Classification and evaluation of cost aggregation methods for stereo correspondence

Federico Tombari
federico.tombari@unibo.it

Stefano Mattoccia
stefano.mattoccia@unibo.it

Luigi Di Stefano
luigi.distefano@unibo.it

Elisa Addimanda
elisa.addimanda@studio.unibo.it

Department of Electronics Computer Science and Systems (DEIS), University of Bologna
Advanced Research Center on Electronic Systems (ARCES), University of Bologna

Abstract

In the last decades several cost aggregation methods aimed at improving the robustness of stereo correspondence within local and global algorithms have been proposed. Given the recent developments and the lack of an appropriate comparison, in this paper we survey, classify and compare experimentally on a standard data set the main cost aggregation approaches proposed in literature. The experimental evaluation addresses both accuracy and computational requirements, so as to outline the best performing methods under these two criteria.

1. Introduction

Given a pair of stereo images, the aim of stereo matching algorithms is to determine the disparity values of the pixels belonging to the image selected as *reference* view. Scharstein and Szelinski [23] provided a valuable taxonomy and evaluation of dense stereo matching algorithms for rectified image pairs, arguing that most algorithms perform four steps: matching cost computation, cost aggregation, disparity computation/optimization and disparity refinement. This paper focuses on the cost aggregation step and addresses methods that perform cost aggregation on a *variable support*. The idea at the basis of the variable support concept is to determine the best set of pixels on which to compute the matching cost (i.e. the *support*) at each pair of candidates under evaluation (i.e. the *correspondence*). Hence, unlike the basic approach that relies on a fixed static support (typically a squared window or a single point), these methods deploy a support which varies along the potential correspondences in order to adapt itself to the local characteristics of each correspondence. This allows for obtaining higher accuracy along depth borders and lower matching ambiguity, especially within low textured regions.

Although works dealing with cost aggregation on a variable support date back to the 70s, 80s and early 90s [19, 2, 21, 11], only in the last years a broad research activity has provided effective ideas allowing local algorithms based on a variable support to yield an accuracy comparable to that of many global methods. Moreover, though typically performed by local algorithms, cost aggregation on a variable support proved to be very effective in improving the performance of global algorithms such as based on Belief Propagation (BP) [31], Dynamic Programming (DP) [28], Scanline Optimization (SO) [20].

We believe that the variety of approaches, as well as the excellent results achieved, deserve a specific classification, highlighting similarities and differences between the main cost aggregation strategies, together with a comparative performance evaluation of the different methods. Recent surveys on stereo matching [23, 6, 16, 14] do not address the above topics since they consider the whole class of stereo methods [23], review advances in computational stereo with particular emphasis on occlusion detection and real-time methods [6], focus on matching functions robust to photometric distortions and noise [16] or address only those cost aggregation methods that are suited to real-time implementation on a GPU [14]. Differently, this paper is specifically focused on classifying the main variable support-based cost aggregation strategies and comparing them experimentally within a plain vanilla *Winner-Take-All* (WTA) framework.

In particular, we firstly compare the considered strategies in terms of accuracy of the retrieved disparity maps based on the methodology proposed in [23]. Then, we extend the evaluation methodology in [23] by comparing strategies also in terms of computational complexity. Moreover, by evaluating jointly the two parameters of comparison we highlight the methods that better trade-off between accuracy and computational complexity.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

2. Classification of cost aggregation strategies

Although several methods concerning the idea of variable support constrain the cost aggregation step to rely on rectangular windows with fixed weights only, alternatives to this basic idea aimed at improving accuracy and mainly based on either two different approaches, have been proposed. The former generalizes the concept of variable support by allowing the support to have any shape instead of being built upon rectangular windows only. The latter assigns adaptive - rather than fixed - weights to the points belonging to the support. While these approaches aim at improved accuracy, on the other hand the irregularity of the support hardly allows for deployment of incremental calculation schemes, thus yielding potentially higher computational costs. It is also worth to point out that, with a few exceptions, most of the cost aggregation strategies determine the support on the basis of a symmetric scheme deploying information from both images.

As for the matching (or error) function employed, this is typically based on the L_p distance between the two vectors representing the supports in the stereo images, such as the *Sum of Absolute Differences* (SAD) or *Sum of Squared Differences* (SSD). Often M-estimators are used to achieve better robustness toward outliers. The basic one simply truncates the values of the matching measure up to a threshold (*e.g. Truncated SAD* [30, 25]), while sometimes other more complex M-estimators are used [12]. Another popular solution regards the use of a measure insensitive to image sampling [3] (e.g. used in [27]). Moreover, a promising similarity function based on point distinctiveness has been recently proposed in [32].

