels is reduced efficiently to between 5% and 15%.

We further tested 16 extra Middlebury datasets. The quantitative evaluation results are shown in Table 3. Only nonoccluded regions are evaluated in this table. First, ST-1 has the worst average rank. However, the average ranks are nearly equal between ST-2 and MST. Nevertheless, the average percentages of erroneous pixels in the three nonlocal methods are extremely close to one another. Besides, our proposed method achieves a tremendous advance, whether to compare the average percentage of erroneous pixels or the average rank. The percentages of erroneous pixels decline distinctly in *Baby*3, *Lampshade*1, and *Laundry*. However, the performance of some images (*Baby*1, *Baby*2, *Books*, *clothes*2, and *Moebius*) exhibits negative growth.

We selected four representative images from the extra 16 datasets (*Baby*3, *Flowerpots*, *Lampshade*1, and *Wood*1) to show the superiority of our proposed method through a visual comparison. The results are shown in Figure 6. Compared to the other nonlocal methods, our proposed



(a) Guidance image (b) MST (c) ST-1 (d) ST-2 (e) Proposed

Figure 5: The final disparity maps of the four most common images in the standard Middlebury datasets. (a) denotes the guidance images. From top to bottom, these are Tsukuba, Venus, Teddy, and Cones. The subfigures (b) to (e) show the disparity maps computed by different nonlocal methods. (b) shows the results of MST [15]; (c) and (d) show the results of the two segment-tree cost aggregations [17], respectively, and (e) shows the results of our proposed method.

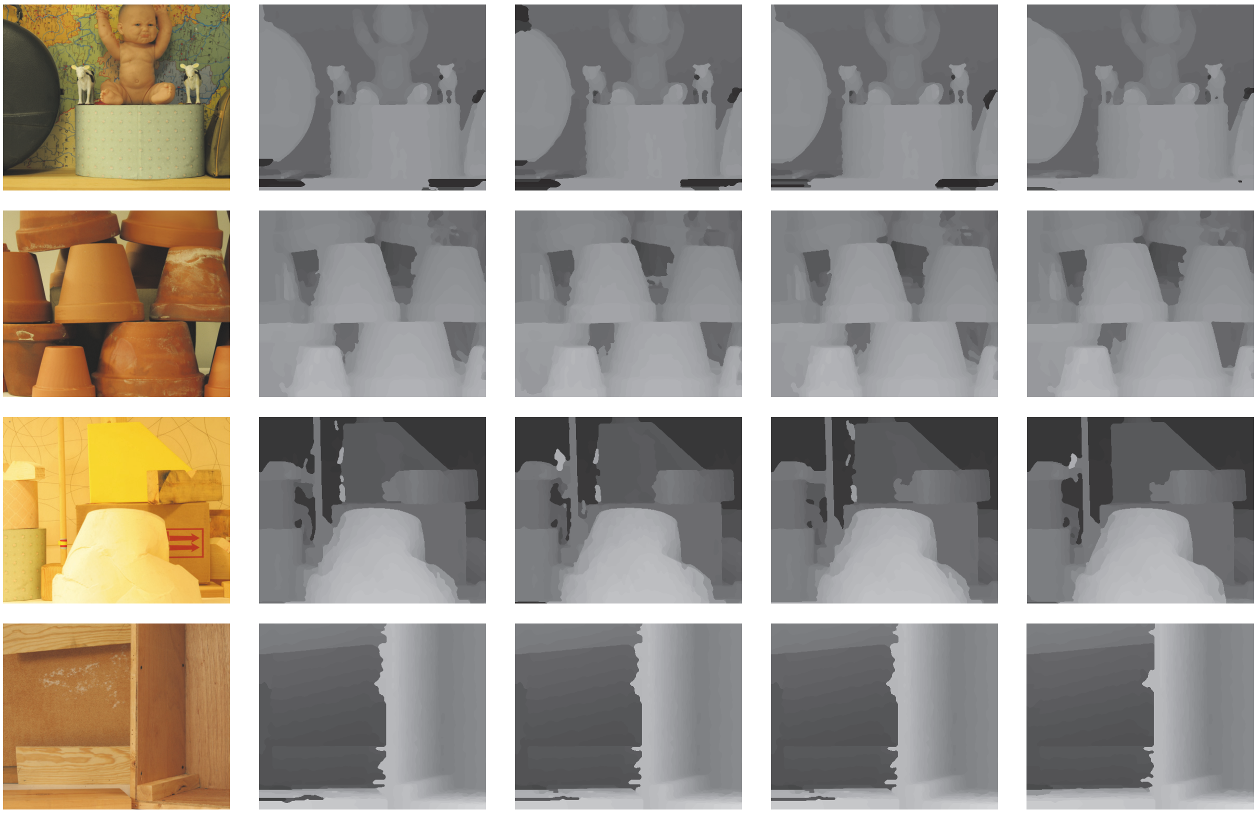
Table 2: Comparison of the four nonlocal algorithms (MST [15], ST-1 [17], ST-2 [17], and the proposed method) with Middlebury datasets and the standard of benchmark. The error threshold is set to 1 and three regions (nonocc, all, and disc) are used to evaluate the performance of the methods. Our proposed method exhibits the best accuracy in every region.

