Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Accurate real-time stereo correspondence using intra- and inter-scanline optimization*

Li YAO[†1,2], Dong-xiao LI[†‡1,2], Jing ZHANG[1,2], Liang-hao WANG[1,2], Ming ZHANG[1,2]

([1]*Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China*)

([2]*Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China*)

[†]E-mail: zjyaoli@163.com; lidx@zju.edu.cn

Received Oct. 23, 2011; Revision accepted Feb. 2, 2012; Crosschecked May 4, 2012

Abstract: This paper deals with a novel stereo algorithm that can generate accurate dense disparity maps in real time. The algorithm employs an effective cross-based variable support aggregation strategy within a scanline optimization framework. Rather than matching intensities directly, the use of adaptive support aggregation allows for precisely handling the weak textured regions as well as depth discontinuities. To improve the disparity results with global reasoning, we reformulate the energy function on a tree structure over the whole 2D image area, as opposed to dynamic programming of individual scanlines. By applying both intra- and inter-scanline optimizations, the algorithm reduces the typical 'streaking' artifact while maintaining high computational efficiency. The experimental results are evaluated on the Middlebury stereo dataset, showing that our approach is among the best for all real-time approaches. We implement the algorithm on a commodity graphics card with CUDA architecture, running at about 35 fames/s for a typical stereo pair with a resolution of 384×288 and 16 disparity levels.

Key words: Stereo correspondence, Scanline optimization, Real-time
doi:10.1631/jzus.C1100311 **Document code:** A **CLC number:** TN911.73

1 Introduction

Stereo correspondence is one of the most widely studied and fundamental topics in computer vision. It refers to the process of reconstructing a 3D model of a scene from two or more images taken from distinct viewpoints. In the last few years, it has received much attention partly due to a number of applications that require 3D depth information for good reconstruction, such as robot navigation, object recognition, realistic scene visualization, and depth image based rendering. However, stereo correspondence is an ill-posed problem with inherent ambiguities produced by projective and photometric distortion, sensor noise, lack of texture, and occlusions. To reduce ambiguities and uncertainties of matching, a variety of constraints and assumptions are exploited in most algorithms. Local approaches determine the disparity of a concerned pixel depending on a finite surrounding area, by making the implicit assumption that all pixels in a support window are from similar depth in a scene. In contrast, global approaches make an explicit piecewise smoothness assumption and solve the optimization problem in the framework of Markov random fields. While 2D optimization turns out an NP-hard problem (Veksler, 1999), the minimization strategy based on dynamic programming (DP) or scanline optimization (SO) techniques can find the global minimum for independent scanlines.

Comprehensive taxonomy and categorization on stereo correspondence algorithms have been reviewed by Scharstein and Szeliski (2002). Currently the standard method of testing these algorithms is to run

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 60802013 and 61072081), the National Science and Technology Major Project of the Ministry of Science and Technology of China (No. 2009ZX01033-001-007), and the China Postdoctoral Science Foundation (No. 20110491804)

them on the benchmark of the Middlebury stereo database (Scharstein and Szeliski, 2001). To date, almost all of the top ranked algorithms have relied on some form of global optimization or have been aided by image segmentation. Unfortunately, the accuracy of the results can be achieved only with higher computational complexity, making the algorithms slower and limited to off-line applications. In general, only local approaches and scanline based optimization methods are suitable for some real-time systems like three-dimensional TV (3DTV), in which high quality depth information at video rate is critical. However, most local methods are prone to mismatch at discontinuities and lead to blurred object boundaries. Scanline based optimization methods, while being efficient, suffer from the well-known horizontal 'streaking' effect due to non-enforcement of the inter-scanline consistency.

Aiming at achieving reasonably accurate disparity maps within a short time, we propose a hybrid scanline based optimization method. Compared with the existing real-time methods, the proposed method contains some distinguishing features. First, the raw matching cost of stereo image pair is aggregated over cross-based adaptive windows, while the traditional global optimization methods often skip the cost aggregation step. The motivation behind it is that the raw matching cost is very sensitive to local ambiguous regions in images. And, the subsequent global optimization is highly dependent on the initial matching result. Second, to reduce the 'streaking' artifacts, the SO-based algorithm is performed in both horizontal and vertical directions. Each independent optimization component is formulated on a tree structure, so that the scanline optimization framework can execute effectively and guarantee to find a unique optimal solution. Moreover, by using the cross check technique, the disparity accuracy in discontinuity regions is further improved. Since most stages in the proposed method can be effectively executed in parallel, we take advantage of high performance capabilities of graphics hardware to accelerate the algorithm. Experimental results show that the proposed approach is able to generate accurate dense disparity maps and it is among the best performing real-time or near real-time methods according to the evaluation on the Middlebury datasets.

2 Related work

In this section, we briefly review the previous stereo algorithms that are feasible for real-time applications. A few years ago, due to the low capability of general computers, real-time performance could be achieved only with special hardware designs (Konolige, 1997; Woodfill and Herzen, 1997; Darabiha *et al*., 2003). With advances in CPU processing power, software based near real-time systems began to emerge (Mulligan *et al*., 2002). Local or correlation window based methods are preferred for fast implementation. The simplest one is using a fixed rectangular window to aggregate the matching cost, and then a winner-take-all (WTA) strategy is performed to determine the disparities. Integral images (Crow, 1984) and the box-filtering (McDonnell, 1981) technique can be utilized for optimization and time acceleration. However, fixed window methods have a common limitation in that they do not explicitly deal with the depth discontinuities. To improve the accuracy, one of the first algorithms exploiting the idea of using a set of windows is 'shiftable windows' (Bobick and Intille, 1999), in which the support window is not constrained to be centered on the central position. The variable window approach (Veksler, 2003) varies the size of the window rather than its displacement, while Hirschmüller *et al*. (2002) used multiple windows and chose the most suitable one. Instead of shaping the support window, Yoon and Kweon (2006) adjusted the support-weights of the pixels in a fixed support window by color similarity and geometric proximity. It outperforms other local methods but is quite time-consuming. Tombari *et al*. (2008a) surveyed and evaluated the widely used local aggregation strategies on both accuracy and computational requirements.

