Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 654139, 11 pages
http://dx.doi.org/10.1155/2013/654139

Research Article

Matching Cost Filtering for Dense Stereo Correspondence

Yimin Lin,[1,2] Naiguang Lu,[1,2] Xiaoping Lou,[2] Fang Zou,[3] Yanbin Yao,[3] and Zhaocai Du[3]

[1] *Institute of Optical Communication & Optoelectronics, Beijing University of Posts & Telecommunications, Beijing 100876, China*
[2] *School of Instrumentation Science & Optoelectronics Engineering, Beijing Information Science & Technology University, Beijing 100192, China*
[3] *Beijing Aeronautical Manufacturing Technology Research Institute, Beijing 100024, China*

Correspondence should be addressed to Yimin Lin; linyimin2012@hotmail.com and Naiguang Lu; nglv2002@sina.com

Received 4 July 2013; Accepted 27 August 2013

Academic Editor: Vishal Bhatnaga

Dense stereo correspondence enabling reconstruction of depth information in a scene is of great importance in the field of computer vision. Recently, some local solutions based on matching cost filtering with an edge-preserving filter have been proved to be capable of achieving more accuracy than global approaches. Unfortunately, the computational complexity of these algorithms is quadratically related to the window size used to aggregate the matching costs. The recent trend has been to pursue higher accuracy with greater efficiency in execution. Therefore, this paper proposes a new cost-aggregation module to compute the matching responses for all the image pixels at a set of sampling points generated by a hierarchical clustering algorithm. The complexity of this implementation is linear both in the number of image pixels and the number of clusters. Experimental results demonstrate that the proposed algorithm outperforms state-of-the-art local methods in terms of both accuracy and speed. Moreover, performance tests indicate that parameters such as the height of the hierarchical binary tree and the spatial and range standard deviations have a significant influence on time consumption and the accuracy of disparity maps.

1. Introduction

Stereo correspondence between stereo images results in a depth image, also called a disparity map, which can be categorized as sparse or dense. Sparse disparity maps are obtained mainly using feature-based methods derived from human vision research [1]. As a result, high processing speeds and accurate disparity maps are achieved but without high density, which has limited their use for many purposes. Dense stereo correspondence, which aims to figure out which parts of an image correspond to which parts of another image, is a challenging issue in the field of computer vision. The requirement of dense disparity maps is motivated by many contemporary applications such as virtual reality, view synthesis, and robot vision navigation [2].

Dense stereo correspondence algorithms can be classified as global or local according to whether they obtain disparities from global or local information. The goal of global methods (energy based) is to minimize a global cost function which combines matching costs and smoothness terms, depending on information derived from the whole image. These methods are time consuming but very accurate [3]. On the other hand, local methods (area based) offer high speed at the expense of matching accuracy and determine the degree of disparity of each pixel according to information provided by its local and neighboring pixels. These methods are also referred to as window-based methods because the disparity computation between two matching pairs depends only on the intensity values within a fixed-size and fixed-shape matching window [4]. However, recent studies have shown that, by ingeniously selecting and aggregating the matching costs of neighboring pixels, the disparity maps produced by a local approach can be more accurate than those generated by global methods [5]. The most noteworthy technique is local filtering, which is an effective way to reduce matching noise and is able to generate high-quality disparity maps.

This paper proposes a dense stereo correspondence approach very similar to the original adaptive support weight

(ASW) method [6] to obtain accurate disparity maps both in depth discontinuities and smooth regions. The basic idea is to accept similar pixels within a matching window by assigning them relatively large support weights and to reject dissimilar pixels by giving them very small support weights. Therefore, it is necessary to divide the neighboring pixels into similar and dissimilar groups. In the present case, adaptive support weights are computed from the color image using a hierarchical clustering algorithm inspired by Gastal's work [7] in high-dimensional filtering of images and videos in real time; the disparity maps after filtering are less noisy, and the depth discontinuity boundaries are preserved fairly well. In addition, the proposed algorithm has improved the results for efficiency and accuracy compared with the guided-image filter (GIF) [8] algorithm used for stereo correspondence, which is by far the best existing algorithm.

The main contributions of this paper include the following.

(1) A novel matching-cost filtering model is proposed based on an edge-preserving filter for which the adaptive support weights are computed using a hierarchical clustering algorithm (as shown in Section 3.2). This solution can reduce mismatching, especially around regions of depth discontinuities, and can reconstruct dense high-accuracy disparity maps.

(2) The computational complexity of the proposed method is essentially linear both in the number of image pixels and the number of clusters, regardless of the matching window size and the intensity range (as described in Section 3.3). Therefore, the method can be easily adjusted to meet real-time requirements with the help of contemporary graphics hardware (a graphics processing unit (GPU)).

(3) A new disparity refinement method is presented, which has been proved to be robust and effective for improving the accuracy of coarse disparity maps (as presented in Section 3.5). This method can be applied to other coarse-to-fine frameworks, which are among the classic, simplest, and most popular stereo matching algorithms.

(4) The influence of algorithm parameters on accuracy and efficiency is discussed, especially regarding the weight coefficient, the height of the hierarchical binary tree, and the size of the spatial and range standard deviations (as discussed in Section 4.2). This study offers recommendations which can be used as a basis for future practical applications.

The rest of this paper is organized as follows: Section 2 describes an overview of the state-of-the-art local filtering methods and our method will be proposed in Section 3. Section 4 presents experimental results which compare the proposed method with other state-of-the-art approaches and discusses the influences of parameter settings. Finally, conclusions and suggestions for future work are discussed in Section 5.

2. Related Work

A disparity map is obtained by determining the disparity which has the lowest matching cost in each local matching window, a method which is widely used in local algorithms. Many local methods have been proposed to obtain a dense disparity map recently. For instance, adaptive-window methods [9, 10] try to find an optimal matching window for each pixel, and multiple-window methods [11] select an optimal matching window among predefined multiple windows located at different positions with the same shape. However, these methods have one limitation in common: the shape of the matching window is constrained to be a rectangle, which is not appropriate for pixels near depth discontinuities. Therefore, it is difficult to find an optimal matching window with an appropriate size and shape for all cases.