Algorithm	Avg.error	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
MST	5.73	1.50	2.18	8.02	0.42	0.85	5.02	5.95	10.89	14.15	3.14	8.68	7.94
ST-1	5.66	1.73	2.52	9.22	0.47	0.71	4.56	6.11	10.88	14.53	2.47	8.28	7.11
ST-2	5.18	1.35	2.00	7.29	0.42	0.69	5.27	5.17	9.95	12.95	2.49	7.90	6.62
Proposed	4.92	1.34	1.77	7.14	0.44	0.64	4.49	5.03	9.57	12.86	2.12	7.24	6.29

method achieves superior results, resulting in a more accurate disparity map and more reliable boundaries.

In *Lampshade*1, the results are adversely affected by illumination. Although other methods fail to detect the authentic boundaries, our method produces a better result. For example, the boundaries of the yellow trapezoid block are extremely close to the ground-truth map. As for *Wood*1, nearly the entire image contributes a similar color intensity. Therefore, it is crucial to calculate a rational result from the discontinuous regions. Unfortunately, all the other methods fail to detect clear boundaries on these datasets. However, the percentage of erroneous pixels declined to 2.49% by using our proposed method, which improves on the other nonlocal methods.

We mentioned the computational complexity in Section 3.5. In this section, we test 4 datasets and the average time consumption of each nonlocal method. The results are listed in Table 4. Most of the time is consumed during tree construction and tree filter requires only a slight amount of time. Moreover, MST is the shortest among the four methods, while our proposed method is a bit shorter than ST-2. The superiority of the proposed improved method over MST, ST-1, and ST-2 methods is demonstrated on experimental results (Tables 2 and 3, Figures 5 and 6). Moreover, in contrast to