Dynamic programming (DP) is an alternative approach often adopted in real-time applications for its high efficiency. DP works by computing the minimum-cost path through the matrix of all matching costs between two corresponding scanlines, i.e., through a horizontal slice of the disparity space image (DSI) (Scharstein and Szeliski, 2002). As the smoothness constraint is enforced in a global optimization framework along each scanline, DP can achieve more reliable performance at depth borders and uniform regions in comparison with the local methods using a greedy strategy (Veksler, 2005;

Wang *et al*., 2006). To overcome the typical 'streaking' problem in DP, related approaches are focusing on the improvements of inter-scanline consistency. For example, Kim *et al*. (2005) proposed a two-pass DP method, in which optimization is performed across the scanline in a similar way. Criminisi *et al*. (2007) proposed a four-state DP algorithm to disambiguate between horizontal/vertical matched moves and occlusion events. Sara (2010) described a variant on DP to refine the process of searching for the min-cost path, which is called '3-label DP'. In Veksler (2005), DP was applied to a tree structure with most important edges, rather than individual scanlines. The best ranked DP based method up to now may be the region-tree based DP (Lei *et al*., 2006). Image segmentation is used to construct a minimum spanning tree for global DP optimization. However, color segmentation of the image is still hard and unsuitable for parallel implementation.

Recently, there have been an increasing number of stereo matching approaches that use the potential parallel processing power of the graphics processing unit (GPU) to decrease the running time. It was first explored in a local method (Yang *et al*., 2002), then widely applied in DP based methods (Gong and Yang, 2005; 2007; Wang *et al*., 2006). Nowadays, even belief propagation (BP) algorithms are implemented on GPU for its high matching accuracy (Yang *et al*., 2006). But BP algorithms inherently consume large memory space and require substantial iterations to converge.

3 The proposed approach

Given a stereo image pair, the goal of stereo correspondence is to infer the disparity map by finding the corresponding points that emanate from the same position in the 3D space. According to the epipolar geometry, correspondence for a point belongs to an epipolar line (Hartley and Zisserman, 2004). To facilitate the matching problem, it is assumed that two input images are rectified so that an epipolar line becomes horizontal and the search range is limited to the horizontal direction. Subsequently, the relationship between two matching pixels can be described as

$$x' = x - d(x, y),\ \ y' = y, \tag{1}$$

where (x, y) and (x', y') are the pixel coordinates of the left and right images respectively, and $d(x, y)$ is the disparity to be determined.

Following the taxonomy proposed by Scharstein and Szelikski (2002), our approach can be partitioned into four main steps: matching cost computation, cost aggregation, optimization, and refinement (Fig. 1). The key idea behind our approach is utilizing the cost aggregation that is usually adopted in local methods to achieve a more reliable initial disparity cost image. A robust data energy term is then constructed based on the aggregated cost. By solving the global energy minimization problem, our two-pass scanline based optimization is able to improve the performance and reduce 'streaking' artifacts, while maintaining a relatively low computational complexity. Finally we use the symmetric disparity maps to refine the results. A more detailed description is given in the following sections.

3.1 Matching cost computation and aggregation

The first component of the proposed approach is to compare the similarity of pixels at corresponding locations. Here a widely used pixel-based matching measure of truncated absolute differences (TAD) is introduced as follows:

$$C_{\text{raw}}(x, y, d) = \min(\left|I_{\text{L}}(x, y) - I_{\text{R}}(x - d, y)\right|,\ \text{Trunc}),$$
$$d \in [d_{\min},\ d_{\max}], \tag{2}$$

where $I_{\text{L}}(x, y)$ and $I_{\text{R}}(x-d, y)$ are the intensities of pixel (x, y) in the left image and $(x-d, y)$ in the right image, respectively. Disparity d is limited between

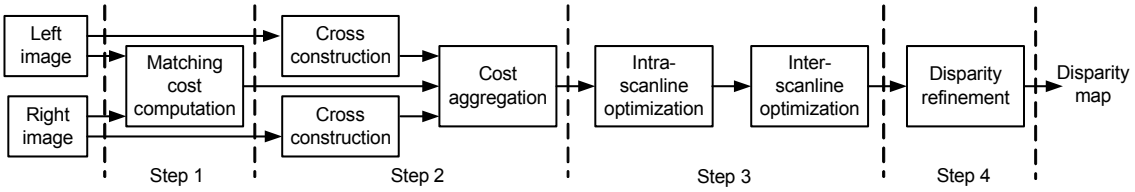

Fig. 1 Schematic overview of the proposed stereo correspondence algorithm

possible minimum ($d_{\min}$) and maximum ($d_{\max}$) disparities. The matching cost $C_{\text{raw}}(x, y, d)$ is stored in a 3D matrix known as the disparity space image (DSI).

As mentioned earlier, the raw DSI is always accompanied with noise. A typical aggregation step allows for the summation or averaging of the DSI over a support region. It implicitly assumes that the support region is a frontal-parallel surface and all pixels share similar disparities. To obtain accurate results at near depth discontinuities, an appropriate support window should be selected adaptively. We adopt a cross-based aggregation strategy proposed by Zhang *et al*. (2009), mainly for two reasons: first, the support window can vary adaptively with arbitrary size and shape according to the color similarity; second, the aggregation over irregularly shaped support windows can be computed rapidly by employing the integral image technique.

In the first stage of the cross-based aggregation algorithm, an upright cross is established for each pixel $p=(x_p, y_p)$ in the left image. It consists of two orthogonal line segments (Fig. 2a). The horizontal segment $H(p)$ of pixel p is decided by its left arm h_p^- and right arm h_p^+, and the vertical segment $V(p)$ is decided by the up arm v_p^- and bottom arm v_p^+. The length of each arm is determined by searching for the extreme point in that direction based on the color similarity. Take h_p^- as an example. The leftmost point p_0 of $H(p)$ is decided as follows:

$$\begin{cases} \max\limits_{c\in\{r,g,b\}}(|I_c(p_i)-I_c(p)|)\le\tau,\ \forall x_{p_i}\in[x_{p_0},x_p],\ y_{p_i}=y_p, \\ \max\limits_{c\in\{r,g,b\}}(|I_c(p^*)-I_c(p)|)>\tau,\ x_{p^*}=x_{p_0}-1,\ y_{p^*}=y_p, \end{cases} \tag{3}$$

where I_c is the intensity of color band c, and τ is the color similarity threshold. Then the arm length is simply computed as $h_p^-=\max(x_p-x_{p_0}, 1)$, which implies that a minimum support region of 3×3 window is enforced.

Once the four arms of each pixel are computed, the final support region for p is defined as a union of horizontal segment $H(q)$, in which q traverses over the vertical segment of p (Fig. 2b):

$$U(p)=\bigcup_{q\in V(p)}H(q). \tag{4}$$

The second stage is to aggregate the cost of the concerned central pixel over the support region which was previously located. To avoid distorting the outlier in the reference image, a symmetric support region is adopted. This means when aggregating cost for p over $U(p)$ with fixed disparity d in the DSI slice, the corresponding pixels should also belong to the support region in the reference image. The arm length of p in the left image is updated as follows:

$$l_p\leftarrow\min(l_p, l'_{p'}),\ \ p'=(x_p-d, y_p), \tag{5}$$

where l_p denotes the element in set $\{h_p^-, h_p^+, v_p^-, v_p^+\}$, and $l_{p'}$ denotes the corresponding arm length of the central pixel (x_p-d, y_p) in the right image.

Then the aggregation cost for p over the updated support region $U(p)$ is computed as

$$C(x_p, y_p, d)=\frac{1}{|U(p)|}\sum_{(x_i,y_i)\in U(p)}C_{\text{raw}}(x_i, y_i, d), \tag{6}$$

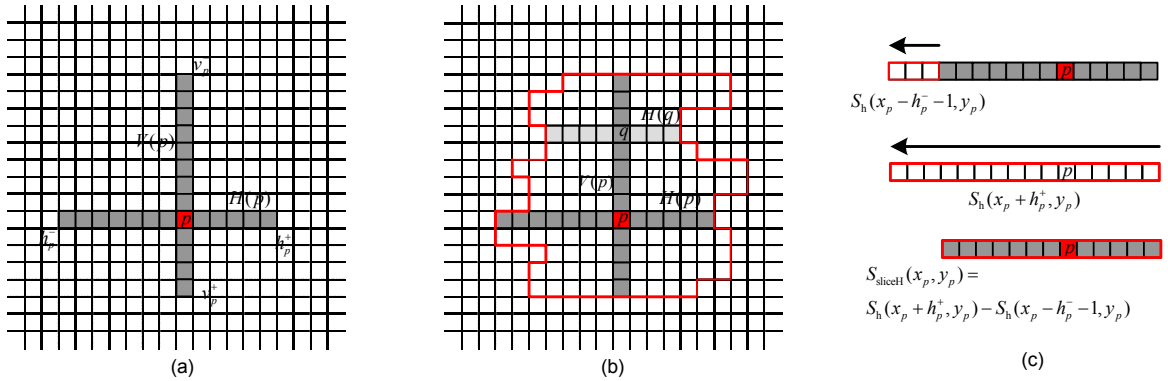

Fig. 2 Graphic illustration of the cross-based aggregation algorithm

(a) gives an upright cross construction. The upright cross consists of a horizontal segment $H(p)=\{(x, y)|x\in[x_p-h_p^-, x_p+h_p^+], y=y_p\}$ and a vertical segment $V(p)=\{(x, y)|x=x_p, y\in[y_p-v_p^-, y_p+v_p^+]\}$. In (b), the support region $U(p)$ is a combination of each horizontal segment $H(q)$ where q traverses over the vertical segment $V(p)$ of p. (c) is a schematic of a 1D integral image technique

where $|U(p)|$ is the number of pixels in support region $U(p)$.

Directly computing Eq. (6) over an irregular region is quite complicated, so we take advantage of the integral image technique (Crow, 1984) which facilities rapid computation. In the horizontal direction, the 1D integral image is defined in the form of a recursive function:

$$S_{\mathrm{h}}(x,y)=S_{\mathrm{h}}(x-1,y)+C_{\mathrm{raw}}(x,y,d), \tag{7}$$

where $S_{\mathrm{h}}(x, y)$ denotes the cumulative row sum of cost and $S_{\mathrm{h}}(-1, y)=0$. Subsequently, for each pixel $q=(x_q, y_q)$, the sum of aggregation cost over each horizontal segment $H(q)$ can be easily obtained using the integral image:

$$S_{\mathrm{segH}}(x_q,y_q)=S_{\mathrm{h}}(x_q+h_q^+,y_q)-S_{\mathrm{h}}(x_q-h_q^--1,y_q). \tag{8}$$

Define the vertical integral image as

$$S_{\mathrm{v}}(x,y)=S_{\mathrm{v}}(x,y-1)+S_{\mathrm{segH}}(x,y), \tag{9}$$

where $S_{\mathrm{v}}(x, -1)=0$. Based on the vertical integral image, the final aggregation cost defined in Eq. (6) can be computed as

$$\begin{aligned}C(x_p,y_p,d)=\frac{1}{|U(p)|}\big(&S_{\mathrm{v}}(x_p,y_p+v_p^+)\\&-S_{\mathrm{v}}(x_p,y_p-v_p^--1)\big).\end{aligned} \tag{10}$$

3.2 Optimization along the scanlines

The first step of global optimization aims to minimize a typical global energy function of independent scanlines, which is defined as

$$E_{\mathrm{H}}(d)=E_{\mathrm{d}}(d)+E_{\mathrm{s}}(d). \tag{11}$$

The data term $E_{\mathrm{d}}(d)$ measures how well the disparity function d agrees with the input image pair. Using previous aggregated DSI, the data energy is defined as

$$E_{\mathrm{d}}(d)=\sum_{p\in P}D_p(d_p)=\sum_{p\in P}C(x_p,y_p,d_p), \tag{12}$$

where P is the set of pixels along the scanline.

The smoothness term $E_{\mathrm{s}}(d)$ indicates that pixels in neighboring areas should have similar disparity values. For each scanline, the smoothness energy is measured only between horizontal neighboring pixels' disparities:

$$E_{\mathrm{s}}(d)=\sum_{(p,q)\in N}V(d_p,d_q), \tag{13}$$

where N is the set of interaction pairs of pixels, and V is an increasing function of disparity difference. A truncated multi-pass jump smoothness cost function is defined as

$$V(d_p,d_q)=\lambda(p)\cdot\min(\left|d_p-d_q\right|,T), \tag{14}$$

where $\lambda(p)$ controls the rate of increase in the cost, and T is the truncation value.

The coefficient $\lambda(p)$ in Eq. (14) is usually set as a constant, which assumes disparity d smooth everywhere. It leads to undesirable over-smoothed results at discontinuities where abrupt changes occur. It is necessary to apply the smoothness constraint while preserving discontinuities. Therefore, we control the smoothness penalty weight depending on local intensity difference. A shorter arm length in the horizontal cross denotes a local higher-texture area, so the smoothness cost should be decreased:

$$\lambda(p)=\begin{cases}\lambda/4, & \text{if } h_p^++h_p^-<\eta,\\ \lambda, & \text{otherwise,}\end{cases} \tag{15}$$

where η and λ are preset constants.

Once the global energy on a scanline has been defined, a variety of algorithms can be used to find the unique minimal solution of the disparity function. Traditional DP algorithms solve the problem by computing the minimum-cost path through a horizontal slice of DSI. Instead of directly finding the solution, we focus on obtaining an optimized cost vector, with each component being proportional to the probability of being an optimal solution at that pixel. It is defined as the minimum global energy by solving Eq. (11) for each pixel $t=(x_t, y_t)$ over $d_i\in[d_{\min}, d_{\max}]$:

$$\begin{aligned}&E_{\mathrm{opt}}(x_t,y_t,d_i)=\min_d\left(\sum_{p\in P}D_p(d_p)+\sum_{(p,q)\in N}V(d_p,d_q)\right)\\&\text{subject to } d\in[d_{\min},d_{\max}]^{|P|}\wedge d_t=d_i.\end{aligned} \tag{16}$$

We use the effective min-sum message passing algorithm (Moallemi and van Roy, 2010) to optimize the problem in form of Eq. (16) (Fig. 3a). A message is introduced as a vector of dimension given by the number of possible disparities (Weiss and Freeman, 2001). It is initialized to zero and combined to compute new outgoing message for each neighbor. Let $ml_t(d_t)$ be the message that node t receives from its left neighboring node s on disparity d_t. It is updated in the following way:

$$ml_t(d_t)=\min_{d_s}\left(V(d_t,d_s)+D_s(d_s)+ml_s(d_s)\right). \quad (17)$$

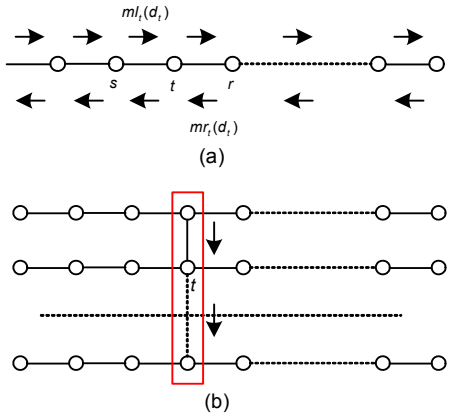

Fig. 3 Graphic illustration of the intra- and inter-scanline optimizations

(a) gives an optimization along each scanline using the min-sum massage passing algorithm. In (b), the disparity of each vertical line is determined by applying dynamic programming on a tree

The message is passed from the leftmost pixel towards the right side until it reaches the end. Similarly, the node t receives the message $mr_t(d_t)$ from its right neighboring node r and passes it in the reverse direction.

$$mr_t(d_t)=\min_{d_r}\left(V(d_t,d_r)+D_r(d_r)+mr_r(d_r)\right). \quad (18)$$

Expanding $ml_t(d_t)$ and $mr_t(d_t)$ by Eqs. (17) and (18) recursively, it is not difficult to show that the minimum of global energy defined in Eq. (16) is exactly given by

$$E_{\text{opt}}(x_t,y_t,d_i)=D_t(d_i)+ml_t(d_i)+mr_t(d_i). \quad (19)$$

3.3 Optimization across the scanlines

To enforce coherence between the scanlines, we additionally define a global energy in the vertical line. It has a similar form to Eq. (11), differing with the data energy term:

$$E_V(d)=\sum_{p\in P}E_{\text{opt}}(x_p,y_p,d_p)+\sum_{(p,q)\in N}V(d_p,d_q). \quad (20)$$

The weight function $\lambda(p)$ in $V(d_p, d_q)$ of Eq. (20) is defined as

$$\lambda(p)=\begin{cases}\lambda/4, & \text{if } v_p^+ + v_p^- < \eta,\\ \lambda, & \text{otherwise,}\end{cases} \quad (21)$$

where all parameters are identical to those of Eq. (15) except that the local intensity similarity is measured by the vertical arms.

When involving the data term with Eq. (19), the optimization process across the scanlines is equivalent to finding the minimal path of the vertical line on a tree in which the horizontal energy has been optimized before (Fig. 3b). We define a simple recursive function to optimize the global energy as follows:

$$M(x,y,d)=E_{\text{opt}}(x,y,d)+\min_{d'}\left(M(x,y-1,d')+V(d,d')\right), \quad (22)$$

where $M(x, y, d)$ is the accumulated energy at pixel (x, y) and all pixels in the up-side have already been optimized. Meanwhile, each pixel stores the optimal path which points to the previous pixel:

$$P(x,y,d)=\arg\min_{d'\in[d_{\min},d_{\max}]}\left(M(x,y-1,d')+V(d,d')\right). \quad (23)$$

Once the recursive procedure reaches the end of each vertical line, the minimum of $M(x, y, d)$ exactly equals the minimum of the global energy in Eq. (20). We select the d which minimizes $M(x, y, d)$ as the optimal disparity at pixel (x, y). Tracking back the optimal path, we can easily assign the optimal disparity value for all pixels along the vertical line:

$$d(x,y-1)=P(x,y,d(x,y)). \quad (24)$$

3.4 Refinement

The goal of this step is to improve the accuracy at disparity discontinuities. A symmetric disparity consistency checking technique is used for detecting outliers that are declared as half-occlusions. Half-occlusions often occur in a disparity discontinuity region which is invisible in either the left image or the right image. There are two typical cases of disparity discontinuities. Fig. 4a shows a positive disparity jump where background region A is partly occluded by foreground object B in the right image. Therefore, the pixels in occlusion regions have no actual correspondence, resulting in poor disparity accuracy without an explicit occlusion model. Conversely, Fig. 4b describes a case of negative disparity jump where the disparity of A is greater than that of B. In this case, the regions of both A and B are visible in the right image. However, the disparities near a discontinuity are likely to be affected by the neighbors due to the smoothness constraint enforced by the global optimization framework.

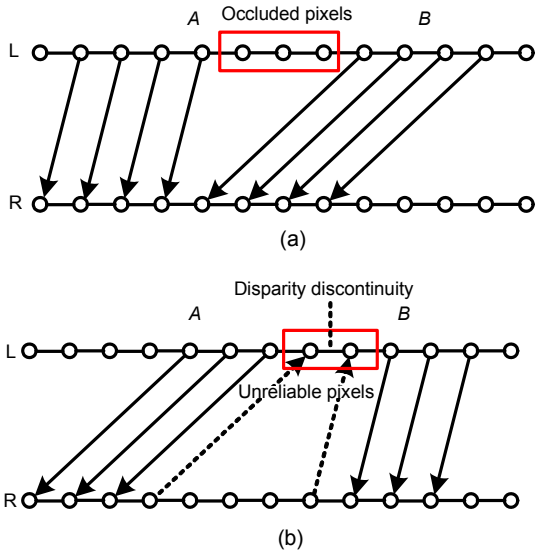

Fig. 4 Two cases of disparity discontinuities
(a) Positive jump, $D(A)<D(B)$, where $D(A)$ denotes the disparities in region A; (b) Negative jump, $D(A)>D(B)$

To implement the post-processing of the disparity map, bidirectional matching is performed according to the above procedure. The pixels whose disparity values are not consistent with the two maps are classified as outliers:

$$\left|d_{\mathrm{LR}}(x,y)-d_{\mathrm{RL}}(x-d_{\mathrm{LR}}(x,y),y)\right|>0, \qquad (25)$$

where $d_{\mathrm{LR}}(x, y)$ denotes the left disparity map and $d_{\mathrm{RL}}(x, y)$ denotes the right disparity map.

Outliers that occur near negative disparity jumps suffer from the over-smoothness artifact. We replace these pixels in the left disparity map by the ones in the right map that exactly point to them. In such a situation, the corresponding pixels of A are far apart from those of B in the right disparity map so that they are more reliable to preserve depth discontinuities (Fig. 4b). Outliers at near positive disparity jumps are considered as left occlusions where no correspondence exists in the right image. The inconsistent pixels in occlusions are properly extrapolated by local high confident surrounding pixels, using a voting scheme. Based on the support region, each unreliable pixel p collects votes from reliable neighbors as follows:

$$\mathrm{Vote}_p(d)=\{r\,|\,r\in U(p)\wedge d_r=d\}. \qquad (26)$$

It means that if r is a consistent pixel in the support region of p, the disparity d_r contributes one vote which is accumulated in the set $\mathrm{Vote}_p(d)$. Accordingly, the final disparity of the pixel is decided by the maximum vote number:

$$d_p^*=\arg\max_{d\in[d_{\min},d_{\max}]}\left|\mathrm{Vote}_p(d)\right|. \qquad (27)$$

We implement the voting process for several iterations. The filled outliers are used as reliable pixels for the next round of iterations so that the occlusion regions can be extrapolated step by step.

4 Experimental results

4.1 Performance evaluation

We evaluate the performance of the proposed method on the benchmark of the Middlebury stereo test bed (Scharstein and Szeliski, 2001). The following four stereo image pairs with corresponding ground truth are tested: 'Tsukuba', 'Venus', 'Teddy', and 'Cones'. The parameters are set constant for different datasets in our experiments: Trunc=20, τ=15, λ=5, T=3.6, η=6. The disparity image of each test dataset is compared with the ground truth, and the pixels where the disparity absolute difference is greater than 1 are counted as errors. The percentage of

error pixels is computed at three different kinds of regions: non-occluded regions, the whole image (all valid regions in the ground-truth image), and pixels near discontinuities.

Fig. 5 shows the estimated disparity maps for all four test images. As can be inferred, the use of an adaptive support window in matching cost aggregation yields higher accuracy than the fixed-window based approach. The accuracy of the result also benefits much from the integration of the aggregation with the scanline optimization based framework. With the aid of two-pass scanline optimization, the result is significantly improved at the depth border as well as in the homogeneous regions. Additionally, the 'streaking' artifact is eliminated after inter-scanline optimization. However, the result suffers from inaccuracies as evidenced by the 'Cones' image. This is partly due to the scene that contains some repetitive patterns not explicitly handled in the proposed method.

Table 1 summarizes an objective evaluation result for comparison with those of other DP/SO based or fast implemented stereo correspondence algorithms. According to the bad-pixels measure, the proposed algorithm is competitive with other existing algorithms. Most notably, our result is the best among the methods that can be implemented in real/near-real time.

The experiments are further performed on some live scenes provided by FhG-HHI (ISO/IEC, 2008) and other more challenging images in new Middlebury dataset. Fig. 6 shows the estimated disparity maps for the 'Book Arrival' sequence. Fig. 7 shows the results for the 'Dolls' and 'Moebius' images. These examples demonstrate the ability of our approach in producing promising results with fine structure as well as distinct object boundaries.

4.2 Implementation and execution time evaluation

We first test the proposed algorithm on a standard PC with a 3.16 GHz Intel Core 2 Duo CPU. Current C++ implementation without a specific optimization takes several seconds for 'Tsukuba' and

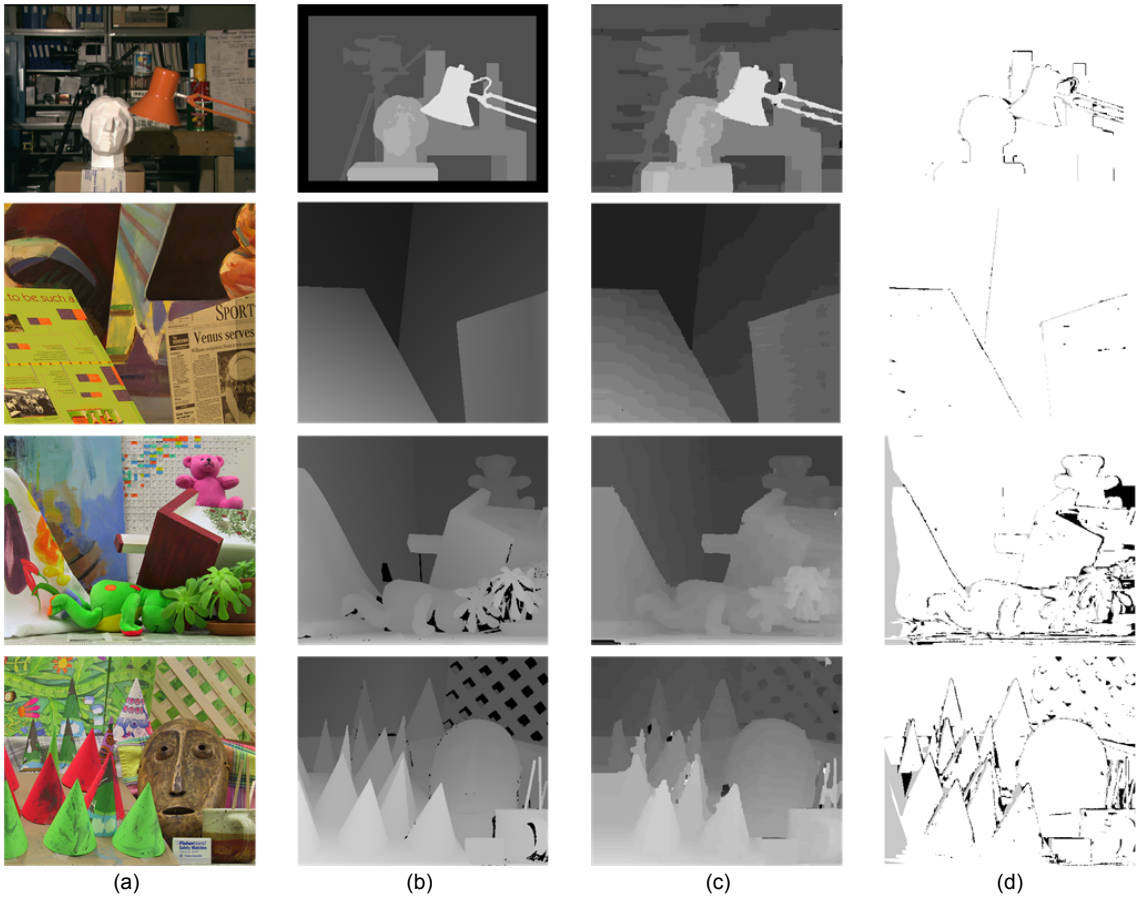

Fig. 5 Results on Middlebury stereo datasets 'Tsukuba', 'Venus', 'Teddy', and 'Cones' (from top to bottom)
(a) Reference images; (b) Ground truth images; (c) Our results; (d) Bad pixels

Table 1 Quantitative evaluation of the proposed algorithm and related algorithms on the four Middlebury stereo datasets, comparing the percentage of bad pixels in non-occluded regions (nonoc), the whole region (all), and pixels near depth discontinuities (disc)

Algorithm	Percentage of bad pixels (%)												Average percentage (%)
	Tsukuba			Venus			Teddy			Cones			
	nonoc	all	disc	nonoc	all	disc	nonoc	all	disc	nonoc	all	disc	
SO+borders	1.29	1.71	6.83	0.25	0.53	2.26	7.02	12.2	16.3	3.90	9.85	10.2	6.03
Proposed method	**1.06**	**2.31**	**5.72**	**0.35**	**0.77**	**3.67**	**5.96**	**11.8**	**15.4**	**4.83**	**11.3**	**13.2**	**6.36**
RegionTreeDP	1.39	1.64	6.85	0.22	0.57	1.93	7.42	11.9	16.8	6.31	11.9	11.8	6.56
VariableCross	1.99	2.65	6.77	0.62	0.96	3.20	9.75	15.1	18.2	6.28	12.7	12.9	7.60
RealtimeBP	1.49	3.40	7.87	0.77	1.90	9.00	8.72	13.2	17.2	4.61	11.6	12.4	7.69
RealTimeABW	1.26	1.67	6.83	0.33	0.65	3.56	10.7	18.3	23.3	4.81	12.6	10.7	7.90
FastAggreg	1.16	2.11	6.06	4.03	4.75	6.43	9.04	15.2	20.2	5.37	12.6	11.9	8.24
OptimizedDP	1.97	3.78	9.80	3.33	4.74	13.0	6.53	13.9	16.6	5.17	13.7	13.4	8.83
RealtimeVar	3.33	5.48	16.8	1.15	2.35	12.8	6.18	13.1	17.3	4.66	11.7	13.7	9.05
RealTimeGPU	2.05	4.22	10.6	1.92	2.98	20.3	7.23	14.4	17.6	6.41	13.7	16.5	9.82
ReliabilityDP	1.36	3.39	7.25	2.35	3.48	12.2	9.82	16.9	19.5	12.9	19.9	19.7	10.7
TreeDP	1.99	2.84	9.96	1.41	2.10	7.74	15.9	23.9	27.1	10.0	18.3	18.9	11.7

Algorithms are listed in increasing order of the average error percentage of all test images. Algorithms used for comparison include: SO+borders (Mattoccia *et al*., 2007), RegionTreeDP (Lei *et al*., 2006), VariableCross (Zhang *et al*., 2009), RealtimeBP (Yang *et al*., 2006), RealTimeABW (Gupta and Cho, 2010), FastAggreg (Tombari *et al*., 2008b), OptimizedDP (Salmen *et al*., 2009), RealtimeVar (Kosov *et al*., 2009), RealTimeGPU (Wang *et al*., 2006), ReliabilityDP (Gong and Yang, 2005), and TreeDP (Veksler, 2005)

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

Fig. 6 Results on the 'Book Arrival' sequence

(a) Reference image of frame 4; (b) Dense disparity map of frame 4; (c) Reference image of frame 39; (d) Dense disparity map of frame 39

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

Fig. 7 Results on 'Dolls' and 'Moebius' images

(a) Reference image for the 'Dolls' dataset; (b) Dense disparity map for the 'Dolls' dataset; (c) Reference image for the 'Moebius' dataset; (d) Dense disparity map for the 'Moebius' dataset

'Venus' stereo pairs. For 'Teddy' and 'Cones' images whose disparity search range is 0–60, the runtime is more than one minute. It is probably because the global optimization step takes a computation complexity which is proportional to the square of disparity levels. Since our algorithm has a high potential to be programmed in parallel, we employ the huge computational power and the parallel processing capabilities of GPU to decrease the execution time. In the second experiment, the algorithm is implemented on the same CPU and an NVIDIA Quadro FX 4800 graphics card with 24×8 CUDA (Compute Unified

Device Architecture) parallel processing cores. Each step of the algorithm is mapped to the SIMD (single instruction multiple data) model of the CUDA kernel. Table 2 summarizes the execution time of different datasets with both CPU and GPU implementations. As shown in the table, with the GPU implementation we attain a speedup factor of more than 200 compared with the CPU implementation executed on the same computer. The speed is about 35 frames/s for a typical stereo image with a resolution of 384×288 and 16 disparity levels. On average (average of all datasets listed in Table 2), our GPU implementation of the proposed approach achieves approximately 57 million disparity estimations per second (MDE/s), which is sufficient for many real-time applications.

Table 2 Evaluation of execution time with CPU and GPU implementations

Dataset	Resolution	Number of disparity levels	Time (ms)		MDE/s	
			CPU	GPU	CPU	GPU
Tsukuba	384×288	16	6798	28.749	0.26	61.55
Venus	434×383	20	15270	51.678	0.22	64.33
Teddy	450×375	60	114457	214.039	0.09	47.30
Cones	450×375	60	114035	211.870	0.09	47.79
Book Arrival	512×384	20	18627	63.328	0.21	62.09

MDE/s: million disparity estimations per second

5 Conclusions

In this paper, we focus on providing a fast and accurate solution to the stereo correspondence problem. The proposed approach employs a local aggregation strategy together with a scanline optimization based framework. For a reliable dissimilarity measure, the initial matching cost is carefully aggregated over an adaptive support region. Pixels in weak textured regions are effectively assembled by aggregation to reduce the matching ambiguities. Then in the disparity optimization step, the global energy minimization framework on a tree structure allows for improving the accuracy at both the homogenous region and the object border in an efficient way. Moreover, the typical 'streaking' artifact is reduced significantly since the piecewise smoothness is enforced in both horizontal and vertical directions. Further improvements are obtained by applying the symmetric consistency check technique. The unreliable pixels near depth discontinuities are detected and extrapolated with more credible results. In particular, using the processing capability and parallelism of commodity graphics hardware, our current implementation achieves more than 60 MDE/s on a common image with 16 disparity levels. Evaluation on the Middlebury stereo benchmark shows that our algorithm outperforms all the other real-time stereo algorithms. In the future, we plan to further investigate the configurable massive parallelism of a field programmable gate array (FPGA) platform to realize the algorithm in an embedded system.

References

Bobick, A.F., Intille, S.S., 1999. Large occlusion stereo. *Int. J. Comput. Vis.*, **33**(3):181-200. [doi:10.1023/A:1008150329890]

Criminisi, A., Blake, A., Rother, C., Shotton, J., Torr, P.H.S., 2007. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *Int. J. Comput. Vis.*, **71**(1):89-110. [doi:10.1007/s11263-006-8525-1]

Crow, F.C., 1984. Summed-Area Tables for Texture Mapping. Proc. 11th Annual Conf. on Computer Graphics and Interactive Techniques, p.207-212. [doi:10.1145/800031.808600]

Darabiha, A., Rose, J., MacLean, W.J., 2003. Video-Rate Stereo Depth Measurement on Programmable Hardware. IEEE Conf. on Computer Vision and Pattern Recognition, p.203-210. [doi:10.1109/CVPR.2003.1211355]

Gong, M., Yang, Y., 2005. Near Real-Time Reliable Stereo Matching Using Programmable Graphics Hardware. IEEE Conf. on Computer Vision and Pattern Recognition, p.924-931. [doi:10.1109/CVPR.2005.246]

Gong, M., Yang, Y., 2007. Real-time stereo matching using orthogonal reliability-based dynamic programming. *IEEE Trans. Image Process.*, **16**(3):879-884. [doi:10.1109/TIP.2006.891344]

Gupta, R.K., Cho, S.Y., 2010. Real-Time Stereo Matching Using Adaptive Binary Window. Proc. 3DPVT, Session 2, p.1-8.

