Content-based guided image filtering, weighted semi-global optimization and efficient disparity refinement for fast and accurate disparity estimation

Georgios A. Kordelas, Dimitrios S. Alexiadis, Petros Daras, *Senior Member, IEEE,* and Ebroul Izquierdo, *Senior Member, IEEE*

***Abstract*—This paper presents a novel approach, which relies on content-based guided image filtering and weighted semi-global optimization for fast and accurate disparity estimation. The approach uses a pixel-based cost term that combines gradient, Gabor-Feature and color information. The pixel-based matching costs are filtered by applying guided image filtering, which relies on rectangular support windows of two different sizes. In this way, two filtered costs are estimated for each pixel. Among the two filtered costs, the one that will be finally assigned to each pixel, depends on the local image content around this pixel. The filtered cost volume is further refined by exploiting weighted semi-global optimization, which improves the disparity estimation accuracy. Finally, the disparity refinement in outlier regions relies on a straightforward and time efficient outliers handling scheme and on a simple approach which deals with the disparity outliers at depth discontinuities. Experimental results on the Middlebury online stereo evaluation benchmark and 27 additional Middlebury stereo pairs, prove that our method is able to generate disparity maps with high accuracy while keeping the computational cost low.**

***Index Terms*—stereo vision, stereo matching, disparity estimation, semi-global optimization, guided image filter, disparity refinement, outliers handling**

I. Introduction

Stereo reconstruction is one of the most active research fields in computer vision [1] and it is exploited in a wide range of applications, such as mobile robot navigation [2], augmented reality [3], [4], automotive [5] and telepresence [6], [7] applications . Although various methods have been proposed so far, the estimation of dense disparity maps from stereo image pairs is still a challenging task and there is further space for improving accuracy, minimizing the computational cost and handling more efficiently occlusions, textureless areas and light variations. Section I-A reports on existing methods in the field. Paper's contribution is described in section I-B. While, section I-C highlights the main differences of the proposed method with respect to other state-of-the-art methods.

A. Review of previous work

The work in [1] presents a complete taxonomy of approaches used for stereo disparity estimation. The categorization of the approaches is based on the following four generic steps, into which most of the stereo algorithms can be decomposed: 1. matching cost computation; 2. cost aggregation; 3. disparity computation/optimization; and 4. disparity refinement.

The matching cost computation step is based on the utilization of a matching metric, which is usually formed as a combination of individual pixel-based cost measures. Pixel-based cost measures include the absolute difference of image intensity values [8], gradient-based measures [8], Gabor-feature-based measures [9] and non-parametric transforms such as CENSUS [10]. Disparity estimation approaches, which use combinations of individual cost measures in order to form a final cost metric that inherits the advantageous characteristics of each measure, have been proposed. In specific, the works in [11], [12], [13] exploit a combination of absolute intensity differences, as well as the hamming distance of CENSUS transform coefficients. The cost term used in [14], [15] combines absolute intensity differences and a gradient based measure. The work in [16] uses a combination of CENSUS, color and gradient based cost measures.

The matching cost values over all pixels and all candidate disparities form the initial cost volume. In order to reduce matching ambiguity, the pixel-based matching costs are locally aggregated in the initial cost volume. The performance evaluation on different cost aggregation approaches, which was presented in [17], shows that until 2008, Adaptive support weight [18] and Segment-support [19] approaches outperformed the rest of cost aggregation approaches. The adaptive support weight method [18] adjusts the weights based on color similarity and proximity principles. In the segment-support approach [19], [20], pixels that belong to the same segment as the support window's central pixel are assigned a weight equal to 1, while the remaining ones are given a weight based on their color similarity to the central pixel. Despite their good disparity accuracy, the main drawback of [18] and [19] is their high computational cost.