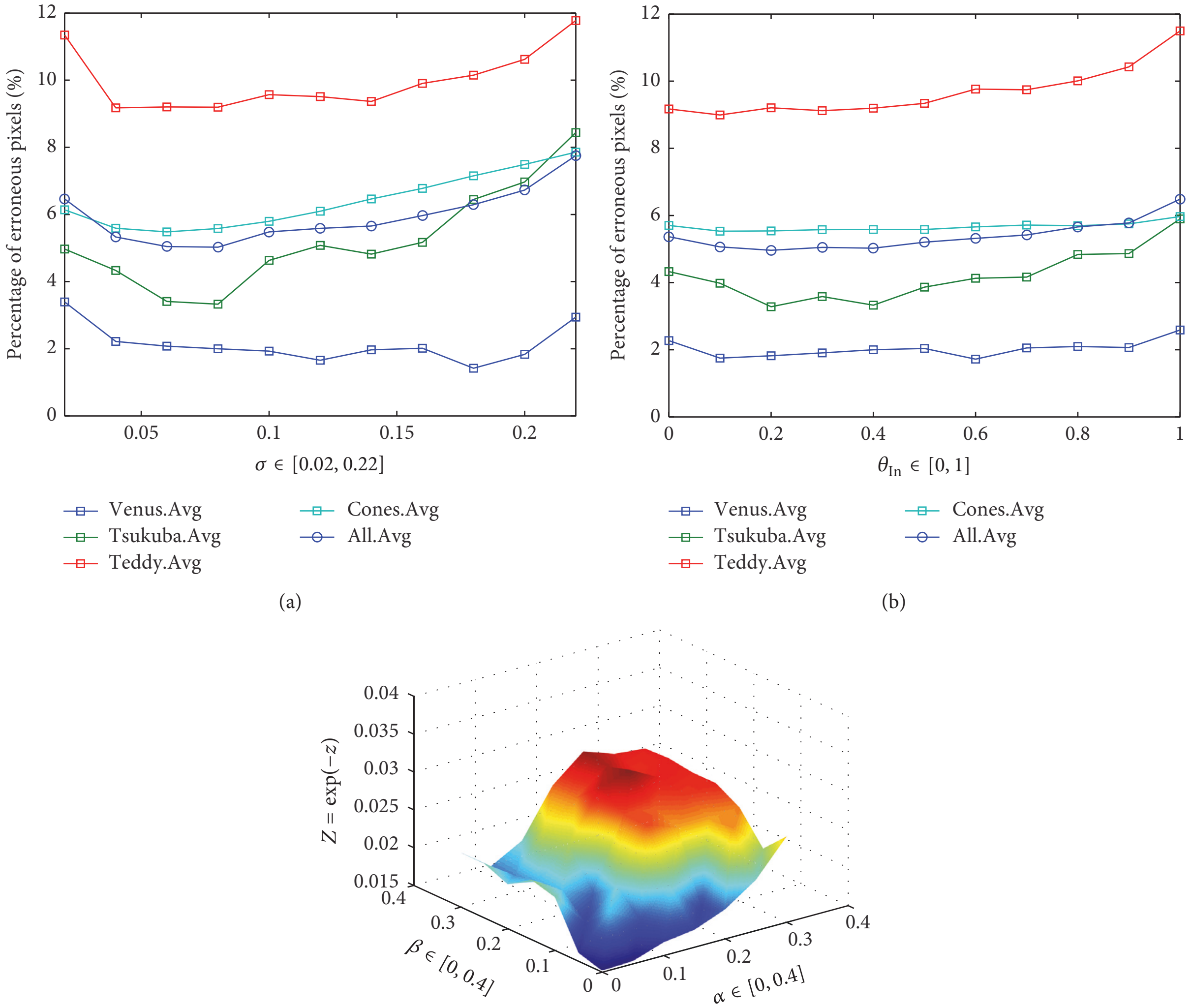


(a) Guidance image (b) MST (c) ST-1 (d) ST-2 (e) Proposed

Figure 6: The final disparity maps of the extra Middlebury datasets. Four representative images were selected to show the superiority of our proposed method. (a) denotes the guidance images. From top to bottom, these are Baby3, Flowerpots, Lampshade1, and Laundry. Subfigures (b) to (e) show the disparity maps computed by different nonlocal methods. (b) shows the results of MST [15]; (c) and (d) show the results of the two segment-tree cost aggregations [17], respectively, and (e) shows the results of our proposed method.

Table 3: The comparison of the four nonlocal algorithms (MST [15], ST-1, ST-2 [17], and the proposed method) with 16 extra Middlebury datasets. The error threshold is set to 1 and only nonoccluded regions are used to evaluate the performance of the methods.

Data	MST	ST-1	ST-2	Proposed
Aloe	8.41_3	9.50_4	8.37_2	7.42_1
Art	13.96_2	14.75_4	13.99_3	12.83_1
Baby1	4.67_3	4.98_4	4.52_1	4.58_2
Baby2	7.93_2	9.1_4	7.50_1	8.53_3
Baby3	6.44_2	6.82_4	6.52_3	3.93_1
Books	6.10_1	6.29_3	6.19_2	6.82_4
Cloth2	2.64_3	2.84_4	2.58_1	2.59_2
Cloth3	1.69_3	2.22_4	1.65_2	1.28_1
Dolls	5.11_3	5.07_2	5.46_4	4.64_1
Flowerpots	9.96_4	9.81_2	9.86_3	9.17_1
Lampshade1	8.56_3	8.43_2	8.82_4	6.80_1
Laundry	16.54_2	16.63_3	16.77_4	14.18_1
Midd1	28.03_4	23.41_2	24.46_3	22.05_1
Moebius	8.66_2	9.34_4	8.35_1	8.91_3
Reindeer	8.99_3	9.15_4	8.68_2	7.53_1
Wood1	4.05_3	4.75_4	3.91_2	2.49_1
Avg.Error	8.88	8.94	8.60	7.74
Avg.Rank	2.69	3.38	2.38	1.56