Hartley, R.I., Zisserman, A., 2004. Multiple View Geometry. Cambridge University Press, Cambridge, UK.

Hirschmüller, H., Innocent, P., Garibaldi, J., 2002. Real-time correlation-based stereo vision with reduced border errors. *Int. J. Comput. Vis.*, **47**(1-3):229-246. [doi:10.1023/A:1014554110407]

ISO/IEC, 2008. HHI Test Material for 3D Video. M15413.

Kim, J.C., Lee, K.M., Choi, B.T., Lee, S.U., 2005. A Dense Stereo Matching Using Two-Pass Dynamic Programming with Generalized Ground Control Points. IEEE Conf. on Computer Vision and Pattern Recognition, p.1075-1082. [doi:10.1109/CVPR.2005.22]

Konolige, K., 1997. Small Vision System: Hardware and Implementation. Proc. Int. Symp. on Robotics Research, p.111-116.

Kosov, S., Thormählen, T., Seidel, H., 2009. Accurate real-time disparity estimation with variational methods. *LNCS*, **5875**:796-807. [doi:10.1007/978-3-642-10331-5_74]

Lei, C., Selzer, J., Yang, Y., 2006. Region-Tree Based Stereo Using Dynamic Programming Optimization. IEEE Conf. on Computer Vision and Pattern Recognition, p.2378-2385. [doi:10.1109/CVPR.2006.251]

Mattoccia, S., Tombari, F., di Stefano, L., 2007. Stereo vision enabling precise border localization within a scanline optimization framework. *LNCS*, **4844**:517-527. [doi:10.1007/978-3-540-76390-1_51]

McDonnell, M.J., 1981. Box-filtering techniques. *Comput. Graph. Image Process.*, **17**(1):65-70. [doi:10.1016/S0146-664X(81)80009-3]

Moallemi, C.C., van Roy, B., 2010. Convergence of min-sum message-passing for convex optimization. *IEEE Trans. Inform. Theory*, **56**(4):2041-2050. [doi:10.1109/TIT.2010.2040863]

Mulligan, J., Isler, V., Daniilidis, K., 2002. Trinocular stereo: a real-time algorithm and its evaluation. *Int. J. Comput. Vis.*, **47**(1-3):51-61. [doi:10.1023/A:1014525320885]

Salmen, J., Schlipsing, M., Edelbrunner, J., Hegemann, S., Lueke, S., 2009. Real-time stereo vision: making more out of dynamic programming. *LNCS*, **5702**:1096-1103. [doi:10.1007/978-3-642-03767-2_133]

Sara, R., 2010. How to Teach Stereoscopic Matching? Proc. 52nd Int. Symp. ELMAR, p.445-453.

Scharstein, D., Szeliski, R., 2001. Middlebury Stereo Vision Page. Available from http://vision.middlebury.edu/stereo/ [Accessed on Oct. 20, 2011].

Scharstein, D., Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.*, **47**(1):7-42. [doi:10.1023/A:1014573219977]

Tombari, F., Mattoccia, S., di Stefano, L., Addimanda, E., 2008a. Classification and Evaluation of Cost Aggregation Methods for Stereo Correspondence. IEEE Conf. on Computer Vision and Pattern Recognition, p.1-8. [doi:10.1109/CVPR.2008.4587677]

Tombari, F., Mattoccia, S., di Stefano, L., Addimanda, E., 2008b. Near Real-Time Stereo Based on Effective Cost Aggregation. Int. Conf. on Pattern Recognition, p.1-4. [doi:10.1109/ICPR.2008.4761024]

Veksler, O., 1999. Efficient Graph-Based Energy Minimization Methods in Computer Vision. PhD Thesis, Cornell University, New York, USA.

Veksler, O., 2003. Fast Variable Window for Stereo Correspondence Using Integral Images. IEEE Conf. on Computer Vision and Pattern Recognition, p.556-561. [doi:10.1109/CVPR.2003.1211403]

Veksler, O., 2005. Stereo Correspondence by Dynamic Programming on a Tree. IEEE Conf. on Computer Vision and Pattern Recognition, p.384-390. [doi:10.1109/CVPR.2005.334]

Wang, L., Liao, M., Gong, M., Yang, R., Nister, D., 2006. High-Quality Real-Time Stereo Using Adaptive Cost Aggregation and Dynamic Programming. 3rd Int. Symp. on 3DPVT, p.798-805. [doi:10.1109/3DPVT.2006.75]

Weiss, Y., Freeman, W.T., 2001. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. Inform. Theory*, **47**(2):736-744. [doi:10.1109/18.910585]

Woodfill, J., Herzen, B.V., 1997. Real-Time Stereo Vision on the PARTS Reconfigurable Computer. Proc. IEEE Symp. on FPGAs for Custom Computing Machines, p.201-210. [doi:10.1109/FPGA.1997.624620]

Yang, Q., Wang, L., Yang, R., Wang, S., Liao, M., Nister, D., 2006. Real-Time Global Stereo Matching Using Hierarchical Belief Propagation. The British Machine Vision Conf., p.989-998.

Yang, R., Welch, G., Bishop, G., 2002. Real-Time Consensus-Based Scene Reconstruction Using Commodity Graphics Hardware. Proc. 10th Pacific Conf. on Computer Graphics and Applications, p.225-234. [doi:10.1109/PCCGA.2002.1167864]

Yoon, K., Kweon, I., 2006. Adaptive support-weight approach for correspondence search. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(4):650-656. [doi:10.1109/TPAMI.2006.70]

Zhang, K., Lu, J., Lafruit, G., 2009. Cross-based local stereo matching using orthogonal integral images. *IEEE Trans. Circ. Syst. Video Technol.*, **19**(7):1073-1079. [doi:10.1109/TCSVT.2009.2020478]