Finally, it is worth pointing out that this work addresses aggregation strategies based on fronto-parallel variable supports only, thus not considering proposals that account for three-dimensional supports (e.g. [33]).

2.1. Cost aggregation based on rectangular windows

Let I_r and I_t be respectively the reference and target image of a rectified stereo pair. Let p, the point at coordinate (x, y) in I_r, and q, the point at coordinate $(x + d, y)$ in I_t, be the two points for which stereo correspondence is currently being evaluated, and let $w_n^r(i, j)$, $w_n^t(i, j)$ denote two squared windows of side n centered on (i, j) respectively in I_r, I_t. We also denote as $W_n(i, j, d)$ the pair of windows $w_n^r(i, j)$, $w_n^t(i + d, j)$.

A first category of variable support methods relies on a fixed set of rectangular window pairs, $S(p, q)$, symmetrically defined on I_r and I_t. When evaluating correspondence (p, q) a subset of $S(p, q)$, determined according to a specific criterion and referred to hereinafter as $S_V(p, q)$ represents the current support. Since $S_V(p, q)$ varies at each correspondence under evaluation, it should adapt itself to the local characteristics of p and q, thus enabling better handling of depth borders and low-textured areas with respect to the use of a fixed static support. The local matching cost is then obtained by computing an error function over $S_V(p, q)$.

It is worth observing that within this category the support at each correspondence depends on both I_r and I_t, since its determination is typically based on the error function itself, whose value depends on both images. Moreover, each weight assigned to the points in the various windows is fixed and does not depend on the image content. Finally, an important advantage of the methods based on this idea is that, since they deploy rectangular windows, they can often exploit incremental schemes in order to achieve significant computational efficiency.

2.1.1 Varying window size and/or offset

One of the first algorithms exploiting the idea of using a set of windows to improve the accuracy of stereo correspondence is *Shiftable windows* [23]. In this case, the set of windows $S(p, q)$ is defined as:

$$S(p, q) = \{W_n(i, j, d) : i \in [x-n, x+n], j \in [y-n, y+n]\} \quad (1)$$

where n is a parameter of the algorithm representing the chosen window size. The support at each correspondence, $S_V(p, q)$, is given by the window minimizing the error function over $S(p, q)$. This approach is useful along depth borders, since it aims at determining the most appropriate displacement with respect to p on which to center the window in order to aggregate points lying at the same depth plane as p. A variation of this basic strategy concerns including in $S(p, q)$ only 9 squared windows in symmetrical positions with respect to the central point [4, 10].

An alternative approach [21, 1] is to vary the size of the window rather than its displacement by properly selecting n between a minimum value N_{min} and a maximum value N_{max}:

$$S(p, q) = \{W_n(x, y, d) : n \in [N_{min}, N_{max}]\} \quad (2)$$

This allows, e.g., to employ bigger windows within low-textured regions.

These schemes can be generalized by selecting the best support between a set of window pairs having different sizes and different displacements. In [17] the best displacement is selected by means of a shiftable window approach, while, to determine the size of the support, starting from $n = N_{min}$ the window is iteratively enlarged until a given minimum variance of the error function is reached.

A slightly more general approach is represented by the method proposed in [27], which selects as support the window minimizing a matching cost over a set of windows

$S(p,q)$ defined as:

$$S(p,q) = \{W_n(i,j,d) : n \in [N_{min}, N_{max}], \\ i \in [x-n, x+n], j \in [y-n, y+n]\} \quad (3)$$

The 3 criteria on which the matching cost is based include minimization of the error function and its variance, plus the use of a biasing weight to favor the choice of large windows within low-textured regions, where the error function and its variance might not vary significantly along the evaluated window sizes. Moreover, this method explicitly proposes an incremental scheme aimed at efficiently computing (3) at each new correspondence.

An analogous approach was proposed in [7]. As for this method, the best displacement is selected out of 9 using the shiftable windows approach. Then, the window size is iteratively decreased until either the error function gets worse or the minimum window size is reached.

In [9] the displacements considered at each correspondence are 4, disposed on the four window corners. As for the window size, starting from an initial value, the window horizontal and vertical sides are iteratively increased until either the error variance on a direction gets higher than a certain threshold or the error function gets worse. Differently from previous approaches, this allows to obtain rectangular supports.