(a)

(b)

(c)

Figure 7: Parameter sensitivity analysis of our experiments. Four standard datasets (Tsukuba, Venus, Teddy, and Cones) were used in this experiment. (a) is a line chart representing the percentage of error pixels as parameter σ increases from [0.02 to 0.22], whereas (b) is a line chart representing the changes as parameter θ_{In} increases from [0 to 1]. Avg denotes the average percentage of erroneous pixels for three evaluation regions (nonocc, all, and disc), and All denotes the four standard datasets. (c) represents the average number of erroneous pixels from the four standard datasets using different weights for the initial components; an exponential function is employed to make the results more intuitive; $\alpha \in [0, 0.4]$ denotes the weight of color cost and $\beta \in [0, 0.4]$ denotes the weight of the vertical gradient cost.

MST and ST-1, the overall runtime cost of our proposed method does not increase obviously and is even shorter than ST-2. In contrast to the color-gradient based matching cost computation method proposed by Rhemann et al. [31], our method also has higher accuracy.

Table 4: Average time consumption for each nonlocal method with 4 Middlebury datasets.

Process	Overall runtime (seconds)			
	MST	ST-1	ST-2	Proposed
Tree construction	0.870	0.894	1.740	1.360
Cost aggregation	0.108	0.110	0.218	0.108
Whole process	0.978	1.004	1.958	1.468

5. Parameter Setting

Several parameters are used in our proposed method. ε_c and ε_g are user-specified parameters used for adjustment in (4). They follow the truncated value in [31] while the predefined parameter $\tau = 1200$ in the tree construction follows the settings of the segment-tree [17] method. In this section, we discuss the rationale and sensitivity of the remaining four parameters, the weights for each component (α and β) in the initial computation, the predefined weight of inner color correlation (θ_{In}) in tree construction, and the adjustment value (σ) of the weight function.

First, we test the adjustment value (σ) of (10). The results are shown in Figure 7(a). When $\sigma \in [0.04, 0.14]$, the experimental results from most of the images are extremely low and vary slightly. In contrast, the erroneous pixels decline to a minimum when $\sigma \in [0.06, 0.08]$, which is due to the variation in the initial cost value. We employ the *Geman – McClure* function to protect the initial cost value from the encroachment of extremum, and the initial cost value converges to 1. With the adjustment of the initial cost value, a parameter σ is required to be adjusted accordingly, or disparity boundaries will be unclear and foreground objects will be confused with background.

As for the weight of the inner color correlation θ_{In}, the parameter range of this experiment is 0 to 1. More details are shown in Figure 7(b). The percentage of erroneous pixels increases significantly when the parameter $\theta_{\text{In}} \in [0.2, 0.4]$. The experimental results show that employing inner color correlation is obviously reasonable but the parameter θ_{In} should be confined to 0.5 or below.

Figure 7(c) evaluates the sensitivity of the initial component weights α and β with four original Middlebury datasets, to clarify that the final results (percentage of erroneous pixels) are processed by an exponential function. The figure shows that the algorithm achieves its best performance when the parameters α and $\beta \in [0.15, 0.3]$. The range of the parameters that achieve dramatic performance is much larger than the original nonlocal methods. And Figure 7 further demonstrates that employing the *Geman – McClure* function helps to resolve the errors caused by outliers more effectively and robustly than the methods described above which use truncated values.

6. Conclusion

In this paper, our work is directly motivated by two original algorithms [15, 17]. We propose an improved nonlocal cost aggregation algorithm based on them. The proposed method is developed with modified initial cost and multiple weight for stereo matching, which modifies the original algorithm in both computational cost and aggregation. Our method has some advantages. First, it has higher accuracy by adding the vertical gradient as one of the components in the process of cost computation. Particularly, the performance near some discontinuous areas is much better than that of other methods. Second, due to its stability and versatility, inner color correlation is employed instead of using a single color component. Thus, it makes constructing a tree structure more robust and reasonable. Besides, we modify the structure of the segment tree.