Instead of searching for an optimal matching window of arbitrary size and shape, it is possible to aggregate costs after local smoothing within a matching window to reduce matching noise. It is clear that most noise can be reduced effectively by a linear filter, such as Gaussian filter, but the disparity map always results in a well-known "edge-fattening" phenomenon. Therefore, the local filtering results will not be a good neighborhood representative close to an edge region. To address this problem, the recently proposed ASW algorithm [6] smoothes the matching costs with an adaptive weighted filter in which the support weights are chosen according to both the color similarity and the Euclidean distance to the center pixel. These methods imitate the way that humans assign different weights to a pixel according to color or brightness in the process of finding the correspondences between their two eyes. Such a filter is also referred to as an edge-preserving filter in computer vision and is widely used for image denoising; examples include the SUSAN filter [12], bilateral filter [13], and the nonlocal means filter [14, 15]. Experimental results show that this approach can produce disparity maps better than those generated using global optimization techniques without needing many user-specified parameters. Although this method leads to high-quality results, its computational speed presents a problem because runtime is computationally expensive. Therefore, many improved and real-time solutions have been presented, such as the O(1) bilateral filter [16–18], the dual-cross-bilateral grid (DCBG) [19, 20], the GIF [21, 22], and the nonlocal filter [23].

3. Cost Aggregation with Local Filtering

A literature review has provided a taxonomy and an evaluation of typical matching algorithms and has emphasized that such a coarse-to-fine algorithm generally performs the following four steps [24]:

(1) cost initialization, in which the matching costs for assigning different disparity hypotheses to different pixels are calculated;

(2) cost aggregation, in which the initial matching costs are aggregated spatially over matching windows;

(3) disparity optimization, in which a cost function is minimized to obtain the best disparity hypothesis for each pixel;

(4) disparity refinement, in which the coarse disparity maps are postprocessed to remove mismatches or to generate fine disparity maps.

According to these four steps, in this paper, the cost aggregation with local filtering consists of five parts: matching cost initialization, cost aggregation with filtering, clustering range values for the sampling points, disparity selection, and refinement. In addition, the computational complexity is discussed.

3.1. Cost Initialization. Generally, it is possible to identify matching pairs in stereo images by measuring their similarity. The most common algorithms which use a matching cost function to establish a correspondence between the two points are the sum of absolute intensity differences (SAD), the sum of squared intensity differences (SSD), and the normalized cross-correlation (NCC) [25].

The cost initialization module computes the initial matching cost $M(u, v, d)$ for assigning disparity hypothesis d to image pixel (u, v), where u, v define the displacements in the x- and y-directions, respectively. Generally, after rectifying a stereo image, there is no shift in the y-direction except for the displacement in the x-direction, in which case the cost can be represented as $M(u, d)$ according to the disparity d. The costs are calculated using the truncated absolute differences in range (intensity or color) and the gradient between corresponding pixels. In other words,

$$\begin{aligned} M(u,d) &= a \cdot \min\left[\left|I_L(u) - I_R(u-d)\right|, \tau_1\right] \\ &\quad + (1-a) \cdot \min\left[\left|\nabla_x I_L(u) - \nabla_x I_R(u-d)\right|, \tau_2\right], \end{aligned} \tag{1}$$

where a is the weight coefficient, $I_L(u)$ is the left image, and the corresponding right image which has disparity d is $I_R(u - d)$. ∇_x is the gray-scale gradients in the x-directions, and τ_1, τ_2 are truncation values for balancing the range and gradient terms. Such a matching cost model has been proved to be robust to illumination changes and is commonly used in stereo correspondence [26].

3.2. Cost Aggregation with Filtering. The original local filtering approach tried to compute the weights which are the average of the adjacent matching costs. The costs aggregated over the weights can therefore be expressed as

$$C(i,d) = \frac{\sum_{j \in N_i} W(i,j)\, M(j,d)}{\sum_{j \in N_i} W(i,j)}, \tag{2}$$

where i and j are pixel indices in the x-direction and N_i is the region around the ith coordinate.

The weights $W(i, j)$ of this linear combination are given by two Gaussian filter kernels which combine the spatial weights based on the distance between two pixels and the range weights based on the intensity difference. Therefore, the filter weights $W(i, j)$ can be represented by spatial and range terms as

$$W(i,j) = \exp\left(-\frac{\left|i-j\right|^2}{\sigma_s^2}\right) \exp\left(-\frac{\left|I_i - I_j\right|^2}{\sigma_r^2}\right), \tag{3}$$

where σ_s and σ_r are two constants used to adjust the spatial and range similarities.

The Gaussian over the range similarity R can be rewritten as a convolution using two Gaussian kernels:

$$\begin{aligned} R &= \exp\left(-\frac{\left|I_i - I_j\right|^2}{\sigma_r^2}\right) \\ &= C \int \exp\left(-\frac{\left|I_i - m\right|^2}{\sigma_r^2/2}\right) \exp\left(-\frac{\left|m - I_j\right|^2}{\sigma_r^2/2}\right) dm, \end{aligned} \tag{4}$$

where C is a normalization factor and m is a sampling range value. Finally, the range R for a Gaussian integral can be evaluated numerically using an approximation according to the Gauss-Hermite quadrature rule as

$$R = C \sum_{n=1}^{K} \exp\left(-\frac{\left|I_i - m_n\right|^2}{\sigma_r^2/2}\right) \exp\left(-\frac{\left|m_n - I_j\right|^2}{\sigma_r^2/2}\right), \tag{5}$$

where K is the number of sampling range values. Increasing the number of sampling points gives a better approximation for the integral in (4). Assuming that pixel i has a sampling set $\{m_{1i}, m_{2i}, ..., m_{Ki}\}$, the filter weights in (3) can be rewritten as

$$\begin{aligned} W(i,j) = {} & \exp\left(-\frac{\left|i-j\right|^2}{\sigma_s^2}\right) \\ & \times \sum_{n=1}^{K} \exp\left(-\frac{\left|I_i - m_{ni}\right|^2}{\sigma_r^2/2}\right) \exp\left(-\frac{\left|m_{ni} - I_j\right|^2}{\sigma_r^2/2}\right). \end{aligned} \tag{6}$$

The normalization factor C was not included because both of the numerator and denominator in (2) contain this factor and it will cancel out after the division.

3.3. Clustering Range Value for Sampling Points. As mentioned before, the key point of Yang's algorithm [23] is that it accepts similar pixels within a matching window by assigning them relatively large support weights and rejects dissimilar pixels by giving them very small support weights. Clearly, it is necessary to divide neighboring pixels into similar and dissimilar groups. Inspired by this opinion, the authors propose a hierarchical clustering algorithm similar to that developed by Gastal and Oliveira [7] to separate iteratively the whole set of image pixels from different range values into different clusters and to perform cost aggregation with local filtering within these clusters. This is actually an expansion

of the method of adaptive manifold filtering in stereo correspondence and results in a modified clustering algorithm.