2.1.2 Selecting more than one window

All previous schemes select, for each correspondence, one window on each image representing the best support over $S(p,q)$. A generalization of this approach is represented by $S_V(p,q)$ being not one single window pair, but a subset of window pairs. In [22], $S(p,q)$ is the same as in (1) and the outcome of the error measure computation on the various window pairs is used to assess whether each point is close or not to a depth edge. Based on that, a variable support strategy is deployed on all points detected as close to a depth edge, where the final matching cost assigned to each correspondence is obtained by averaging the error function along those displacement positions detected as lying on the same border side as p and q.

In one version of method [15], $S(p,q)$ is defined as a set of 5 squared windows

$$S(p,q) = W_n(x,y,d) \cup \{W_n(x \pm n, y \pm n, d)\} \quad (4)$$

At each correspondence the variable support is obtained as the union of 3 *best supporting* windows (i.e., on I_r, the one centered on p plus those 2 out of the 4 windows around p scoring the lowest error function, and symmetrically on I_t). Variations of this scheme employ variable supports of a total of either 5 or 13 best supporting windows out of a set including, respectively, 9 and 25 windows.

2.1.3 Associating different weights to window points

It is worth pointing out that with the methods described in Section 2.1.2 the resulting shape of the support is no longer constrained to a rectangle. Moreover, getting a support made out of several partially overlapping windows having uniform weights is equivalent to getting a support made out of a single rectangular window where each point is weighted differently. This latter strategy concerns the explicit assignment of different weights to the points of each window belonging to $S(p,q)$. In method [18], the aggregation stage defines $S(p,q)$ as a set of 108 rod-shaped windows. Each window is characterized by a specific orientation and weight set, and the support at each correspondence is determined by the window minimizing the error function. Each point is then classified as *homogeneous* or *heterogeneous* based on the outcome of the application of a LoG filter, and on those points denoted as homogeneous the minimum error score determined on $S(p,q)$ is also compared to that yielded by a basic shiftable windows approach.

In the strategy proposed in [13], $S(p,q)$ is defined as a set of 5×5 window pairs centered on (p,q), each window point being characterized by a weight belonging to the set $\{0,1,2,4\}$. For each correspondence a window pair is selected according to the local structure of I_r, which is extracted by means of either edge detection or segmentation. The authors propose also to iterate the process k times, and suggest to use $k = 4$ or $k = 2$ respectively in a local or global framework.

2.2. Cost aggregation based on unconstrained shapes

An important generalization to the idea of determining $S(p,q)$ as a set of rectangular windows is to allow supports to have any shape. This potentially allows supports to better adapt to the local characteristics of the data, though sometimes this approach does not translate into computationally efficient algorithms.

The first method exploiting this idea was proposed in [5]. At each correspondence (p,q), each pixel p_i on I_r is classified either as plausible or not-plausible based on an estimation of the photometric relation between p_i and its correspondent on I_q at the same disparity as (p,q). The best disparity for p is simply selected as that yielding the largest set of connected plausible pixels. This allows to have variable supports which can ideally extend to all pixels of the image. Differently, in [26] the support shape at each correspondence is represented by a polygonal line around p, which is extracted by applying the *minimum ratio cycle* technique.

Finally, in [12] the support shape is represented by the intersection between the segment on which p lies, G_p, and a squared window centered on p, $w_n^r(x,y)$. This approach is intrinsically based on the assumption, introduced by [24],

that disparity is constant over each segment obtained from a segmentation process. Moreover, this approach relies only on segmentation information concerning I_r (i.e. it is not symmetrical). Those points belonging to $w_n^r(x, y)$ and not to G_p are included in the error function by means of a small constant weight.

2.3. Use of adaptive weights

Another important generalization of the variable support concept refers to the assignment of different and variable weights to the points surrounding p and q. The concept of support and shape are more controversial in this case: since every point receives a weight, the distinction between points belonging or not to the support is seamless. Moreover, since the whole set of weights has to be re-computed at each new correspondence, the variable support typically does not include the whole image but only a subset of points represented by a squared or a round window centered on p and q, with the assumption that points lying farther than a certain distance are uncorrelated. Once the weights are determined, the error function is typically computed by weighting each pixel-wise error measurement with the corresponding coefficient.

The method proposed in [29] can be regarded as the first proposal for stereo exploiting this idea. It was inspired by [8], which proposed a method to segment a foreground object from its background in an image based on the radial propagation of similarity starting from a foreground point. In [29] 3 different cues are deployed to determine the support weights for points belonging to the reference view I_r. The first one (the *certainty*) is based on the variance of the error function: since weights are propagated radially starting from p, each point weight depends from the error variance of previous points along its ray. With increasing variances, the assigned weight is lower since it corresponds to a low certainty. The two other cues are color and disparity distribution correlation: the weight assigned to a point p_i increases as the difference in the color space between p_i and p decreases and as the correlation between p_i and p disparity distribution increases. Each cue is weighted by means of a gaussian function in the final weight formulation, the 3 gaussian variances being 3 parameters of the method.