The performance was tested on a PC with a 3.40 GHz CPU and 4 GB of memory. The proposed method was evaluated on Middlebury datasets. The experimental results verified that our proposed method could achieve better accuracy with a minor cost of increased execution time. In the near future, we would like to focus on more novel tree structures. And we will continue to study nonlocal methods and image segmentation, proposing new ideas to resolve the issues mentioned above.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61471150), the International Cooperation and Exchange of the National Natural Science Foundation of China (Grant no. 2014DFA12040), and Zhejiang Provincial Natural Science Foundation of China (Grant no. LY13F020033).

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 7–42, 2002.

[2] C. Lei, J. Selzer, and Y.-H. Yang, "Region-tree based stereo using dynamic programming optimization," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*, pp. 2378–2385, June 2006.

[3] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, pp. 74–81, IEEE, 2004.

[4] Q. Yang, L. Wang, and N. Ahuja, "A constant-space belief propagation algorithm for stereo matching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1458–1465, San Francisco, Calif, USA, June 2010.

[5] H. Hirschmüller, "Accurate and efficient stereo processing by Semi-Global Matching and Mutual Information," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, pp. 807–814, June 2005.

[6] S. K. Gehrig and C. Rabe, "Real-time semi-global matching on the CPU," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, pp. 85–92, San Francisco, CA, USA, June 2010.

[7] S. Gehrig, F. Eberli, and T. Meyer, "A real-time low-power stereo vision engine using semi-global matching," in *Proceedings of the the 7th International Conference on Computer Vision Systems*, M. Fritz, B. Schiele, and P. H. Justus, Eds., vol. 5815 of *LNCS*, pp. 134–143, Springer, Berlin, Germany, 2009.

[8] F. Schumacher and T. Greiner, "Matching cost computation algorithm and high speed FPGA architecture for high quality real-time Semi global matching stereo vision for road scenes," in *Proceedings of the 2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, pp. 3064–3069, Quingdao, China, October 2014.

[9] F. Cheng, H. Zhang, M. Sun, and D. Yuan, "Cross-trees, edge and superpixel priors-based cost aggregation for stereo matching," *Pattern Recognition*, vol. 48, no. 7, pp. 2269–2278, 2015.

[10] 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.

[11] 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.

[12] L. Zhou, G. Xu, K. Li, B. Wang, Y. Tian, and X. Chen, "Stereo matching algorithm based on census transform and modified

adaptive windows," *Acta Aeronautica et Astronautica Sinica*, vol. 33, no. 5, pp. 886–892, 2012.

[13] Y. Qu, J. Jiang, X. Deng, and Y. Zheng, "Robust local stereo matching under varying radiometric conditions," *IET Computer Vision*, vol. 8, no. 4, pp. 263–276, 2014.

[14] 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.

[15] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1402–1409, Providence, RI, USA, June 2012.

[16] Q. Yang, "Stereo matching using tree filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 834–846, 2015.

[17] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang, "Segment-tree based cost aggregation for stereo matching," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 313–320, June 2013.

[18] K. Zhang, Y. Fang, D. Min et al., "Cross-scale cost aggregation for stereo matching," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 407–414, June 2014.

[19] Y. Peng, Z. Hua, X. Yanbing et al., "Iterative color-depth MST cost aggregation for stereo matching," in *Proceedings of the 2016 IEEE International Conference on Multimedia and Expo, ICME 2016*, pp. 1–6, July 2016.

[20] C. C. Pham, V. Q. Dinh, and J. W. Jeon, "Robust non-local stereo matching for outdoor driving images using segment-simple-tree," *Signal Processing: Image Communication*, vol. 39, pp. 173–184, 2015.

[21] M. Gerrits and P. Bekaert, "Local stereo matching with segmentation-based outlier rejection," in *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision, CRV 2006*, pp. 66–72, June 2006.