Cost aggregation methods that use a support window with variable size and/or shape, adaptive to the image content, can also be found in the literature. In [21] a fast method, where an upright cross local support skeleton is adaptively constructed

Georgios A. Kordelas is with CERTH/Information Technologies Institute, Greece and with School of Electronic Engineering and Computer Science, Queen Mary University of London, UK (e-mail:kordelas@iti.gr)

Dimitrios S. Alexiadis and Petros Daras are with the Information Technologies Institute, Centre for Research and TechnologyHellas, 6th km CharilaouThermi, GR-57001, Thessaloniki, Greece (e-mail: dalexiad@iti.gr; daras@iti.gr)

Ebroul Izquierdo is with the Electronic Engineering and Computer Science department, Queen Mary University of London, UK (e-mail:ebroul.izquierdo@eecs.qmul.ac.uk)

for each anchor pixel, is presented. Then, given the local cross-decision results, a shape-adaptive full support region is dynamically constructed by merging horizontal segments of the crosses in the vertical neighborhood.

In recent years, several approaches perform cost aggregation by filtering the initial cost volume. For instance, the work in [22] proposes a recursive implementation of the bilateral filter [23], where the computational complexity is linear in both input size and dimensionality. Recently, the use of more efficient filters has been proposed. The edge preserving guided image filter [24] has been exploited in [15] and [25]. While, the constant weighted median filter has been proposed and exploited in [26].

The disparity optimization step includes local and global methods. Local methods [12], [15], [18], [19], [21], [25], [27] give emphasis on the matching cost computation and the cost aggregation steps. On the other hand, global optimization methods aim at assigning a disparity label to each pixel, so that a global cost function is minimized over the whole image area. Efficient global optimization techniques include Graph Cuts [28], Belief Propagation [14] and cooperative optimization [29]. Several works propose approaches for reducing the computational cost of global methods. For instance the work in [30] proposes a hierarchical bilateral disparity structure approach in order to reduce the computational complexity and slightly improve the accuracy of the Graph Cuts algorithm.Local methods usually fail on ambiguous low texture areas, while global methods, although more accurate, require much higher processing time. Semi-global optimization approaches [13], [20], [31], [32] incorporating advantages of both groups, provide a good compromise between complexity and accuracy.

The disparity results have to be refined, since they are "polluted" with outliers in occluded areas, uniform areas and depth discontinuities. Several stereo algorithms, such as those in [31], [33], use segmented regions for reliable outlier handling. The work in [13] uses iterative region voting and proper interpolation to fill outliers.

Stereo vision methods, except for still stereo pairs, can also use as input stereo video sequences. Several works, implemented in GPU or FPGA, are able to generate disparity information from stereo videos in real-time. The method in [13] is able to generate disparity information from low-resolution video at a rate of 10 frames per second (fps). The work in [34], which describes a FPGA implementation architecture of a semi-global matching method, is able to generate disparity maps from VGA images at a rate of 30 fps. The paper in [35] presents a hardware-oriented disparity estimation algorithm that uses iterative refinement. The implementation of [35] in FPGA can process XGA images at a rate of 60 fps.

B. Contributions of this paper

In this paper, a methodology for fast and accurate dense disparity estimation is proposed. Most significant contributions of this work include:

- A novel strategy for exploiting guided image filtering [24]. In brief, the guided image filtering is applied separately for support windows of two different sizes and the appropriate support window size for each pixel is selected based on the texture homogeneity within the local region around this pixel. The texture homogeneity is examined by exploiting the mean-shift segmentation maps [36] of the stereo pair.
- An innovative weighted variant of the semi-global optimization method of [31], where the path costs of a pixel may have different weights, depending on the pixels that precede this pixel along each path direction. Furthermore, in the proposed variant, possible depth discontinuities are identified according to an adaptive threshold that depends on the intensity of the examined pixel. These two features improve the overall performance of the original approach.