In [30] this approach is enhanced by symmetrically extending the weight computation to points on I_t. Weights are computed based on the two cues of distance in the color space and distance in the coordinate space (proximity) by means of gaussian functions, this approach being motivated by the Gestalt principles of similarity. Then, a final weighted error function is proposed including normalization by means of the weight coefficients. Points farther than a certain distance from p and q are not evaluated. An efficient though simplified asymmetric version of this method is proposed within a Dynamic Programming framework in [28], so to allow a GPU implementation with real-time capabilities.

Finally, the method proposed in [25] improves [30] by demonstrating that the cue of proximity can lead to incorrect weight assignments along depth borders as well as in presence of low-textured regions, high-textured regions and repetitive patterns. Hence, instead of this cue, the output of a segmentation process is exploited so as to embody more effectively information about color-spatial connectivity between points.

3. Experimental evaluation and comparison

In this section an extensive performance evaluation and comparison between different cost aggregation strategies based on a variable support is proposed. Since the aim is to specifically evaluate the effectiveness and efficiency of the various aggregation methods, all the considered strategies have been embodied into the same plain WTA framework. The two criteria used for the evaluation are accuracy and computational cost. Evaluation according to the first criterion is accomplished by using the Middlebury Stereo Evaluation Dataset[1] [23]. Computational cost is assessed by measuring for each method the execution time needed to process a reference stereo pair on the same machine (Intel Core Duo 2.14 GHz CPU, 2 GB RAM).

The selected approaches are those that represent the state-of-the-art for the different classes of cost aggregation strategies identified in Section 2. In particular, 14 methods are compared, which are now listed together with the nickname used hereinafter to refer them to. The basic method that uses a fixed square window is referred to as *Fixed window*. As for the approaches based on a selection over a set of windows, we evaluated *Shiftable window* [4], *Reliability* [17], *Variable windows* [27], *Recursive adaptive* [7], *Multiple adaptive* [9], *Multiple windows* [15] (tested in the 3 versions based respectively on 5, 9 and 25 *supporting windows*), *Oriented rod* [18], *Gradient guided* [13]. With regards to the approaches that allow for unconstrained support shapes we evaluated *Max connected* [5] and *Segmentation based* [12]. Finally, within the methods based on adaptive weights we considered *Radial adaptive* [29], *Adaptive weight* [30] and *Segment support* [25].

In order to carry out an as fair as possible comparison, all method were implemented using the same criteria, except in the case of *Adaptive weight* [30] for which the authors' source code is publicly available. For each method, only the proposed aggregation stage was implemented and tested. In particular, neither pre-processing stages nor typical post-processing stages, such as median filtering and left-right consistency check, were applied. In order to better assess the performance of *Oriented rod* and *Multiple*

[1] http://vision.middlebury.edu/stereo

windows, where the proposed pre-processing stage is intrinsically connected with the aggregation stage, for both methods we considered two versions, that is with and without pre-processing. Those versions where pre-processing was excluded will be denoted hereinafter with the symbol $*$. Since for *Oriented rod* pre-processing served as a way to discriminate between homogeneous and heterogeneous points, in the version without pre-processing all points are considered homogeneous and thus compared with the result of a shiftable filter. This generally implies higher computation times and in some case better accuracy. Moreover, for the sake of fairness the cost function is the same for all methods. In particular, since many aggregation strategies rely on colour information [13, 12, 29, 30, 25], the selected cost function is the *Sum of Absolute Differences* (SAD) on RGB pixels, exception made for method *Max connected*, which was implemented as originally proposed by the authors since it is not explicitly based on a cost function. Finally, for each method which was not originally proposed with this cost function or where parameter values were not explicitly specified by authors, parameter values were selected by means of a tuning process ran on the considered dataset.

As far as execution times are concerned, in our implementations we took into account all the guidelines and details originally proposed by the authors, including e.g. the use of incremental schemes for a more efficient implementation (e.g. Variable windows [27]). Our implementations of the basic *Fixed window* and of *Shiftable windows* also deploy incremental schemes. When implementation details were not explicitly provided by the authors we adopted the same plain criteria across the considered algorithm, so as to the render the comparison of execution times as fair as possible. However, it is clear that by extensively optimising each algorithm according to its own structure one would get different and perhaps much faster execution times. Therefore, the reported measurements should be interpreted only as useful indicators aimed at comparing the computational costs of the considered methods.

3.1. Analysis of extracted supports

Fig. 1 allows for a qualitative evaluation and comparison of the variable supports extracted by the considered methods on 6 representative hand-chosen points of the stereo pair *Teddy*. The selected points, highlighted in the top picture of Fig. 1, refer to regions where stereo methods based on fixed or variable support are often ambiguous due to one or more of the following causes: presence of depth borders (points 1, 2, 4, 6), low-textured areas (points 2, 5), highly textured areas (point 3). For those methods associating weights to points belonging to the support, higher weights are represented by brighter grayscale values. Furthermore, since method *Max connected* can have supports extending to the whole image area, only for this method, for each of the evaluated point, each extracted support is shown on a different picture (indicated in brackets in the figure). Moreover, for the sake of simplicity the supports displayed for [13] are relative to $k = 1$ (although in our implementation we use $k = 4$, as originally proposed in [13], to obtain the experimental results shown in Subsection 3.2).

From a qualitative point of view, it is worth noticing how aggregation strategies deploying sets of window pairs are generally able to adapt their supports according to the position of the depth border (points 1, 2, 4, 6), though it seems clear that supports made out of rectangular windows lack in flexibility. For instance, this can be clearly seen at point 4 for methods *Shiftable windows*, *Reliability*, *Multiple window (25W)*.For what regards low-textured regions (points 2, 5), only a subset of methods which allow the support windows to vary their size (i.e. *Multiple Adaptive*, *Reliability* and, to some extent, *Variable windows*) succeeds in correctly expanding over these regions.

As for methods deploying supports characterized by unconstrained shapes, method *Segmentation based* seems to adapt very well its supports along depth borders as well as in presence of low-textured regions. As for method *Max connected*, though generally it correctly limits the support shape when approaching a depth border, it often annexes points at different disparities causing ambiguities in the disparity retrieval stage. Moreover, this causes the extracted supports for points 4 and 5 to coincide, even though these two points lie at a different disparity.

Finally, all methods deploying adaptive weights seem to extract the supports with notable accuracy. Together with *Segmentation based*, they outperform other aggregation schemes, leading to the best variable supports. Moreover, since they evaluate a high number of points surrounding the correspondence currently evaluated, this often allows them to include within the same support a high number of points lying at the same disparity: this turns out to be effective especially in presence of depth borders and low-textured regions. Between these methods, *Segment support* and *Radial adaptive* seem more effective than *Adaptive weight* within low-textured regions (points 2, 5), while *Segment support* also handles better the considered high-textured region (point 3) compared to the other two approaches which, conversely, at this point retrieve very small supports.

3.2. Accuracy and computational cost

For what concerns accuracy we rely on a testbed and evaluation methodology analogous to that adopted on the Middlebury Stereo Evaluation site. In particular, as it can be seen from Table 1, we use 4 reference stereo pairs (Tsukuba, Venus, Teddy and Cones) and for each of them evaluate the error rates on the two ground truth maps *NonOcc* (all

Algorithm	Rank Accuracy	Tsukuba NonOcc	Tsukuba Disc	Venus NonOcc	Venus Disc	Teddy NonOcc	Teddy Disc	Cones NonOcc	Cones Disc	Rank Time	Time (mm:ss)	Average Rank
Segment support [25]	1.00	2.28 1	7.50 1	1.21 1	5.88 1	10.99 1	22.01 1	5.42 1	11.83 1	17	33:34	9.00
Adaptive weight [30]	2.50	4.66 3	8.25 2	4.61 3	13.30 4	12.70 2	22.40 2	5.50 2	11.90 2	15	18:14	8.75
Variable Windows [27]	4.00	4.10 2	10.79 3	10.66 13	9.94 2	13.93 3	25.53 3	7.24 3	13.86 3	11	00:25	7.50
Reliability [17]	5.38	5.14 4	18.31 5	3.86 2	11.51 3	16.96 6	30.62 6	13.52 13	21.55 4	16	21:51	10.69
Shiftable Windows [4]	5.63	6.53 7	21.80 8	6.60 5	13.54 5	16.16 5	30.19 5	9.55 4	22.99 6	7	00:15	6.31
Segmentat. based [12]	7.38	8.18 10	18.77 6	8.06 8	20.85 7	15.78 4	29.66 4	13.22 12	24.55 8	2	00:02	4.69
Multi. Win. (25W)* [15]	8.13	6.52 6	21.91 9	6.77 6	21.57 8	18.60 10	33.11 9	11.87 10	23.69 7	9	00:16	8.56
Recursive Adaptive [7]	10.00	9.22 14	26.69 16	8.36 9	14.86 6	18.48 9	32.93 8	11.60 9	24.80 9	14	16:55	12.00
Gradient Guided [13]	10.50	7.51 9	16.20 4	13.07 14	33.10 13	20.39 13	32.82 7	13.67 14	25.60 10	3	00:03	6.75
Mult. Win. (9W)* [15]	10.75	8.51 11	27.59 17	6.47 4	34.30 15	17.57 8	38.04 13	10.75 6	26.60 12	6	00:13	8.38
Mult. Win. (5W)* [15]	11.25	9.36 15	25.74 14	8.57 10	38.65 17	17.11 7	37.45 11	9.86 5	25.33 11	8	00:16	9.63
Mult. Win. (25W) [15]	11.38	6.34 5	24.13 11	9.04 11	29.61 10	20.77 14	36.77 10	14.20 15	27.45 15	10	00:17	10.69
Mult. Win. (9W) [15]	11.50	7.12 8	25.00 13	10.21 12	33.44 14	18.91 12	37.76 12	10.95 7	27.05 14	5	00:09	8.25
Mult. Win. (5W) [15]	13.00	8.94 13	23.55 10	16.33 15	35.56 16	22.29 15	38.09 14	11.13 8	26.99 13	4	00:07	8.50
Fixed Window	14.25	8.66 12	36.67 20	7.05 7	40.53 19	18.68 11	41.95 17	12.79 11	33.32 17	1	< 1 s	7.63
Multiple Adaptive [9]	15.88	15.11 19	32.85 19	16.88 16	22.31 9	25.40 16	39.44 15	21.40 17	29.66 16	18	39:47	16.94
Max Connected [5]	15.88	11.81 17	26.39 15	42.47 20	50.87 20	34.46 18	41.01 16	17.70 16	22.70 5	20	≈ 2 h	17.94
Oriented Rod [18]	16.63	11.29 16	24.21 12	26.33 18	30.09 11	37.68 19	47.94 19	39.60 19	47.88 19	12	12:22	14.31
Oriented Rod* [18]	17.50	15.87 20	27.75 18	26.40 19	30.58 12	30.65 17	42.72 18	30.52 18	41.82 18	13	12:33	15.25
Radial Adaptive [29]	17.50	14.84 18	21.79 7	22.40 17	40.40 18	49.64 20	50.13 20	50.18 20	53.60 20	19	58:52	18.25

Table 1. Performance evaluation in terms of accuracy and execution times of the considered variable-support based strategies.

points except for occluded areas) and *Disc* (only points along depth discontinuities, not including occluded areas). Each single error rate is also denoted by its respective ranking along the considered methods. Error rates on all image points including occlusions have not been taken into account since the tested algorithms do not explicitly handle disparity retrieval for occluded points due to the adopted WTA framework. Besides, an overall accuracy ranking obtained by averaging the single rankings of each method along the dataset is shown in the second column.

As for computational costs, Table 1 reports for each method the measured execution time on the stereo pair *Teddy*. Similarly to the evaluation of accuracy, the Table shows the ranking of methods according to the measured execution times. Finally, the Table reports in the rightmost column the ranking obtained by averaging the overall accuracy ranking and the time ranking, so as to highlight the methods that better trade-off between accuracy and computational efficiency.

Coherently with the qualitative analysis based on Fig. 1, the Table shows that the most accurate methods are those deploying adaptive weights. In particular, *Segment support* [25] and *Adaptive weight* [30] outperform the other methods almost on the whole dataset, followed by *Variable Windows* [27]. Conversely, the fastest methods are those based on the evaluation of the support over a set of windows or based on unconstrained shapes. It is worth observing that, apart from the basic *Fixed window* approach, methods such as *Segmentation based* [12], *Gradient guided* [13], *Multiple window* [15], *Shiftable window* [4] and *Variable windows* [27] can run in seconds or tens of seconds, while some methods, i.e. *Max connected* [5] and *Radial adaptive* [29] may require hours. As regards the accuracy/efficiency trade-off, the best method turned out to be *Segmentation based* [12], with similar rankings obtained by *Shiftable Windows* [4] and *Gradient guided* [13].

The disparity maps obtained by the various methods on the Middlebury dataset as well as the qualitative comparison of the extracted supports concerning the *Tsukuba* and *Teddy* stereo pairs can be found on-line[2]. In addition, this web site includes the results dealing with other cost measures (Truncated SAD, SSD) and the program used for generating the supports depicted in Fig. 1.

4. Conclusions

A survey and evaluation of stereo matching methods based on a variable support has been proposed. The first part, concerning classification, illustrated a categorization of the approaches proposed in literature. Then, in the second part, a vast experimental comparison between variable support-based methods within a WTA framework has been presented. The evaluation work has been aimed at assess-

[2]available at www.vision.deis.unibo.it/spe

ing both accuracy and computational complexity of the considered methods. This allowed us to determine also which methods provide better trade-offs between accuracy and computational complexity. Future work will be aimed at extending this comparison by including other cost aggregation methods not considered in this paper, such as e.g. [26] and [22], as well as at evaluating variable support-based method on images affected by photometric distortions and noise using stereo pairs with ground-truth.

References

[1] S. Adhyapak, N. Kehtarnavaz, and M. Nadin. Stereo matching via selective multiple windows. *Journ. of Electronic Imaging*, 16(1):013012, 2007. 2

[2] R. Arnold. Automated stereo perception. Technical Report AIM-351, Artificial Intelligence Laboratory, Stanford University, 1983. 1

[3] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. PAMI*, 20(4):401–406, 1998. 2

[4] A. Bobick and S. Intille. Large occlusion stereo. *IJCV*, 33(3):181–200, 1999. 2, 4, 6

[5] Y. Boykov, O. Veksler, and R. Zabih. A variable window approach to early vision. *IEEE Trans. PAMI*, 20(12):1283–1294, 1998. 3, 4, 6

[6] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. PAMI*, 25(8):993–1008, 2003. 1

[7] S. Chan, Y. Wong, and J. Daniel. Dense stereo correspondence based on recursive adaptive size multi-windowing. In *Proc. Image and Vision Computing New Zealand*, volume 1, pages 256–260, 2003. 3, 4, 6

[8] T. Darrel. A radial cumulative similarity transform for robust image correspondence. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 656–662, 1998. 4

[9] C. Demoulin and M. Van Droogenbroeck. A method based on multiple adaptive windows to improve the determination of disparity maps. In *Proc. IEEE Workshop on Circuit, Systems and Signal Processing*, pages 615–618, 2005. 3, 4, 6

[10] A. Fusiello, V. Roberto, and E. Trucco. Symmetric stereo with multiple windowing. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 14(8):1053–1066, 2000. 2

[11] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *IJCV*, 14(3):211–226, 1995. 1

[12] M. Gerrits and P. Bekaert. Local stereo matching with segmentation-based outlier rejection. In *Proc. Conf. Computer and Robot Vision*, pages 66–66, 2006. 2, 3, 4, 5, 6

[13] M. Gong and R. Yang. Image-gradient-guided real-time stereo on graphics hardware. In *Proc. Conf. 3D Digital Imaging and Modeling (3DIM)*, pages 548–555, 2005. 3, 4, 5, 6

[14] M. Gong, R. Yang, W. Liang, and M. Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. *IJCV*, 75(2):283–296, 2007. 1

[15] H. Hirschmuller, P. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 47:1–3, 2002. 3, 4, 6

[16] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 1–8, 2007. 1

[17] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 103–110, 2001. 2, 4, 6

[18] J. Kim, K. Lee, B. Choi, and S. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1075–1082, 2005. 3, 4, 6

[19] M. Levine, D. O'Handley, and G. Yagi. Computer determination of depth maps. *Computer Graphics and Image Processing*, 2:131–150, 1973. 1

[20] S. Mattoccia, F. Tombari, and L. Di Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. In *Proc. Asian Conf. on Computer Vision*, pages 517–527, 2007. 1

[21] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *IJCV*, 7(2):143–162, 1992. 1, 2

[22] M. Okutomi, Y. Katayama, and S. Oka. A simple stereo algorithm to recover precise object boundaries and smooth surfaces. *IJCV*, 47(1-3):261–273, 2002. 3, 7

[23] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, 2002. 1, 2, 4

[24] H. Tao, H. Sawheny, and R. Kumar. A global matching framework for stereo computation. In *Proc. Int. Conf. Computer Vision*, volume 1, pages 532–539, 2001. 3

[25] F. Tombari, S. Mattoccia, and L. Di Stefano. Segmentation-based adaptive support for accurate stereo correspondence. In *Proc. Pacific-Rim Symposium on Image and Video Technology*, 2007. 2, 4, 5, 6

[26] O. Veksler. Stereo matching by compact windows via minimum ratio cycle. In *Proc. Int. Conf. Computer Vision*, volume 1, pages 540–547, 2001. 3, 7

[27] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 556–561, 2003. 2, 4, 5, 6

[28] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In *Proc. Int. Symp. 3D Data Proc., Vis. and Transm. (3DPVT)*, pages 798–805, 2006. 1, 4

[29] Y. Xu, D. Wang, T. Feng, and H. Shum. Stereo computation using radial adaptive windows. In *Int. Conf. Pattern Recognition*, volume 3, pages 595– 598, 2002. 4, 5, 6

[30] K. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. PAMI*, 28(4):650–656, 2006. 2, 4, 5, 6

[31] K. Yoon and I. Kweon. Stereo matching with symmetric cost functions. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 2371 – 2377, 2006. 1

[32] K. Yoon and I. Kweon. Stereo matching with the distinctive similarity measure. In *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007. 2

[33] C. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Trans. PAMI*, 22(7):675–684, 2000. 2

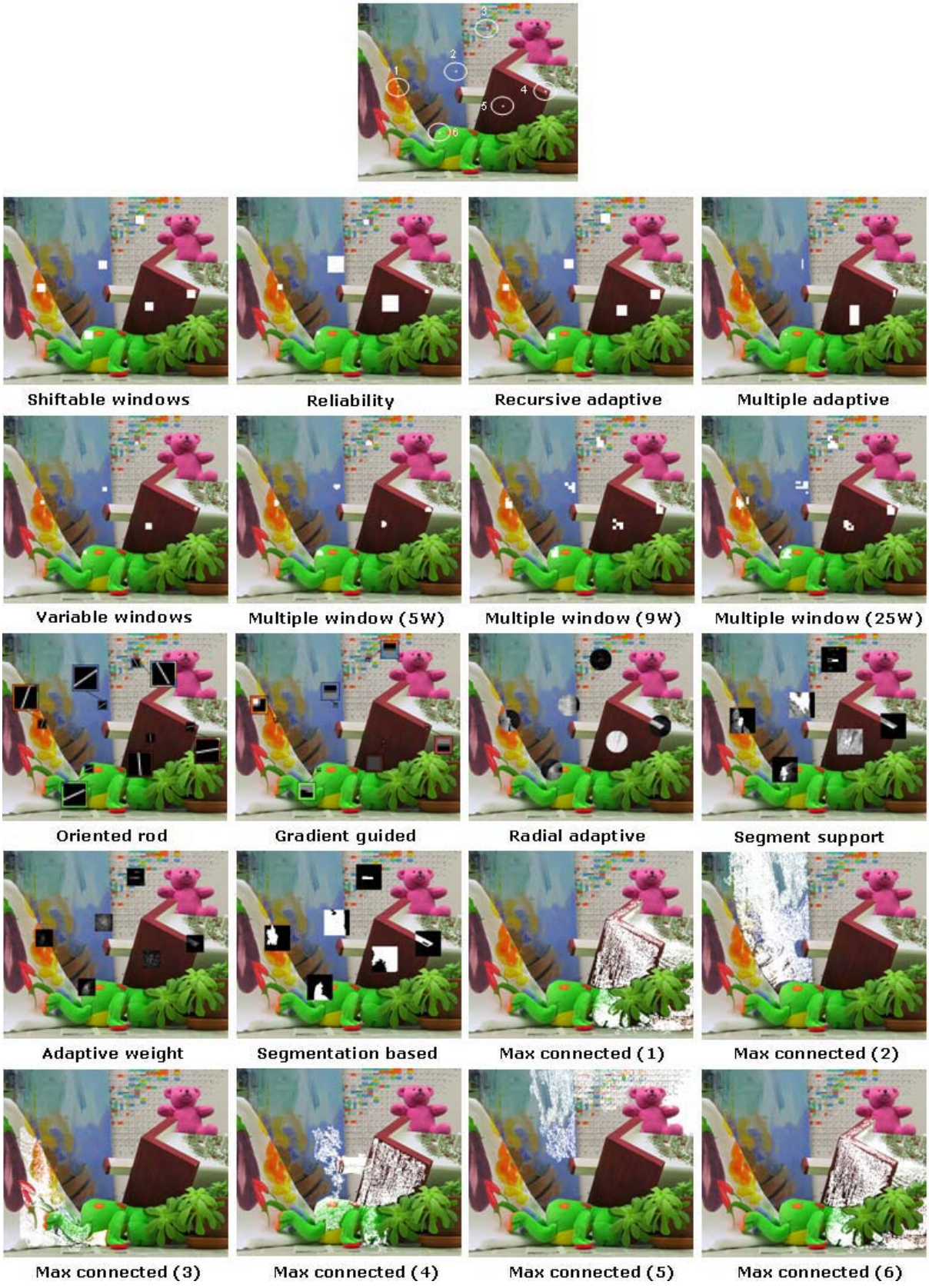

Figure 1. Qualitative comparison of the supports obtained by different strategies on 6 points of stereo pair *Teddy*. The 6 points are depicted