Assume that pixel i and its neighboring pixel j within a cluster, where their nth sampling points have similar range values, satisfy

$$m_{ni} = m_{nj}. \tag{7}$$

Averaging values only from pixels belonging to the same cluster generates better estimates for the local filtering output. Therefore, after clustering range values for the sampling points, the cost aggregation in (2) can be rewritten using the filter weights in (6) and the cluster constraints in (7) as

$$\begin{aligned} C(i,d) = {} & \left(\sum_{n=1}^{K} \exp\left(-\frac{\left| I_i - m_{ni} \right|^2}{\sigma_r^2/2} \right) \right. \\ & \times \sum_{j \in N_i} \exp\left(-\frac{\left| i - j \right|^2}{\sigma_s^2} \right) \\ & \left. \times \exp\left(-\frac{\left| m_{nj} - I_j \right|^2}{\sigma_r^2/2} \right) M(j,d) \right) \\ & \times \left(\sum_{n=1}^{K} \exp\left(-\frac{\left| I_i - m_{ni} \right|^2}{\sigma_r^2/2} \right) \right. \\ & \times \sum_{j \in N_i} \exp\left(-\frac{\left| i - j \right|^2}{\sigma_s^2} \right) \\ & \left. \times \exp\left(-\frac{\left| m_{nj} - I_j \right|^2}{\sigma_r^2/2} \right) \right)^{-1}. \end{aligned} \tag{8}$$

Compared with the complexity of the original bilateral filter in (3), the proposed filter in (8) reduces the complexity from $O(N^2)$ to $O(KN)$, where $K \ll N$ and N is the number of pixels within the whole image.

After introducing the improved cost aggregation and complexity analysis, an algorithm for clustering the range values can be summarized as follows.

Step 1. Generate the first sampling point m_{1i} at pixel i by low-pass filtering the input signal I within neighborhood N_i:

$$m_{1i} = \frac{\sum_{k \in N_i} \exp\left(-|i-k|^2/\sigma_s^2 \right) I_{i,k}}{\sum_{k \in N_i} \exp\left(-|i-k|^2/\sigma_s^2 \right)}, \tag{9}$$

where $I_{i,k}$ represents the range value with distance k around pixel i.

Step 2. Generate the nth $(1 < n \le K)$ sampling point m_{ni}. The first step is to compute an optimal hyperplane R_n,

$$\Sigma \cdot R_n = \lambda_{\max} R_n, \quad \text{where } \lambda_{\max} = \max\left[\text{eigenvalue}(\Sigma) \right], \tag{10}$$

which corresponds to the eigenvector associated with the largest eigenvalue of the covariance matrix:

$$\Sigma = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - u)(x_i - u)^T, \tag{11}$$

where x_i is the difference between the range value I_i and the previous sampling point $m_{n-1,i}$ associated with each pixel i:

$$x_i = I_i - m_{n-1,i}, \tag{12}$$

and u is equal to the sum of the values x_i divided by the number of pixels N:

$$u = \frac{1}{N} \sum_{i=0}^{N-1} x_i. \tag{13}$$

Step 3. Segment the pixels into two clusters C^+ and C^- using the sign of the projection:

$$P_i = R_n^T x_i \in \begin{cases} C^+, & P_i \ge 0 \\ C^-, & P_i < 0. \end{cases} \tag{14}$$

Step 4. Compute a new sampling point m_{ni}^+ also by low-pass filtering the input signal, but giving weight zero to pixels not in C^+, as

$$\begin{aligned} m_{ni}^+ = {} & \left(\sum_{k \in (N_i \cap C^+)} (1 - w_{nk}) \right. \\ & \left. \times \exp\left(-\frac{|i-k|^2}{\sigma_s^2} \right) I_{i,k} \right) \\ & \times \left(\sum_{k \in (N_i \cap C^+)} (1 - w_{nk}) \right. \\ & \left. \times \exp\left(-\frac{|i-k|^2}{\sigma_s^2} \right) \right)^{-1}. \end{aligned} \tag{15}$$

The values w_{nk} are the weights calculated using the range value and the previous sampling points:

$$w_{nk} = \exp\left(-\frac{\left| I_{i,k} - m_{n-1,k} \right|^2}{\sigma_r^2/2} \right). \tag{16}$$

Perform the same processing for m_{ni}^- using pixels belonging to C^-; then the combination of m_{ni}^+ and m_{ni}^- is the whole set of sampling points m_{ni}.

Step 5. The number of sampling range values K determines whether more clusters are needed for sampling points. Therefore, the next step is to repeat recursively Step 2 onwards until $n = K$.

Remember that Steps 2 and 3 can be directly rewritten using the sign (positive or negative) of the differences when the range value is a gray one:

$$P_i = I_i - m_{n-1,i} \in \begin{cases} C^+, & P_i \ge 0 \\ C^-, & P_i < 0. \end{cases} \tag{17}$$

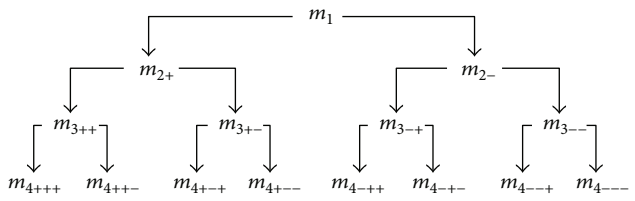


Figure 1: Hierarchical binary tree generated by the clustering algorithm.

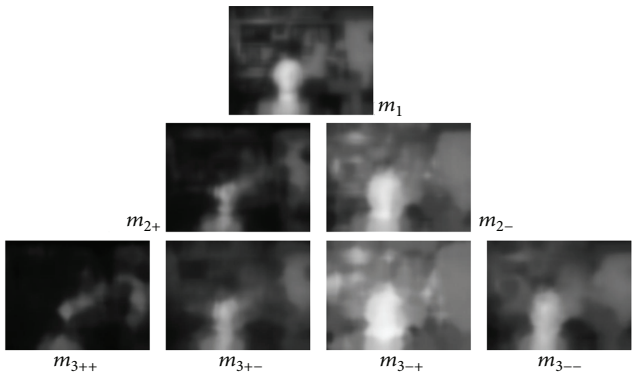


Figure 2: The first three levels of sampling points m_{ni} after clustering the range values.

These five steps serve to construct the hierarchical binary tree shown in Figure 1. The whole tree with height H has $K = 2^H - 1$ nodes. Each sampling point m_n has $2^n - 1$ nodes. For example, if $n = 3$, the third m_3 has four nodes $\{m_{3++}, m_{3+-}, m_{3-+}, m_{3--}\}$; note that the first subscript plus or minus is the same as the upper nodes and the second nodes generated by the current clustering procedure. The rest can be generated in the same manner.

At the top of the tree, the sampling points are better adapted to smooth regions. Points further down this tree would become gradually better adapted to edge regions.