[22] J. Chen, C. Cai, and C. Li, "A multi-window stereo matching algorithm in rank tranform domain," in *Proceedings of the 2012 11th International Conference on Signal Processing, ICSP 2012*, pp. 997–1000, October 2012.

[23] V. Q. Dinh, D. D. Nguyen, V. Dinh Nguyen, and J. W. Jeon, "Local stereo matching using an variable window, census transform and an edge-preserving filter," in *Proceedings of the 2012 12th International Conference on Control, Automation and Systems, ICCAS 2012*, pp. 523–528, October 2012.

[24] G.-B. Kim and S.-C. Chung, "An accurate and robust stereo matching algorithm with variable windows for 3D measurements," *Mechatronics*, vol. 14, no. 6, pp. 715–735, 2004.

[25] Z. Gu, X. Su, Y. Liu, and Q. Zhang, "Local stereo matching with adaptive support-weight, rank transform and disparity calibration," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1230–1235, 2008.

[26] F. Tombari, S. Mattoccia, and L. Di Stefano, "Segmentation-based adaptive support for accurate stereo correspondence," in *Advances in Image and Video Technology*, vol. 4872 of *Lecture Notes in Computer Science*, pp. 427–438, Springer, Berlin, Germany, 2007.

[27] 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.

[28] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *Proceedings of the 16th IEEE International Conference on Image Processing, ICIP 2009*, pp. 2069–2072, November 2009.

[29] D. Chen, M. Ardabilian, X. Wang, and L. Chen, "An improved Non-Local Cost Aggregation method for stereo matching based on color and boundary cue," in *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo, ICME 2013*, pp. 1–6, July 2013.

[30] D. Scjarstein and R. Szeliski, Middlebury stereo evaluation, 2012 http://vision.middlebury.edu/stereo/eval/.

[31] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3017–3024, June 2011.

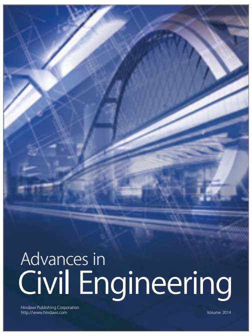
Journal of
Engineering

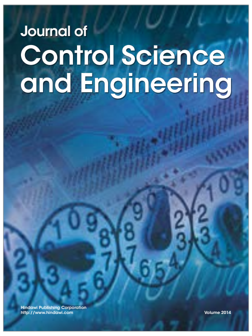
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

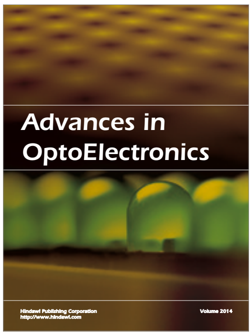
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

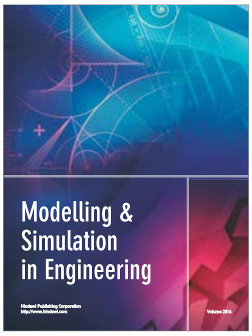
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

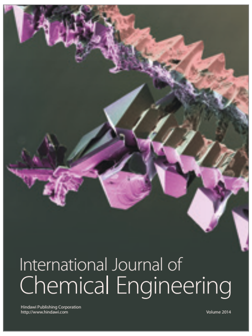

Advances in
OptoElectronics

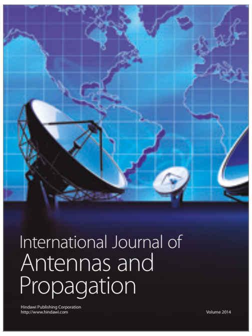
VLSI Design

International Journal of
Navigation and
Observation

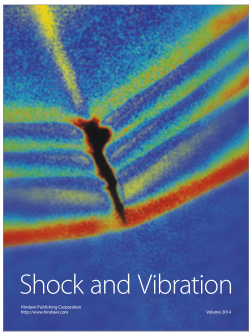

Modelling &
Simulation
in Engineering

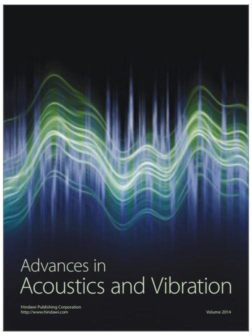
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration