Cross-Based Local Stereo Matching Using Orthogonal Integral Images

Ke Zhang, Jiangbo Lu, and Gauthier Lafruit

***Abstract*— We propose an area-based local stereo matching algorithm for accurate disparity estimation across all image regions. A well-known challenge to local stereo methods is to decide an appropriate support window for the pixel under consideration, adapting the window shape or the pixelwise support weight to the underlying scene structures. Our stereo method tackles this problem with two key contributions. First, for each anchor pixel an upright cross local support skeleton is adaptively constructed, with four varying arm lengths decided on color similarity and connectivity constraints. Second, given the local cross-decision results, we dynamically construct a shape-adaptive full support region on the fly, merging horizontal segments of the crosses in the vertical neighborhood. Approximating image structures accurately, the proposed method is among the best performing local stereo methods according to the benchmark Middlebury stereo evaluation. Additionally, it reduces memory consumption significantly thanks to our compact local cross representation. To accelerate matching cost aggregation performed in an arbitrarily shaped 2-D region, we also propose an orthogonal integral image technique, yielding a speedup factor of 5–15 over the straightforward integration.**

***Index Terms*— Cross-based region construction, orthogonal integral images, shape adaptive approximation, stereo matching.**

I. Introduction

STEREO MATCHING as an important vision problem estimates disparities from a given stereo image pair. A substantial amount of work on this topic has been surveyed and evaluated by Scharstein and Szeliski [1]. Different from most global stereo matching methods that are computationally expensive and involve many parameters, local stereo methods are generally efficient and easy to implement. To reduce the image ambiguity, local stereo methods commonly aggregate the support from the neighboring pixels in a given size-constrained window. For accurate disparity estimates near depth discontinuities, a local support window is desired to adapt its shape and size, therefore only collecting the support from the pixels of the same depth. To this end, many local stereo methods have been proposed, and they roughly fall into two categories.

Manuscript received May 21, 2008; revised September 2, 2008. First version published April 7, 2009; current version published July 22, 2009. This paper was recommended by Associate Editor H. Chen.

K. Zhang and J. Lu are with the Department of Electrical Engineering, University of Leuven, Leuven, 3001, Belgium, and Multimedia Group, Interuniversity Microelectronics Center, Leuven, 3001, Belgium (e-mail: zhangke@imec.be; jiangbo.lu@imec.be).

G. Lafruit is with the Multimedia Group, Interuniversity Microelectronics Center, Leuven, 3001, Belgium (e-mail: gauthier.lafruit@imec.be).

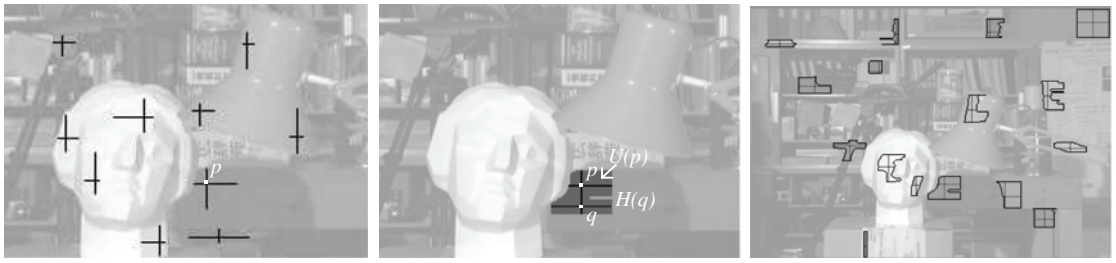
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2020478

The stereo methods in the first category focus on either the optimal support window selection among predefined multiple windows [2], [3], or the pixelwise adaptation of the local support window's shape and size [4], [5]. However, a common limitation to these methods is that the shape of a local support window is rectangular or constrained, and hence is inappropriate for pixels near arbitrarily shaped depth discontinuities. In our recent work [6], a pointwise shape adaptive support polygon is built on a varying-scale sectorial basis, yielding accurate disparity results. Nevertheless, the rigid polygons are still not flexible enough to approximate various scene structures. Moreover, the adaptive scale decision method and cost aggregation over adaptive polygons are not efficient.

On the other hand, local stereo methods from the second category adjust the support weights of the pixels in a given support window while fixing the shape and size of a support window. For instance, Xu *et al*. [7] determined adaptive support weights by radial computations, but this method is very sensitive to the initial disparity estimation. Yoon and Kweon [8] assigned a support weight to the pixel in a support window based on color similarity and geometric proximity. Despite of the accurate disparity results, this method consumes a huge amount of memory due to the storage of the center pixel-dependent support weights. Employing the image segmentation information, Tombari *et al*. [9] proposed a modified weight function for every pixel in a large (51×51) support window. Their improved disparity accuracy is at the cost of a significant computational complexity increase [10].

In this letter, we propose a cross-based local stereo matching algorithm. The key algorithmic ideas are twofold. First, a locally adaptive upright cross is decided upon the color similarity, defining an initial support skeleton for the anchor pixel. Second, we dynamically construct a shape-adaptive full support region in the cost aggregation step, reusing the pre-computed neighboring cross configurations. The end result is that appropriate local support regions are efficiently derived from the fairly compact cross-based representation, leading to accurate disparity estimates for different pixel locations. Decomposing the conventional cost aggregation into two orthogonal 1-D integrations, we further propose an orthogonal integral image (OII) technique for fast cost aggregation over any arbitrarily shaped windows in constant time. Our OII method represents a novel instantiation of the general integral image technique [11], previously applied to computing rectangular sums only [5], [12]. Overall, the proposed method is among the best performing



(a) (b) (c)

Fig. 1. Cross-based local support region representation and construction on the *Tsukuba* image [13]. (a) A pixelwise adaptive cross defines a local support skeleton for the anchor pixel, e.g., p. (b) A shape-adaptive full support region $U(p)$ is dynamically constructed for the pixel p, integrating multiple horizontal line segments $H(q)$ of neighboring crosses. (c) Sample shape-adaptive local support regions, approximating local image structures appropriately.

local stereo methods based on the Middlebury stereo evaluation [13]. In particular, it has pronounced advantages in the execution time compared to the state-of-the-art local methods [6], [8], [9].

In the rest of this letter, Section II presents our cross-based local stereo matching method. The OII technique for fast cost aggregation is described in Section III. We show experimental results in Section IV and conclude the letter in Section V.

II. Cross-Based Local Stereo Matching

A. Cross-Based Local Support Region Construction

For accurate local stereo matching, it is important to decide an appropriate local support region for each pixel adaptively. In principle, this local support region should contain only the neighboring pixels from the same depth with the pixel under consideration. However, without disparity information beforehand, the support region for a pixel can only be adequately derived from the raw images. A common assumption is that pixels with similar intensity within a constrained area are likely from the same image structure, therefore having similar disparity. Utilizing this assumption, we propose a cross-based local support region representation and construction approach.

The key idea of the proposed approach is to decide an upright cross for every pixel $p = (x_p, y_p)$ in the input image I. As shown in Fig. 1(a), this pixelwise adaptive cross consists of two orthogonal line segments, intersecting at the anchor pixel p. We represent the horizontal segment by $H(p)$ and the vertical segment by $V(p)$, and they jointly define the local support skeleton for the pixel p. Instead of fixing the size of a local cross, we adaptively change its four arm lengths to reliably capture the local image structure. More specifically, to configure an appropriate cross for the pixel p, we first decide a quadruple $\{h_p^-, h_p^+, v_p^-, v_p^+\}$ that denotes the left, right, up, and bottom arm length, respectively (see Fig. 2).

As our cross-based representation is a general concept, there are actually a variety of specific implementations to decide the arm lengths $\{h_p^-, h_p^+, v_p^-, v_p^+\}$. Here, we present an efficient approach based on color similarity under the connectivity constraint [6]. First, we apply a 3×3 median filter to the input image I, suppressing the impact of image noise as well as subtle non-Lambertian effects. Next, the arm lengths are decided upon color similarity. Taking h_p^- as an example,

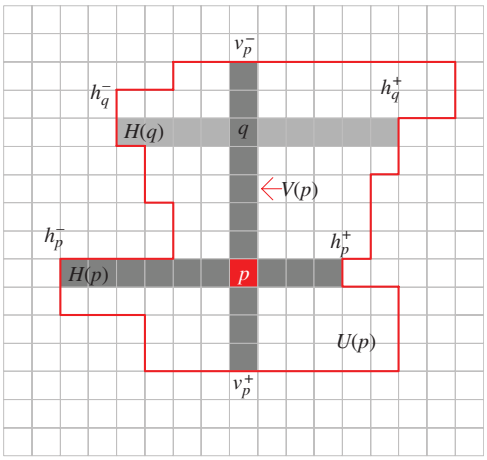


Fig. 2. Configuration of a local upright cross $H(p) \cup V(p)$ for the anchor pixel p, and the constructed full support region $U(p)$. The quadruple $\{h_p^-, h_p^+, v_p^-, v_p^+\}$ defines the left, right, up, and bottom arm length of the cross, respectively. $q \in V(p)$ is a pixel on the vertical segment $V(p)$ in (2).

we perform a color similarity testing for a consecutive set of pixels which reside on the left horizontal side of the pixel p. The purpose is to search for the largest left span r^*, where all the pixels covered are similar to the anchor pixel p in color. The computation of r^* can be formulated as follows:

$$r^* = \max_{r \in [1,L]} \left(r \prod_{i \in [1,r]} \delta(p, p_i) \right) \tag{1}$$

where $p_i = (x_p - i, y_p)$ and L is the preset maximum arm length. $\delta(p_1, p_2)$ is an indicator function evaluating the color similarity between the pixel p_1 and p_2 based on all color bands

$$\delta(p_1, p_2) = \begin{cases} 1, & \max\limits_{c \in \{\mathrm{R,G,B}\}} \left(|I_c(p_1) - I_c(p_2)| \right) \le \tau \\ 0, & \text{otherwise} \end{cases}$$

where I_c is the intensity of the color band c. Set empirically, τ controls the confidence level of color similarity. Once the largest left span r^* is derived from (1), we set the left arm length $h_p^- = \max(r^*, 1)$. In effect, this enforces a minimum support region of 3×3 for robust correspondence matching.

Based on the arm lengths $\{h_p^-, h_p^+, v_p^-, v_p^+\}$ decided for the pixel p, two orthogonal cross segments $H(p)$ and $V(p)$ are

$$\begin{cases} H(p) &= \{(x, y) \mid x \in [x_p - h_p^-, x_p + h_p^+], y = y_p\} \\ V(p) &= \{(x, y) \mid x = x_p, y \in [y_p - v_p^-, y_p + v_p^+]\}. \end{cases} \quad (2)$$

Fig. 2 shows the local cross configuration schematically. Apparently, for each pixel p we only need to store four arm lengths, to represent an adaptive local support cross. This compact representation is in sharp contrast to the adaptive support weight method of a huge memory demand [8].

Given the pixelwise local cross decision results, we can readily construct a shape-adaptive full support region $U(p)$ for the pixel p. The key idea is to model the local support region $U(p)$ as an area integral of multiple horizontal segments $H(q)$, sliding along the vertical segment $V(p)$ of the pixel p

$$U(p) = \bigcup_{q \in V(p)} H(q) \quad (3)$$

where q is a support pixel located on the vertical segment $V(p)$. From the local cross representation computed for the pixel q, we can easily retrieve its horizontal segment $H(q)$. In essence, $H(q)$ provides a connected set of valid support pixels in the 2-D neighborhood of the pixel p, as shown in Fig. 2. Note that $U(p)$ can also be constructed from multiple vertical segments, utilizing $H(p)$ as the anchor axis. In fact, our experiments show that there is only a slight difference in functional performance between the two configurations.

Fig. 1(b) and (c) demonstrate the effectiveness of the proposed technique on the *Tsukuba* image [13]. Our cross-based local support construction method yields appropriate shape-adaptive windows, closely approximating local image structures. This local support region adaptation is clearly desirable for accurate disparity estimation across different image areas.

As we model a local support region $U(p)$ by a vertical integral of multiple horizontal segments $H(q)$, there is a big potential to reduce computation redundancy in practice. Reusing the "overlapping" data computed first from the horizontal scan, we propose an optimization technique in Section III.

B. Locally Adaptive Matching Cost Aggregation

As an area-based local stereo matching method, the proposed algorithm places a key emphasis on the cost aggregation step (see Fig. 3). Given a pair of hypothetical correspondences, i.e., $p = (x_p, y_p)$ in the left image I and $p' = (x'_{p'}, y'_{p'})$ in the right image I', we measure the matching cost between them by aggregating raw matching costs in a local support region. Here, the coordinates of p and p' are correlated with a disparity hypothesis d, i.e., $x'_{p'} = x_p - d$ and $y'_{p'} = y_p$. 多窗口方法

For reliable cost aggregation, unlike most of existing local methods [2], [5] we symmetrically consider both local support regions $U(p)$ and $U'(p')$ decided for the pixel p and p', respectively. If we only consider the local support region

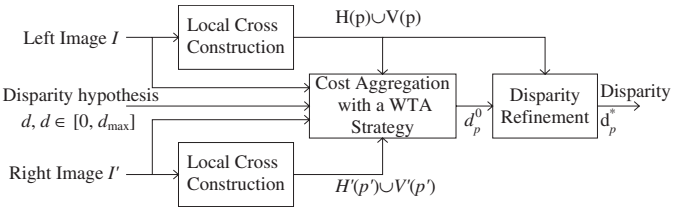


Fig. 3. Framework of the proposed local stereo matching algorithm.

$U(p)$ for the left image, the matching cost aggregation will be polluted by outliers in the right image, i.e., pixels from different depths with the pixel p' in the support window. Therefore, combining two local support regions to define the aggregation region, the normalized matching cost $\overline{E}_d(p)$ between the pixel p and p' is computed as follows:

$$\overline{E}_d(p) = \frac{1}{\|U_d(p)\|} E_d(p) = \frac{1}{\|U_d(p)\|} \sum_{s \in U_d(p)} e_d(s) \quad (4)$$

with

$$U_d(p) = \{(x, y) \mid (x, y) \in U(p), (x - d, y) \in U'(p')\}.$$

In (4), $e_d(s)$ denotes the pixel-based raw matching cost and $U_d(p)$ is the combined local support region that contains only the valid pixels, likely having similar disparities with the anchor pixels p and p' in both images. Because of the restriction of $U'(p')$, $U_d(p) \subseteq U(p)$. $\|U_d(p)\|$ is the number of support pixels in $U_d(p)$, used to normalize the aggregated cost $E_d(p)$. The raw matching cost is computed from a pair of corresponding pixels, i.e., s in the left image and s' in the right image (with a disparity value d)

$$e_d(s) = \min\left(\sum_{c \in \{R,G,B\}} \left| I_c(s) - I'_c(s') \right|, T \right) \quad (5)$$

where T controls the truncation limit of the matching cost.

C. Disparity Selection and Refinement

After matching cost aggregation, we use a Winner-Takes-All (WTA) strategy [1] for the disparity selection as follows:

$$d_p^0 = \arg\min_d \overline{E}_d(p), \quad d \in [0, d_{max}] \quad (6)$$

where d_{max} is the maximum value of possible disparities. d_p^0 is the initial disparity estimate for the pixel p. It can be further refined using a local high-confidence voting scheme developed in our previous work [6]. As shown in Fig. 1(c), pixels contained in a local support region mostly originate from the same scene patch, and hence they share similar disparities. For each anchor pixel p, if we create a histogram φ_p of the initial disparity estimate d_s^0, a distribution peak is very likely to occur, where $s \in U(p)$ is a pixel in the adaptive neighborhood $U(p)$. The histogram bin associated with this peak corresponds to a statistically optimal disparity value d_p^*, with which we implicitly perform a piecewise smoothness regularization [6]. Accordingly, the final disparity of the pixel p, d_p^*, is decided as 局部窗口内的投票策略?

$$d_p^* = \arg\max_d \varphi_p(d), \quad d \in [0, d_{max}]. \quad (7)$$

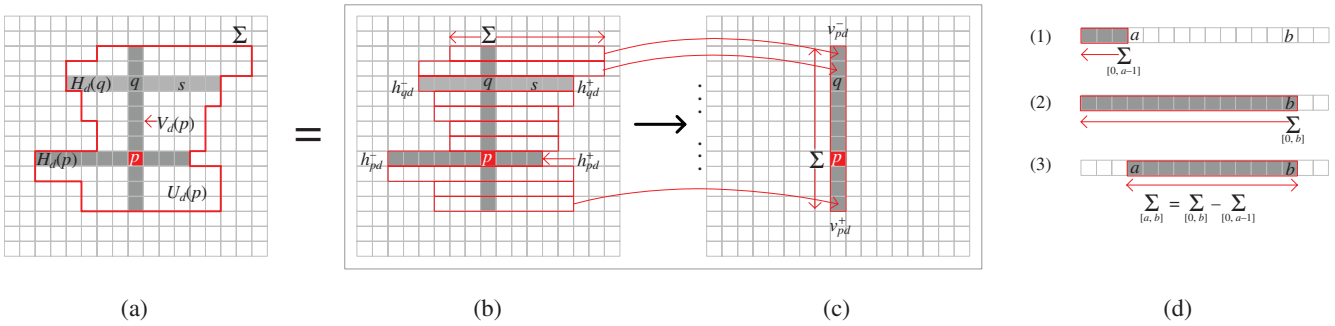


Fig. 4. Illustration of the proposed OII technique. (a) In the matching cost aggregation step of (4), raw matching costs are aggregated from an arbitrarily shaped local region $U_d(p)$, for each anchor pixel p. The combined aggregation region $U_d(p)$ is an expansion of the combined local cross $H_d(p) \cup V_d(p)$. In the proposed OII technique, we decompose the matching cost aggregation into two orthogonal 1-D integration steps, i.e., (b) a horizontal integration pass followed by (c) a vertical integration pass, and (d) a degenerated 1-D integral image technique is further applied to accelerate these two 1-D integration steps.