Figure 2 shows the first three levels of sampling points m_{ni} of the Tsukuba image which was downloaded from the Middlebury benchmark database [27]. Based on clustering range values for more than one sampling points, the filtering results of (8) can be guaranteed to be an edge-preserving smoothing.

3.4. Disparity Optimization. Once the matching costs have been filtered using a cluster method, the disparity optimization step computes an optimal disparity map $D(u, v)$ using the local winner-takes-all (WTA) approach, which computes the coarse disparities associated with the minimum cost value at each pixel. In other words,

$$D(u, v) = \arg\min_{d \in L} C(u, v, d), \tag{18}$$

where $C(u, v, d)$ represents the matching cost obtained after cost aggregation for assigning a disparity hypothesis to pixel (u, v) and L is the number of disparity levels.

3.5. Disparity Refinement. The coarse disparity maps generated by WTA may contain some mismatches because local optimization does not obey the smoothness constraint. Therefore, a two-step postprocessing method for fine disparity maps is proposed.

The first step is a left and right cross-checking procedure for mismatches. Two corresponding disparity maps with the left and the right images as reference images are obtained. Then the left and right consistency check divides all the pixels into stable or unstable pixels. Note that all stable pixels in the left and right disparity maps have the same disparity value and that the rest of the pixels are labeled as unstable, represented by a value of zero for all disparity levels.

Secondly, let $D(u, v)$ represent the left disparity map; a new disparity space volume (DSV) [28] is then computed for each stable (S) or unstable (U) pixel (u, v) at each disparity level d as

$$\mathrm{DSV}(u, v, d) = \begin{cases} |d - D(u, v)| & (u, v) \in S \\ 0 & (u, v) \in U. \end{cases} \tag{19}$$

Then an edge-preserving filter such as GIF is applied to smooth the DSV at each disparity level, and the unstable pixels are assigned a new disparity value which depends on the lowest value of the DSV.

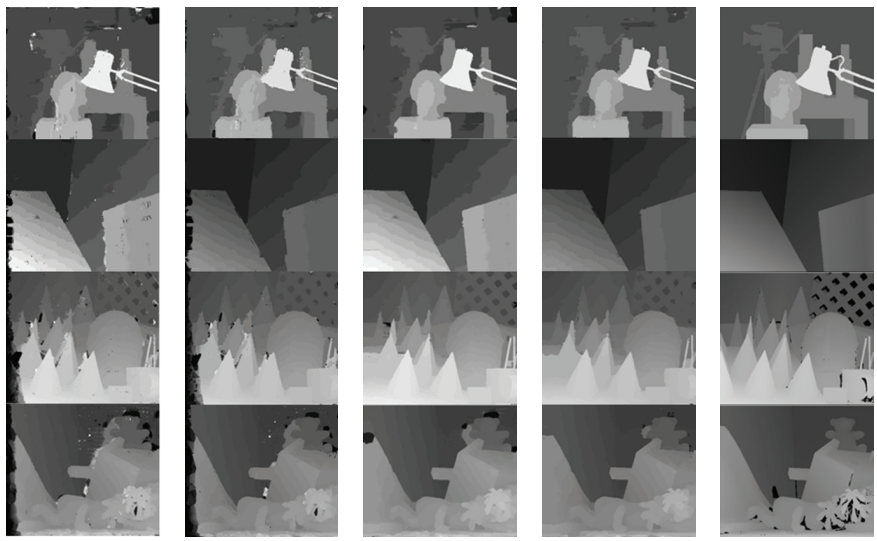
4. Experimental Results

In this section, the performance of the proposed method is evaluated using the Middlebury stereo benchmark, which provides stereo images with known ground truth [27]. The experimental results are then compared with other local filtering methods which have recently been proven to be the best edge-preserving local stereo methods in terms of both speed and accuracy on the Middlebury benchmark website. Therefore, the comparison results will serve to demonstrate that the proposed method performs well among all local stereo correspondence algorithms. Moreover, this section analyzes the impacts of different parameter settings on the computational complexity and accuracy of the dense disparity maps.

The proposed method was run with constant parameter settings for all four testing images: $\{a, \tau_1, \tau_2, H, \sigma_r, \sigma_s\} = \{0.1, 0.028, 0.08, 4, 0.08, 11\}$. To analyze and compare the quality of the stereo matching algorithms, a widely accepted quantitative performance evaluation criterion, the percentage of bad pixels (PBP), was introduced:

$$\mathrm{PBP} = \left[\frac{1}{N} \sum_{x,y} \left(\left| d_t(x, y) - d_g(x, y) \right| > \delta \right) \right] \times 100\%, \tag{20}$$

where N is the total number of pixels, d_t and d_g are the computed depth mapping and the ground truth mapping, and δ is an absolute disparity error threshold. A value of $\delta = 1$ was chosen in these experiments because this setting is the same as in some previously published studies. Hence, a smaller PBP number means a better-performing algorithm. The preferred metric (PBP) used in this paper, which is



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

Figure 3: Experimental results on the Middlebury benchmark. Dense disparity maps from the first to the last row are the "Tsukuba," "Venus," "Cones," and "Teddy" images. ((a) and (b)) The results of GIF and the proposed method without refinement procedure. ((c) and (d)) The disparity maps obtained using the GIF and the proposed method with refinement procedure. (e) Ground truth.

Table 1: Quantitative evaluation for the Middlebury image pairs.

Method	Tsukuba			Venus			Teddy			Cones			AE
	Non	All	Disc	Non	All	Disc	Non	All	Disc	Non	All	Disc	
HCR	1.56	1.78	8.07	0.22	0.34	2.96	6.36	11.93	15.62	2.88	8.14	8.22	**5.67**
GIR	1.87	2.23	7.92	0.27	0.47	2.60	6.74	12.28	16.20	2.94	8.35	8.36	**5.85**
ASW	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.30	18.60	3.97	9.79	8.26	**6.66**
HCN	2.14	2.94	9.16	1.25	1.94	9.36	7.22	15.28	17.96	3.41	12.93	9.61	**7.77**
GIN	2.53	3.32	8.63	1.98	3.13	15.81	8.35	16.87	18.81	3.64	12.64	9.70	**8.78**
DCBG	5.90	7.26	21.0	1.35	1.91	11.20	10.5	17.2	22.2	5.34	11.9	14.9	**10.89**

considered the most representative of the quality of the results, will be used to make comparison easier.

4.1. Comparison of Disparity Maps

4.1.1. Accuracy of the Dense Disparity Maps. The GIF-based cost-aggregation method and the proposed hierarchical clustering method were first used to aggregate matching costs. Then winner-take-all and refinement operations were used to obtain the dense disparity maps. As shown in Figure 3, both methods yielded accurate results for the depth discontinuities as well as in the smooth regions for the test images.

The corresponding quantitative results are presented in Table 1, which records PBP in the nonoccluded, depth-discontinuous, and overall regions of the "Tsukuba," "Venus," "Cones," and "Teddy" images. The rightmost column of the table contains the average errors (AE), which were calculated using the average PBP over all twelve columns. As can be seen from the fourth and fifth rows of Table 1, the AE values obtained using the GIF (GIN) and the proposed method (HCN) without the refinement procedure were 8.78% and 7.77%, respectively. The first two rows show the errors obtained using the GIF (GIR) and the proposed method (HCR) with the refinement procedure; the AE values were 5.85% and 5.67%, respectively. This shows that the proposed method outperformed the GIF for filtering matching costs during cost aggregation. As expected, the proposed refinement method is suitable for removing mismatches, and the improvement is evident. In particular, as can be seen in Table 1, HCR can also outperform the original ASW algorithm [6] and the fast DCBG technique [20]. In the authors' opinion, the method proposed in this research may well achieve the topmost position among local stereo correspondence algorithms.

To verify algorithm stability, the performance of the GIF and the proposed methods was compared on an additional 27 Middlebury stereo images [27]. As described above, the PBP values with a disparity error larger than one pixel in all the regions were used to build the average of this measure over all 27 test images. The corresponding quantitative evaluation is summarized in Table 2. Note that both methods may be less accurate in large untextured regions such as the Midd1 and Monopoly pairs. Errors in untextured regions are due mostly to mismatches and will cause inconsistencies between the left

Table 2: Evaluation for stereo methods on all 27 Middlebury stereo pairs.

Method	Aloe	Baby1	Baby2	Baby3	Bowling1	Bowling2	Cloth1	Cloth2	Cloth3	Cloth4
HCN	12.71	11.14	11.81	17.87	26.70	19.10	9.71	16.37	11.15	14.95
GIN	13.42	12.39	12.88	17.98	27.37	19.19	10.36	16.56	11.30	15.40
HCR	8.19	4.99	7.24	9.74	20.69	14.38	4.61	10.96	5.34	10.66
GIR	8.78	5.41	7.59	9.99	20.26	14.61	5.03	11.43	5.43	10.81
Method	Flowerpots	Lampshade1	Lampshade2	Midd1	Midd2	Monopoly	Plastic	Rocks1	Rocks2	Wood1
HCN	23.60	23.03	30.95	45.66	41.66	36.51	43.62	11.90	12.24	16.78
GIN	23.41	24.13	32.64	46.58	42.90	34.78	47.81	11.72	11.83	17.61
HCR	18.48	15.86	23.46	43.95	37.30	22.71	35.60	5.72	5.19	5.26
GIR	18.81	16.67	24.01	44.35	38.62	25.01	38.33	5.55	5.02	5.57
Method	Wood2	Art	Books	Dolls	Laundry	Moebius	Reindeer	AE		
HCN	15.43	26.26	21.59	17.32	27.98	20.32	21.96	**21.79**		
GIN	14.83	26.41	21.10	16.68	29.19	20.06	23.12	**22.28**		
HCR	0.64	18.60	17.85	11.90	20.80	14.68	8.19	**14.92**		
GIR	0.57	18.59	17.50	11.96	22.80	14.22	8.36	**15.38**		

Table 3: Run time comparison of the GIF and the proposed method in seconds.

Version	Method	Tsukuba	Venus	Teddy	Cones
CPU	GIF	32	80	251	257
	HC	28	62	204	203
GPU	GIF	0.330	0.543	1.677	1.695
	HC	0.270	0.434	1.307	1.315

and right disparity maps. However, the proposed HC method is still the winner and slightly outperforms the GIF technique. In a comparison of HCN and HCR, the proposed refinement method is expected to perform well.

4.1.2. Computational Complexity. We have implemented two versions of the local matching filter described in this paper and tested them on the four benchmark images. These implementations include CPU versions written in MATLAB and a GPU version written in CUDA. The performance numbers reported in this paper were measured on a 2.99-GHz Intel Core 2 Duo processor with 3.25 GB of memory and on a GPU (GeForce 9500GT) with 512 MB of memory. Note that all of the algorithms were run on the same testing platform to achieve a fair comparison.

As demonstrated by the results shown in Table 3, the proposed method is slightly faster than GIF for the testing images both on CPU and GPU platforms. The reason for this is that the total complexity of GIF on three-dimensional color images for disparity maps is O(17N) [21], while that of the proposed method is O(15N), with a tree height $H = 4$ and therefore a constant $K = 2^H - 1 = 15$. Moreover, the proposed method also has the same linear time requirement as the GIF, regardless of the filter kernel size and the intensity range.

Obviously, all the run times increase with the dimensional size of the disparity maps, where the "Tsukuba," "Venus," "Cones," and "Teddy" disparity maps are $384 \times 288 \times 15$, $434 \times 383 \times 19$, $450 \times 375 \times 59$, and $450 \times 375 \times 59$, respectively. As a result, our CPU implementation processes a 1-megapixel image in about 16 to 20 seconds, resulting in a time-consuming process. Due to the simple and parallel operations used by our approach, our filter achieves significant performance gains on GPU platform. The total time required for filtering a 1-megapixel image ranges from 0.1 to 0.2 seconds. This represents a speedup from 80 to 200 compared to our CPU implementation.

Consequently, the proposed approach seems to perform slightly better than others in terms of accuracy and computational efficiency.

4.2. Influence of Parameter Settings

4.2.1. Robust Illumination-Independent Behavior. All of the stereo benchmark images used in Section 4.1 have been acquired under normal lighting conditions and there are no significant variations of luminosity between the two images of a stereo pair. However, this condition is often not valid for a real environment [29, 30]. Due to illumination effects, the color value is not always reliable for stereo matching. Therefore, it has been suggested to supplement the constraint on the gradient in (1), which is invariant to additive illumination changes.

In order to confirm that the proposed method is robust when applied to illumination-variant stereo pairs, PBP results of the altered Tsukuba images with different weight coefficient were presented in Table 4. Refer to Nalpantidis [29], each stereo pair consisted of the left image of the Tsukuba image set and a mount of different versions of the right image whose luminosity alteration ranged from −25% to +25% with 5% increments.

It can be seen from Table 4 that the algorithm only based on color value (as $\alpha = 1$) leads to many false matches with the lighting nonuniformity, while the quality of the algorithm that just relied on gradient (as $\alpha = 0$) remains almost the same for every tested lighting condition. Moreover the algorithm combining color with gradient value produces the best results for ideal lighting conditions ($L = 0\%$). As a result, the quality

Table 4: Evaluation on illumination-variant stereo pairs with different weight coefficient.