Additional secondary contributions include: a) an efficient matching cost metric that can be rapidly estimated, which combines horizontal gradients, Gabor features and a sampling-insensitive dissimilarity measure; b) a disparity refinement approach that comprises: i) a simple outliers-handling scheme, which examines whether the pixels on the right or the left side of an outlier pixel are more similar in terms of color to that pixel, before assigning a disparity value to it and ii) an efficient technique for correcting disparity outliers at depth discontinuities.

This work, by encompassing the aforementioned contributions, manages to generate quite fast disparity maps of high accuracy, as it is verified in section III.

C. Proposed methodology in comparison to state-of-the-art methods

Several methods require iteration cycles in order to improve gradually the accuracy of the estimated disparity maps [29], [37], [38], [39]. Consequently, the number of iterations affects the computational cost of an approach. On the contrary, the proposed method gives accurate disparity results without performing any repetitive refinement.

Plenty of methods, such as [14], [29], [40], [41], exploit image segmentation algorithms to separate images into segments and then solve the disparity estimation problem by assigning a disparity plane to each estimated segment. In contrast to this class of approaches, the proposed method does not require plane fitting to give accurate disparity results.

Many methods, such as [9], [13], [14], [19], [20], [21], [29], [37], [38], are evaluated using just the four stereo pairs of the Middlebury Stereo Online Evaluation Benchmark and some of them [13], [14], [29] manage to rank among the top methods. In this work, for more thorough evaluation, 27 additional stereo pairs for assessing the overall performance are used.

The rest of this paper is organized as follows. In section II, the proposed method is presented in detail. Section III provides information on the parameters used and presents the experimental results, while conclusions are drawn in section IV.

II. Proposed Method

An overview of the proposed method is given in the schematic flow diagram of Fig. 1. As depicted in the diagram,

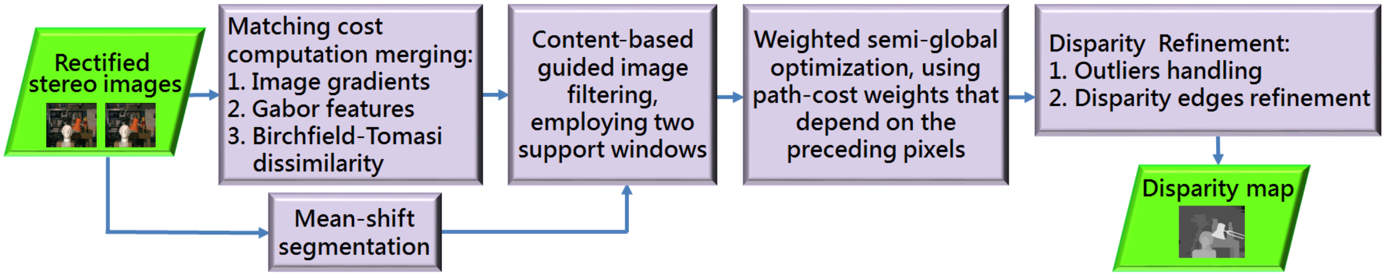

Rectified stereo images
Matching cost computation merging:
1. Image gradients
2. Gabor features
3. Birchfield-Tomasi dissimilarity
Content-based guided image filtering, employing two support windows
Weighted semi-global optimization, using path-cost weights that depend on the preceding pixels
Disparity Refinement:
1. Outliers handling
2. Disparity edges refinement
Mean-shift segmentation
Disparity map

Fig. 1: Schematic flow diagram of the proposed method.

Mean-Shift segmentation is applied to the input stereo images as a pre-processing step of the algorithm, necessary for several of the subsequent steps. The core algorithm is then divided into four sequentially applied components: a) Matching cost computation, based on the image gradient, gabor features and the RGB information (via the Birchfield-Tomasi dissimilarity measure); b) Cost filtering using content-based guided image filtering that is applied for rectangular support windows of two different sizes; c) Disparity optimization via weighted semi-global optimization, where the path costs of a pixel may have different weights depending on the pixels that precede the considered pixel along a path direction; d) Disparity Refinement relying on: i) outliers handling based on background and generic outliers handling and ii) disparity edges refinement using disparity histograms. These algorithmic steps are analyzed in sections II-B through II-D.

A. Preprocessing step

a) Mean-shift segmentation: In this work, the stereo images are initially segmented into non-overlapping regions by running the Mean Shift algorithm [36], which relies on color and edge information. The parameters used for the mean-shift segmentation are the segmentation spatial radius σ_s, which is set to $\sigma_s = 3$ and the segmentation feature space radius σ_R, which is set to $\sigma_R = 3$. The selection of these strict values ensures that the segmentation map will be of high reliability, meaning that most likely a segment will not overlap a depth discontinuity, and this fact is also verified in [19] and [42]. Except for these strict values, other sets of segmentation parameters have been also tested in subsection III-B4. The segmentation maps of the left and the right image are computed once and then used in the subsequent algorithmic steps. As an example of mean-shift segmentation, the segmentation map for the “Tsukuba” left image is visualized in Fig. 3a.

B. Content-based guided image filtering

1) Matching cost computation: This subsection describes the definition of the combined matching cost term, which is used for the computation of the initial cost volume. The initial cost volume is then filtered relying on content-based guided image filtering.

Let I_l^c and I_r^c be the left and right color images of the stereo pair, while I_l and I_r are their respective grayscale images. Given a pixel $\mathbf{p}$ on the left image (reference image), the corresponding pixel on the right image (target image) for a candidate disparity value d is denoted as $\mathbf{p}^d$. This step defines a matching cost metric for estimating the similarity between two pixels. The proposed cost metric is composed of three individual pixel-based cost terms: (i) a gradient-based cost term, (ii) a Gabor-Feature-Image based term [9] and (iii) a Birchfield-Tomasi dissimilarity term [43].

The gradient-based cost term for a pixel $\mathbf{p}$ and disparity d is given by:

$$C_{\text{gra}}(\mathbf{p}, d) = |\nabla_{\text{H}}(I_l(\mathbf{p})) - \nabla_{\text{H}}(I_r(\mathbf{p}^d))|, \tag{1}$$

where $\nabla_{\text{H}}(I(\mathbf{p}))$ denotes the gradient in horizontal direction at pixel $\mathbf{p}$ on the grayscale image I.

The second term, as in [9], is based on the Gabor-Feature-Image, which is extracted after applying a Gabor filter on an image. Let $G_{\text{H}}(I_l(\mathbf{p}))$ and $G_{\text{H}}(I_r(\mathbf{p}^d))$ denote the outputs of the vertically-varying Gabor kernel (detection of horizontal features) for I_l and I_r, respectively (see the supplementary appendix for more details). The cost term $C_{\text{gab}}(\mathbf{p}, d)$ for pixel $\mathbf{p}$ at disparity d is given by:

$$C_{\text{gab}}(\mathbf{p}, d) = |G_{\text{H}}(I_l(\mathbf{p})) - G_{\text{H}}(I_r(\mathbf{p}^d))|. \tag{2}$$

The third term is given by:

$$C_{\text{BT}}(\mathbf{p}, d) = \sum_{\text{c=R,G,B}} \frac{D^{\text{c}}(\mathbf{p}, \mathbf{p}^d)}{3}, \tag{3}$$

where $D^{\text{c}}(\mathbf{p}, \mathbf{p}^d)$ is the Birchfield-Tomasi dissimilarity measure between pixels $\mathbf{p}$ and $\mathbf{p}^d$ [43] (see the supplementary appendix for more details on the Birchfield-Tomasi dissimilarity measure).

The combined matching cost term, merging Eq. (1), Eq. (2) and Eq. (3) is expressed as:

$$\begin{aligned} C(\mathbf{p}, d) = {} & \alpha_1 \cdot \min\left(C_{\text{gra}}(\mathbf{p}, d), T_{\text{gra}}\right) + \\ & \alpha_2 \cdot \min\left(C_{\text{gab}}(\mathbf{p}, d), T_{\text{gab}}\right) + \\ & (1 - \alpha_1 - \alpha_2) \cdot \min\left(C_{\text{BT}}(\mathbf{p}, d), T_{\text{BT}}\right) \end{aligned} \tag{4}$$

where α_1, α_2 are balance weights and T_{gra}, T_{gab}, T_{BT} are truncation thresholds. Experiments on the selection of optimal values for balance weights α_1, α_2 in Eq. (4) are given in subsection III-A. The reasons for using these three terms to compute $C(\mathbf{p}, d)$ are the following:

- The gradient-based cost term shows high robustness to illumination changes, has strong local minima and can be estimated very fast [8].
- The Gabor-Feature-Image, according to [9] is appropriate for texture representation and discrimination, robust to

illumination changes, insensitive to image noise and can be calculated quite fast.

- The Birchfield-Tomasi dissimilarity measure, presented in [43], is insensitive to image sampling and can be estimated fast at the same time.

The combined cost term of Eq. (4), which was firstly introduced and evaluated in our preceding work presented in [44], is inspired by previous works in the literature. In more detail, the horizontal gradient term and the absolute color difference term have been combined in [9], [15] to estimate the initial cost volume. An additional Gabor-based term, which helps to improve the final disparity estimation results, is used in [9]. Our prior work in [44] proposes a modification of the combined matching cost term of [9]. In specific, it proposes the replacement of the absolute color difference term with the Birchfield-Tomasi dissimilarity term. The modified combined matching cost term leads to better disparity estimation results, than using the one of [9], as it is experimentally verified in [44]. The Gabor-feature term is complementary to the horizontal gradient term, since it performs edge detection in the vertical direction. Therefore, edge information on both horizontal and vertical direction is exploited in the combined matching cost term. On the other hand, the Birchfield-Tomasi dissimilarity term is used to exploit the color information, complementing the horizontal gradient and the Gabor-feature terms that rely on the gradient information and not on the color information. In the supplementary appendix, the combined matching cost of Eq. (4) is compared against the combined matching costs proposed in [9], [15], [25] and is proved that the use of Eq. (4) helps to acquire more accurate disparity maps.

The initial cost volume $C(\mathbf{p}, d)$ is a three dimensional array which stores the matching costs for all pixels and all possible disparity candidates. The initial disparity map of Fig. 2 is acquired after applying Winner-Take-All (WTA) to $C(\mathbf{p}, d)$, i.e. selecting for a pixel $\mathbf{p}$ the disparity d that minimizes $C(\mathbf{p}, d)$. The initial disparity map of Fig. 2 is heavily corrupted by estimation-error noise.

2) Guided image filtering: In order to reduce matching ambiguity that results to noisy disparity maps, the matching costs $C(\mathbf{p}, d)$ are filtered over support windows using the guided image filter [15]. In detail, the filtered cost value of pixel $\mathbf{p}$ at a fixed disparity d is given by:

$$C^{'}(\mathbf{p}, d) = \sum_{\mathbf{q}} W(\mathbf{p}, \mathbf{q}) C(\mathbf{p}, d), \tag{5}$$

where the filter weights $W(\mathbf{p}, \mathbf{q})$ depend on the color guidance image I. The reference (left) stereo image is used as the "guidance image", which is referred to as such because its content influences ("guides") the filtering of the matching costs. These weights are given from [15]:

$$W(\mathbf{p}, \mathbf{q}) = \frac{1}{|w_{\mathbf{k}}|^2} \sum_{(\mathbf{p},\mathbf{q}) \in w_{\mathbf{k}}} \Big(1 + (I(\mathbf{p}) - \mu_{\mathbf{k}})^T (\Sigma_{\mathbf{k}} + \varepsilon U)^{-1} (I(\mathbf{q}) - \mu_{\mathbf{k}})\Big), \tag{6}$$