III. Fast Cost Aggregation with the OII Technique

Aggregating raw matching costs in a shape-adaptive support region enhances disparity accuracy, but on the other side it also poses a challenge for fast implementation. The reason is that multiple raw cost additions—in an irregular support region $U_d(p)$—are required for each anchor pixel p. As (4) shows, the computational load is directly correlated with the local support region size $\|U_d(p)\|$. In fact, for the *Tsukuba* image, $\|U_d(p)\|$ on average is as large as 320.

Instead of summing raw matching costs $e_d(s)$ in (4) directly, we propose an efficient OII technique to accelerate the aggregation over irregularly shaped regions. The major ideas are twofold. First, we decompose cost aggregation performed in an arbitrarily shaped region into two orthogonal 1-D aggregations, namely, a horizontal aggregation step, followed by a vertical one. In such a way, the computational load becomes linear to the 1-D span of the support region, rather than the area size. Second, we accelerate these two orthogonal 1-D aggregation steps by pre-computing a horizontal integral image as well as a vertical one. This optimization makes 1-D aggregation over any line segments of varying lengths in constant time. Compared with the conventional application of the integral image technique in fast rectangular sum computation [5], the proposed OII technique goes one step further. It provides a general approach for fast aggregation over an irregularly shaped 2-D area.

Before presenting the procedure of the proposed OII technique, we first formulate the combined aggregation region $U_d(p)$ as an expansion of the combined local cross, similar to the local support region $U(p)$ in (3). For a pixel $p = (x_p, y_p)$, the combined local cross consists of two orthogonal line segments $H_d(p)$ and $V_d(p)$, as shown in Fig. 4(a)

$$\begin{cases} H_d(p) &= \{(x, y_p) \mid (x, y_p) \in H(p), (x-d, y_p) \in H'(p')\} \\ V_d(p) &= \{(x_p, y) \mid (x_p, y) \in V(p), (x_p-d, y) \in V'(p')\}. \end{cases} \tag{8}$$

Here, $H(p) \cup V(p)$ and $H'(p') \cup V'(p')$ are the local cross configurations for the pixel p and its correspondence p', respectively. Given the pixelwise combined local cross $H_d(p) \cup V_d(p)$ decided from (8), the combined local aggregation region $U_d(p)$ for the pixel p can be precisely expressed as

$$U_d(p) = \bigcup_{q \in V_d(p)} H_d(q) \tag{9}$$

where q is a pixel located on the vertical segment $V_d(p)$.

Substituting the above orthogonal decomposition of $U_d(p)$ into (4), we now transform a double integral of raw matching costs $e_d(s)$ into two orthogonal iterated integrals. More specifically, the aggregated cost $E_d(p)$ is computed as follows:

$$E_d(p) = \sum_{s \in U_d(p)} e_d(s) = \sum_{q \in V_d(p)} \left(\sum_{s \in H_d(q)} e_d(s) \right) = \sum_{q \in V_d(p)} E_d^H(q) \tag{10}$$

where $E_d^H(q)$ represents the result after the horizontal integration step. Clearly, the computation of $E_d(p)$, over an irregular aggregation region $U_d(p)$, is now decomposed into a horizontal pass plus a vertical pass [see Fig. 4(b) and (c)]. As a result, the proposed OII technique reduces computation redundancy significantly, by reusing the horizontal integration result $E_d^H(q)$ among the pixels in the vertical neighborhood. In addition, we have adopted the general integral image technique [11] in a degenerated 1-D form, further accelerating the integration over any horizontal or vertical line segments [see Fig. 4(d)].

The overall OII technique can be summarized as the following four steps.

Step 1: Given the pixelwise raw matching cost $e_d(x, y)$, we first build a horizontal integral image $S^H(x, y)$, storing the cumulative row sum as

$$\begin{aligned} S^H(x, y) &= \sum_{0 \le m \le x} e_d(m, y) \\ &= S^H(x-1, y) + e_d(x, y). \end{aligned} \tag{11}$$

TABLE I
Matching Cost Aggregation Speedup Using Our OII Technique

	Tsukuba	*Venus*	*Teddy*	*Cones*
Image resolution	384×288	434×383	450×375	450×375
Max disparity d_{max}	15	19	59	59
Straightforward time (s)	1.6	4.0	7.8	4.5
Our OII time (s)	0.14	0.26	0.82	0.84
Speedup factor	11.4	15.4	9.5	5.4

TABLE II
QUANTITATIVE EVALUATION RESULTS OF AREA-BASED LOCAL STEREO METHODS FOR THE ORIGINAL MIDDLEBURY STEREO DATABASE [13]