L	α														
	0			0.1			0.5			0.9			1		
	Non	All	Disc	Non	All	Disc	Non	All	Disc	Non	All	Disc	Non	All	Disc
−25%	3.33	4.24	12.27	3.86	4.82	13.50	15.7	16.8	27.2	40.9	41.6	48.1	67.0	67.3	69.9
−20%	3.12	4.00	11.96	3.47	4.36	12.80	17.0	18.1	27.9	40.8	41.5	46.5	60.2	60.6	62.6
−15%	2.97	3.84	11.65	3.27	4.10	11.86	18.8	19.7	29.0	44.8	45.3	46.6	55.4	55.8	55.4
−10%	3.03	3.93	11.87	3.43	4.22	11.23	18.0	18.7	25.3	41.4	41.8	41.9	44.7	45.0	43.7
−5%	3.08	4.07	12.24	3.25	4.01	10.50	12.5	13.3	19.2	25.1	25.7	27.9	26.4	27.0	29.0
0%	**3.21**	**4.21**	**12.36**	**2.14**	**2.94**	**9.16**	**1.95**	**2.75**	**9.18**	**2.19**	**3.09**	**9.63**	**2.24**	**3.15**	**9.73**
5%	3.33	4.32	12.47	2.25	3.09	9.79	10.9	11.7	16.3	21.3	21.9	23.5	22.4	23.0	24.2
10%	3.51	4.54	13.12	3.22	4.10	11.45	19.8	20.5	23.6	41.2	41.4	38.1	42.6	42.8	39.3
15%	3.85	4.88	13.74	3.88	4.82	13.03	20.5	21.3	26.3	49.5	49.9	46.0	52.3	52.6	48.8
20%	4.09	5.14	14.39	4.23	5.24	14.17	19.0	20.1	29.6	54.4	54.9	53.7	59.2	59.5	59.0
25%	4.54	5.62	15.32	4.67	5.76	15.37	20.0	21.1	31.6	53.1	53.7	54.3	62.0	62.5	62.6

Table 5: Run time and PBP of the disparity maps vary with respect to tree height increasing.

Height	Time (s)	Non	All	Disc
1	0.031	4.39	5.91	20.63
2	0.056	3.00	3.99	13.78
3	0.128	2.32	3.21	9.93
4	0.270	2.14	2.94	9.16
5	0.499	2.05	2.78	8.90
6	1.203	2.02	2.68	8.99

of our proposed method (when $\alpha = 0.1$) can be less affected by any difference of the lighting conditions and be satisfied with a suitable accuracy.

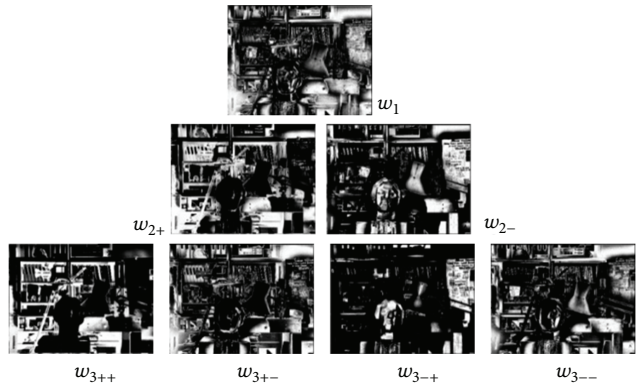


Figure 4: The first three levels of weight (16) distribution.

4.2.2. Selection of the Tree Height. The first step is to discuss how tree height affects the performance of the proposed method. "Tsukuba" was chosen as the test image, and the GPU run time and PBP of the disparity maps were recorded with increasing tree height, as shown in Table 5. Note that the spatial $\sigma_s = 11$ and the range $\sigma_r = 0.08$ are constants.

It is clear from the second column that the proposed algorithm will increase greatly in compilation time with increasing tree height. Because the number of sampling points $K = 2^H - 1$ increases with tree height, the greater number of summation operations (8) for the sampling points will be time consuming. On the contrary, the accuracy of the disparity maps for nonoccluded, depth-discontinuous, and overall regions, which is demonstrated in the last three columns, is dramatically improved with increasing height. The reason for this, as mentioned before, is that increasing the number of sampling points reduces the errors between the continuous integration (4) and the discrete summation (5). Figure 4 shows the first three levels of weights (16) for the test image corresponding to the sampling tree (Figure 1). Similar pixels with relatively large weights are shown in white, while black denotes dissimilar pixel areas with very small weights. Moving down this tree, the large weights will be gradually assigned to edge regions, and image integrity will be guaranteed. For example, the missing information from the edge regions of the lamp in w_{2-}, which is black with very small weights, can be compensated by more detail from the white regions with large weights in w_{3-+} and w_{3--}.

However, accuracies improve slightly or even become worse between $H = 5$ and $H = 6$ because the spatial and range parameters are constant and unsuitable for the tree height. To confirm this cause, $H = 6$ is kept constant, and the PBP for "non" are improved, with values of 1.98 and 1.95 when $\sigma_s = 11$, $\sigma_r = 0.09$ and $\sigma_s = 12$, $\sigma_r = 0.08$, respectively. Therefore, the spatial and range parameters also affect performance, which will be further discussed below.

4.2.3. Influences of σ_s and σ_r. σ_s and σ_r are two standard deviations used to adjust the spatial similarity and the range similarity, respectively. The spatial spread σ_s is chosen based on the desired amount of low-pass filtering. A large σ_s creates more blurring, meaning that more high-frequency components are removed and the image becomes obviously blurred. Similarly, the range spread σ_r is set to achieve the desired amount of combination of pixel range values. Generally speaking, pixels with range differences less than σ_r

Table 6: PSNR with different parameters σ_s and σ_r.

σ_r	σ_s									
	1	10	20	30	40	50	60	70	80	90
0.01	11.63	12.99	13.10	13.21	13.24	13.28	13.46	13.38	13.08	12.73
0.1	12.71	***14.01***	***14.06***	***14.07***	***14.01***	***14.00***	***14.00***	***13.86***	***13.86***	***13.87***
0.2	12.87	***13.99***	13.99	13.97	13.89	13.85	13.83	13.67	13.64	13.62
0.3	12.93	***13.96***	13.93	13.88	13.77	13.70	13.65	13.47	13.43	13.39
0.4	12.96	***13.94***	13.88	13.81	13.70	13.62	13.56	13.38	13.33	13.29
0.5	12.98	***13.93***	13.85	13.77	13.64	13.56	13.50	13.32	13.27	13.23
0.6	12.99	***13.92***	13.83	13.73	13.60	13.52	13.45	13.27	13.22	13.18
0.7	12.99	***13.91***	13.81	13.71	13.57	13.49	13.42	13.24	13.19	13.15
0.8	13.00	***13.91***	13.80	13.69	13.55	13.46	13.39	13.21	13.16	13.12
0.9	13.00	***13.91***	13.79	13.68	13.53	13.44	13.37	13.19	13.13	13.09

are mixed together, and those with differences greater than σ_r are removed [13].

The results obtained from varying σ_s, σ_r in (8) are equivalent to adjusting the spatial and range spread for a bilateral filter. However, the influence of changes in σ_s, σ_r on the clustering weights (16) is also significant. To analyze the error source qualitatively, the following two propositions are defined.

Proposition 1. *More sampling points will be needed for good accuracy when the range spread σ_r is small or the spatial spread σ_s is large. If the height is constant, the matching error would suffer from the edge-losing effect (ELE).*

Proof. Using (16), the weight of each pixel is reduced when the value of σ_r is small with respect to the overall range of values in the image or when the set of sampling points m is dissimilar to the image value I because m appears to be hazy due to larger σ_s (9) or (15). Moreover, each m covers a limited sampling region, which means that in turn, more m values are needed to adapt to the signal [7]. □

Proposition 2. *The filter weights* (6) *in the proposed method behave more like a low-pass filter when the range spread σ_r is large or the spatial spread σ_s is small. The matching error would be caused by the edge-smoothing effect (ESE).*

Proof. Using (16), the weights of all pixels are increased when the value of σ_r is large with respect to the overall range of values in the image or when the set of sampling points m is similar to the image value I because m appears to be less hazy due to smaller σ_s (9) or (15). Therefore, all pixel values in any given neighborhood have approximately the same weight from range filtering for (6), and the resulting filter approximates a standard Gaussian filter [13]. □

"Tsukuba" was chosen as the test image. A fast way to determine the best choice of σ_s and σ_r using the filtering results of peak signal-to-noise ratios (PSNR) [31] is

$$\mathrm{PSNR} = 10\log_{10}\left\{\frac{255^2 MN}{\sum_{x,y\in(M,N)}\left[f(x,y)-g(x,y)\right]^2}\right\}, \tag{21}$$

where $M \times N$ is the image size, $f(\cdot)$ is the local filtering result (as in Figures 3(a)–3(d)), and $g(\cdot)$ is the ground truth (as in Figure 3(e)). Table 6 shows the PSNR distributions with $\sigma_s \in (1, 90)$, $\sigma_r \in (0.01, 0.9)$. The following can be determined.

(1) The PSNR decreases as σ_s or σ_r becomes smaller when $\sigma_s \in (1, 10)$ and $\sigma_r \in (0.01, 0.1)$. The reason for this is that ESE obeys Proposition 2, that the proposed method behaves more like a low-pass filter when σ_s decreases. The reason for the latter is that ELE obeys Proposition 1 that accuracy is reduced due to lack of more information in the filtering results around the edge regions due to a limited number of sampling points.

(2) The PSNR decreases with increasing σ_s or σ_r when $\sigma_s \in (10, 90)$ and $\sigma_r \in (0.1, 0.9)$. It obeys Propositions 1 and 2 that the accuracy is reduced due to ELE with large σ_s in a constant-height tree and due to ESE with large σ_r for each sampling point.

From the two findings, it can be confirmed that the optimal values for σ_s and σ_r are approximately 10 and 0.1, respectively, which are shown using bold italic font in Table 6.

The PBP distributions for the "non," "all," and "disc" disparity maps were then recorded with $H = 4$, but with σ_s and σ_r varying according to $\sigma_s \in (1, 20)$, $\sigma_r \in (0.01, 0.2)$, as shown in Figure 5. Results derived from Figure 5 can be summarized as follows.

(1) All the PBP perform like the results of PSNR; the PBP values increase as σ_s or σ_r becomes smaller. They decrease with increasing σ_s or σ_r, but only up to a certain point, which constitutes the best parameter setting. After that point, the PBP values will gradually increase.

(2) Figure 5(c) is more obviously different from the first two PBP because it was calculated only from the edge regions. The accuracy reduction refers to the nonoccluded and overall regions generated by ESE or ELE, which are smaller than the depth-discontinuous regions.

Consequently, accuracy was reduced when σ_s and σ_r became too small or too large within a constant-height tree.

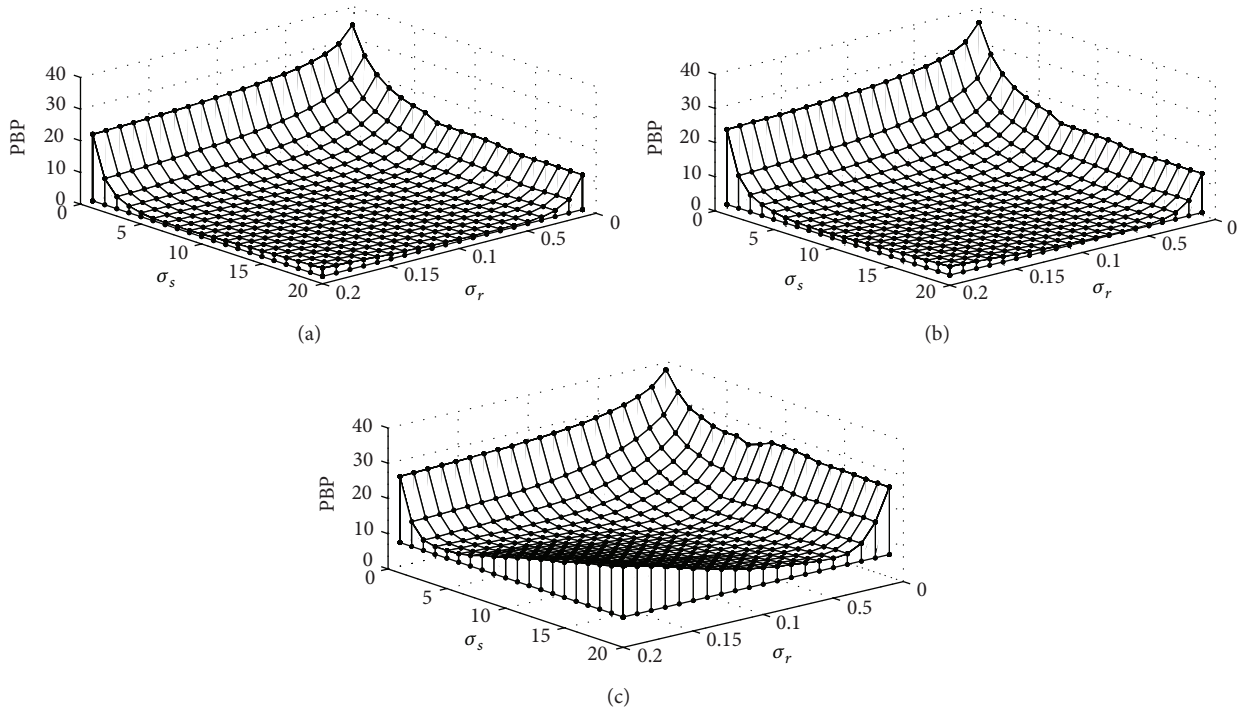


Figure 5: PBP for the (a) "non," (b) "all," and (c) "disc" disparity maps with various σ_s and σ_r.

In terms of computational cost, the range component depends linearly $O(N)$ on the image, regardless of the filter kernel for each sampling layer. To this end, the authors suggest that the tree height be first determined according to the time consumption and then that the filtering results for PSNR be used to determine the general choice of σ_s and σ_r.

5. Conclusions and Future Work

In this paper, a new local solution for fast, high-quality dense stereo correspondence has been proposed that focuses on matching cost filtering method which is based on a high-performance hierarchical clustering algorithm. Instead of filtering the matching costs using an edge-preserving smoothing operator as in the popular bilateral filter, the cost aggregation model was adjusted to compute the matching responses for all image pixels at a set of sampling points generated using a clustering method. The computational complexity for this filtering is linear both in the number of image pixels and the number of clustering classes. The experimental results of the comparison have demonstrated that the proposed method outperforms the GIF-based matching algorithm, which is one of the best local methods on the Middlebury benchmark in terms of both speed and accuracy. Moreover, the results of performance tests, which provide effective guidelines for parameter selection, indicate that good accuracy is highly dependent on the weight coefficient, the height of the hierarchical binary tree, and the spatial and range standard deviations. As a result, it can now be confirmed that the proposed approach can be capable of high-speed processing and offer high-quality disparity maps for dense stereo correspondence.

In the experimental results, we show that both of the GI and HC filtering methods make some of the erroneous disparity values due to the lack of texture, which is a traditional challenge for stereo algorithms. The reason is that a pixel's disparity value is obtained by selecting the point of highest matching score and independently of disparity assignments of neighboring pixels. Hence, most of the disparity values in the low-texture areas maybe incorrect using a local matching method. To overcome this bottleneck, the authors plan to make the algorithm capable of handling large untextured regions, which remains an active area for future research [32].

Acknowledgments

This work was supported by the open project of Beijing Key Laboratory on Measurement and Control of Mechanical and Electrical System (no. KF20121123206), Key Laboratory of Modern Measurement and Control Technology (BISTU), Ministry of Education, Funding Project for Academic Human Resources Development Institutions of Higher Learning under the Jurisdiction of Beijing Municipality (no. PHR201106130), and Funding Project of Beijing Municipal Science & Technology Commission (no. Z121100001612011).

References

[1] S. Y. Chen, H. Tong, and C. Cattani, "Markov models for image labeling," *Mathematical Problems in Engineering*, vol. 2012, Article ID 814356, 18 pages, 2012.

[2] J. A. Kalomiros, "Dense disparity features for fast stereo vision," *Journal of Electronic Imaging*, vol. 21, no. 4, Article ID 043023, 2012.

[3] S. Park and H. Jeong, "High-speed parallel very large scale integration architecture for global stereo matching," *Journal of Electronic Imaging*, vol. 17, no. 1, Article ID 010501, 2008.

[4] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: from software to hardware," *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[5] M. Gong, R. Yang, L. Wang, and M. Gong, "A performance study on different cost aggregation approaches used in real-time stereo matching," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 283–296, 2007.

[6] 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.

[7] E. S. L. Gastal and M. M. Oliveira, "Adaptive manifolds for real-time high-dimensional filtering," *ACM Transactions on Graphics*, vol. 31, no. 4, 2012.

[8] C. Rhemann, A. Hosni, M. Bleyer, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2012.

[9] T. Kanade and M. Okutomi, "Stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.

[10] K.-H. Bae, J.-J. Kim, and E.-S. Kim, "New disparity estimation scheme based on adaptive matching windows for intermediate view reconstruction," *Optical Engineering*, vol. 42, no. 6, pp. 1778–1786, 2003.

[11] S. A. Adhyapak, N. Kehtarnavaz, and M. Nadin, "Stereo matching via selective multiple windows," *Journal of Electronic Imaging*, vol. 16, no. 1, Article ID 013012, 2007.

[12] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[13] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE 6th International Conference on Computer Vision*, pp. 839–846, January 1998.

[14] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 60–65, June 2005.

[15] Y. S. Heo, K. M. Lee, and S. U. Lee, "Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, June 2007.

[16] F. Porikli, "Constant time O(1) bilateral filtering," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[17] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time O(1) bilateral filtering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 557–564, Miami, Fla, USA, June 2009.

[18] M.-H. Ju and H.-B. Kang, "Constant time stereo matching," in *Proceedings of the 13th International Machine Vision and Image Processing Conference (IMVIP '09)*, pp. 13–17, Dublin, Republic of Ireland, September 2009.

[19] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Transactions on Graphics*, vol. 26, no. 3, Article ID 1276506, pp. 103-1–103-9, 2007.

[20] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, vol. 6313 of *Lecture Notes in Computer Science*, pp. 510–523, Springer, Heraklion, Greece, 2010.

[21] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, vol. 6311 of *Lecture Notes in Computer Science*, pp. 1–14, Springer, Heraklion, Greece, 2010.

[22] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza, "Linear stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1708–1715, Barcelona, Spain, November 2011.

[23] 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, 2012.

[24] 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.

[25] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, Minn, USA, June 2007.

[26] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[27] "Middlebury stereo vision database," http://vision.middlebury.edu/stereo/.

[28] K. Mühlmann, D. Maier, J. Hesser, and R. Männer, "Calculating dense disparity maps from color stereo images, an efficient implementation," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 79–88, 2002.

[29] L. Nalpantidis and A. Gasteratos, "Stereo vision for robotic applications in the presence of non-ideal lighting conditions," *Image and Vision Computing*, vol. 28, no. 6, pp. 940–951, 2010.

[30] L. Nalpantidis and A. Gasteratos, "Biologically and psychophysically inspired adaptive support weights algorithm for stereo correspondence," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 457–464, 2010.

[31] K. Bae, J. Ko, and J. Lee, "Stereo image reconstruction using regularized adaptive disparity estimation," *Journal of Electronic Imaging*, vol. 16, no. 1, Article ID 013013, 2007.

[32] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 492–504, 2009.