Algorithm	*Tsukuba*			*Sawtooth*			*Venus*			*Map*	
	All	Untex.	Disc.	All	Untex.	Disc.	All	Untex.	Disc.	All	Disc.
Our method	**2.06**	**2.16**	**7.41**	**0.99**	**0.08**	**3.93**	**0.69**	**0.29**	**3.72**	**1.02**	**9.55**
Adapt.Weights [8]	1.51	0.65	7.24	1.14	0.27	5.48	1.14	0.61	4.49	1.47	13.58
Adapt. Polygon [6]	2.29	1.98	9.39	1.32	0.32	4.77	0.80	0.61	3.67	0.70	9.53
Variable Win. [5]	2.35	1.65	12.17	1.28	0.23	7.09	1.23	1.16	13.35	0.24	2.98
Compact Win. [4]	3.36	3.54	12.91	1.61	0.45	7.87	1.67	2.18	13.24	0.33	3.94
Bay. Diff. [14]	6.49	11.62	12.29	1.45	0.72	9.29	4.00	7.21	18.39	0.20	2.49
Shiftable Win. [3]	5.23	3.80	24.66	2.21	0.72	13.97	3.74	6.82	12.94	0.66	9.35

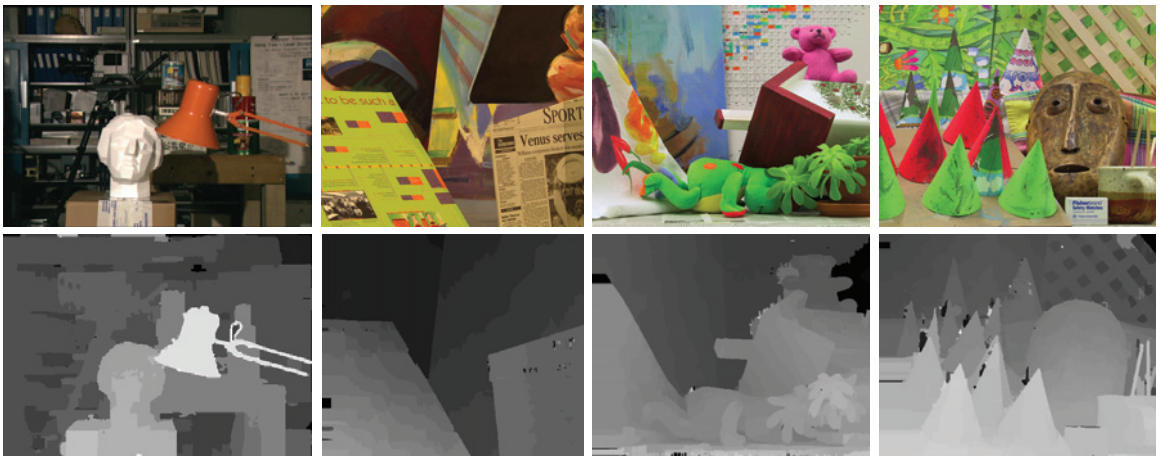


Fig. 5. Dense disparity maps for the *Tsukuba*, *Venus*, *Teddy*, and *Cones* stereo datasets (from left to right), using the proposed local stereo matching algorithm. Top row: the input left images. Bottom row: the resulting disparity maps. Rather than estimating two disparity maps for left-right consistency check [8], [9], we applied a simple border extrapolation step here. All the results are available at http://vision.middlebury.edu/stereo/eval.

Here, $S^H(x, y)$ can be iteratively computed from $S^H(x-1, y)$ with only one addition. When $x = 0$, $S^H(-1, y) = 0$.

Step 2: For each pixel $q = (x_q, y_q)$ on the left image lattice, we then compute the horizontal integral $E_d^H(q)$ in (10), using the horizontal integral image $S^H(x, y)$, as follows:

$$E_d^H(q) = S^H\left(x_q + h_{qd}^+, y_q\right) - S^H\left(x_q - h_{qd}^- - 1, y_q\right). \quad (12)$$

As shown in Fig. 4(a), h_{qd}^- and h_{qd}^+ are the left and right arm lengths of the combined aggregation cross, decided for the anchor pixel q. Fig. 4(d) illustrates the subtraction in (12).

Step 3: Taking the computed horizontal matching cost $E_d^H(x, y)$ as the input, we create a vertical integral image $S^V(x, y)$. It stores the cumulative column sum as

$$S^V(x, y) = \sum_{0 \le n \le y} E_d^H(x, n) = S^V(x, y-1) + E_d^H(x, y). \quad (13)$$

As in Step 1, only one addition is needed to compute $S^V(x, y)$. When $y = 0$, $S^V(x, -1) = 0$.

Step 4: Based on the vertical integral image $S^V(x, y)$, we derive the fully aggregated matching cost $E_d(p)$ for the pixel $p = (x_p, y_p)$, with one final subtraction

$$E_d(p) = S^V\left(x_p, y_p + v_{pd}^+\right) - S^V\left(x_p, y_p - v_{pd}^- - 1\right). \quad (14)$$

Similarly, v_{pd}^- and v_{pd}^+ are the up and bottom arm lengths of the combined aggregation cross, for the anchor pixel p.

Following the above steps, we only need four additions/subtractions for an anchor pixel to aggregate raw matching costs over any arbitrarily shaped regions, regardless of the region size. As a comparison, we measured the respective execution time of the straightforward cost aggregation in (4) and that of our proposed OII technique. Including also memory access overhead in the total execution time, Table I shows that our OII technique leads to a speedup factor of 5–15 over the straightforward computation method. The speedup factor generally increases as the average size of the combined aggregation region $\|U_d(p)\|$ increases. As a result, the *Venus* dataset gains the largest speedup among the test images, because it has large piecewise planar objects (see Fig. 5).

TABLE III
QUANTITATIVE EVALUATION RESULTS FOR THE NEW MIDDLEBURY STEREO DATABASE (NOCC. MEANS NON-OCCLUDED REGIONS) [13]

Algorithm	*Tsukuba*			*Venus*			*Teddy*			*Cones*			Average percent of bad pixels
	Nocc.	All	Disc.	Nocc.	All	Disc.	Nocc.	All	Disc.	Nocc.	All	Disc.	
SegSupport [9]	1.25	1.62	6.68	0.25	0.64	2.59	8.43	14.2	18.2	3.77	9.87	9.77	6.44
Adapt. Weights [8]	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26	6.67
Our method	**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 [15]	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
2OP+occ [16]	2.91	3.56	7.33	0.24	0.49	2.76	10.9	15.4	20.6	5.42	10.8	12.5	7.75
Layered [17]	1.57	1.87	8.28	1.34	1.85	6.85	8.64	14.3	18.5	6.59	14.7	14.4	8.24
MultiCamGC [18]	1.27	1.99	6.48	2.79	3.13	3.60	12.0	17.6	22.0	4.89	11.8	12.1	8.31

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

We evaluated the performance of the proposed stereo matching method using the benchmark Middlebury stereo database [13]. The parameters are set constant in all experiments on different stereo datasets, i.e., $L = 17$, $\tau = 20$, and $T = 60$.

For the original Middlebury stereo database with four test pairs, i.e., *Tsukuba*, *Sawtooth*, *Venus*, and *Map*, Table II summarizes the quantitative performance of our method and those of other area-based local stereo methods, roughly in descending order of overall performance. Specifically, Table II reports the percentage of "bad pixels" whose absolute disparity error is greater than 1. For each pair of images, the error rates for all regions, untextured regions (untex., except for the *Map* image), and depth discontinuities (disc.) are reported respectively. In general, the proposed method is the best among the state-of-the-art area-based local methods, outperforming the leading local methods [6], [8].

Fig. 5 and Table III present the visual and quantitative results for the new Middlebury stereo database [13], which includes more challenging stereo images, i.e., *Teddy* and *Cones*. The proposed method yields discontinuity-preserving smooth disparity maps for the new testbed images. It even achieves a better performance than some complex global stereo matching methods [16]–[18], as shown in Table III. Though our method is slightly outperformed by the two local methods [8], [9], its execution speed is enormously faster than theirs as shown later.

测试算法中参数的稳定性也是一个常见的步骤

We have also examined the robustness of the proposed method when varying parameter settings. First, fixing τ to 20, we changed the value of L in (1). Fig. 6(a) shows that when L is larger than 15, the proposed method is fairly insensitive to the maximum possible arm length L of a local cross. Second, fixing L to 17, we changed the value of τ, as shown in Fig. 6(b). In general, the disparity accuracy remains nearly constant, when τ varies from 18 to 28.

Without any special code-level optimization, the proposed local stereo method runs at a favorable speed on a 3.0 GHz Intel Pentium 4 CPU. The execution time for the *Tsukuba*, *Venus*, *Teddy*, and *Cones* stereo datasets is 0.9, 1.6, 2.4, and 2.4 s, respectively. This speed is dozens of times faster than the adaptive support weight method [8], which takes 60 s for the *Tsukuba* image on an AMD 2700+ machine. On a more advanced CPU than ours, Tombari *et al*'s approach [9] even needs about 33 min for the *Teddy* image [10].

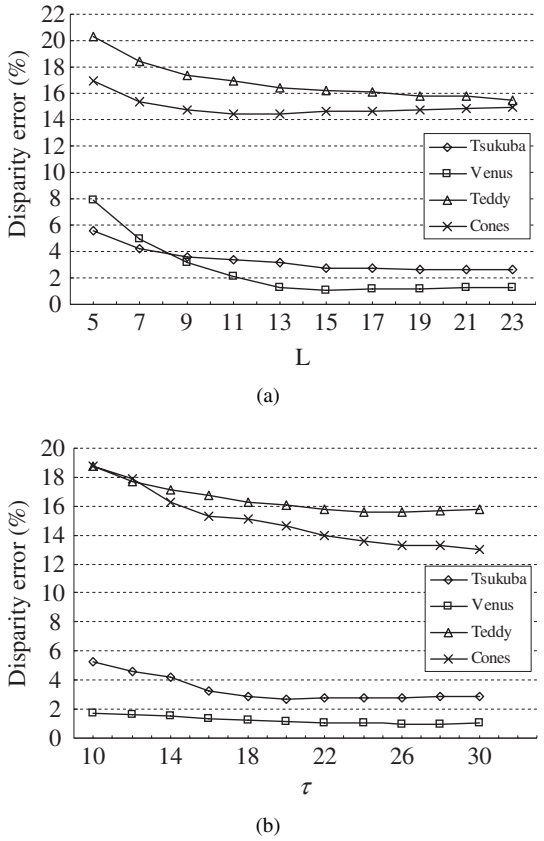


Fig. 6. Performance evaluation (disparity error rates for all regions) of the proposed method when varying the parameter values, tested on four stereo images [13]. (a) Changing L while $\tau = 20$. (b) Changing τ while $L = 17$.

V. CONCLUSION

This letter has proposed a cross-based local stereo matching algorithm. Based on the pixelwise compact cross, we dynamically construct shape-adaptive local support regions that approximate varying image structures accurately. Evaluation with the Middlebury stereo benchmark shows that the proposed method is ranked among the best performing local methods. Furthermore, we have proposed an efficient orthogonal integral image technique, which accelerates cost aggregation over irregularly shaped support windows. In future work, we intend to optimize the proposed stereo method on a parallel platform.

REFERENCES

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

[2] A. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, San Juan, PR, 1997, pp. 858–863.

[3] S. B. Kang, R. Szeliski, and J. Chai, "Handling occlusions in dense multi-view stereo," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, vol. 1, 2001, pp. 103–110.

[4] O. Veksler, "Stereo correspondence with compact windows via minimum ratio cycle," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1654–1660, Dec. 2002.

[5] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, vol. 1, 2003, pp. 556–561.

[6] J. Lu, G. Lafruit, and F. Catthoor, "Anisotropic local high-confidence voting for accurate stereo correspondence," in *Proc. SPIE-IS&T Electron. Imaging*, vol. 6812, Jan. 2008, pp. 605822-1–605822-10.

[7] Y. Xu, D. Wang, T. Feng, and H.-Y. Shum, "Stereo computation using radial adaptive windows," in *Proc. Int. Conf. Pattern Recognition*, vol. 3, 2002, pp. 595–598.

[8] K. J. Yoon and S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, Apr. 2006.

[9] F. Tombari, S. Mattoccia, and L. D. Stefano, "Segmentation based adaptive support for accurate stereo correspondence," in *Proc. Pacific-Rim Symp. Image Video Technol.*, 2007, pp. 427–438.

[10] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2008, pp. 1–8.

[11] F. Crow, "Summed-area tables for texture mapping," in *Proc. ACM SIGGRAPH*, 1984, pp. 207–212.

[12] P. Viola and M. Jones, "Robust real-time face detection," in *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[13] D. Scharstein and R. Szeliski, *Middlebury Stereo Vision Page (2008)*. [Online]. Available: http://vision.middlebury.edu/stereo/

[14] D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *Int. J. Comput. Vision*, vol. 28, no. 2, pp. 155–174, 1998.

[15] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nistr, "Realtime global stereo matching using hierarchical belief propagation," in *Proc. Brit. Mach. Vision Conf.*, 2006, pp. 989–998.

[16] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon, "Global stereo reconstruction under second order smoothness priors," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Anchorage, U.K., 2008, pp. 1–8.

[17] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video interpolation using a layered representation," in *Proc. ACM SIGGRAPH*, 2004, pp. 600–608.

[18] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Proc. Eur. Conf. Comput. Vision*, 2002, pp. 8–40.