where $|w_{\mathbf{k}}|$ is the total number of pixels in a support window $w_{\mathbf{k}}$ centered at pixel $\mathbf{k}$ and ε is a smoothness parameter. $\Sigma_{\mathbf{k}}$ and $\mu_{\mathbf{k}}$ are the covariance and the mean of pixels colors within $w_{\mathbf{k}}$. $I(\mathbf{p})$, $I(\mathbf{q})$ and $\mu_{\mathbf{k}}$ are 3×1 (color) vectors, while $\Sigma_{\mathbf{k}}$ and the unary matrix U are of size 3×3.

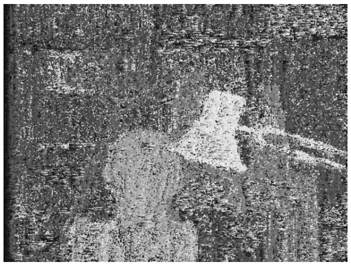


Fig. 2: Initial disparity map.

The selection of the appropriate support window size for each pixel, based on its local image content, is discussed in the next subsection. With the term "local image content" of a pixel, we refer to the image content within a local region around this pixel.

3) Selection of the window size based on local image content: This subsection proposes a novel scheme for exploiting guided image filtering. First of all, the shape of the support window is selected to be rectangular and the largest dimension of the support window to be the horizontal one (width). The window's width is twice its height. A support window elongated along the horizontal dimension, i.e. along the dimension in which disparity varies, is used in order to increase the discriminating ability of the window. This fact is experimentally verified in subsection III-B2.

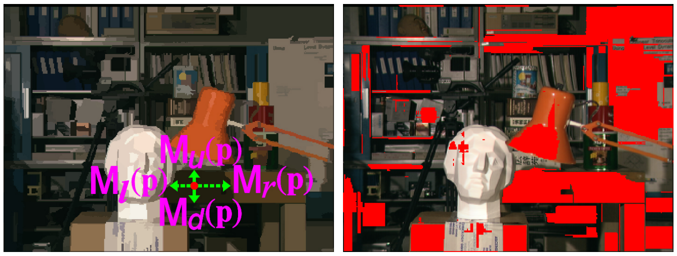
Except for the rectangular shape, in our solution, windows of two sizes are used. The small window size is R_{S} x $\lceil R_{\mathrm{S}}/2 \rceil$ and the large one is $2R_{\mathrm{S}}$ x R_{S}. The guided image cost filtering is performed separately for both window sizes. Given the two filtered costs that were estimated after applying guided image filtering for both window sizes, the filtered cost that is finally assigned to a pixel, depends on the local image content around this pixel. In specific, the preferred support window size for each pixel is selected according to the information about the texture homogeneity within the local region around the pixel. Hence, if the neighborhood around a pixel is homogeneous, then the large support window size, which contains more information, shall be preferred. The criterion to decide which support window is appropriate for a pixel depends on image's segmentation map, which provides information about the homogeneity of the image region around a pixel.

In detail, based on image's segmentation map, for each pixel $\mathbf{p}$ the lengths of its "arms" stretching to the left (F_l), right (F_r), up (F_u) and down (F_d) directions are estimated as visualized in Fig. 3a: Given a pixel $\mathbf{p}$ and a direction F_i, $i \in (l, r, u, d)$, the arm length of $\mathbf{p}$ along the considered direction is given by the number of pixels between $\mathbf{p}$ and the end of the segment where $\mathbf{p}$ belongs. The arm length is denoted as $\mathrm{M}_i(\mathbf{p})$, $i \in (l, r, u, d)$.

The average length of the arms is given by:

$$\bar{\mathrm{M}}(\mathbf{p}) = \Big(\mathrm{M}_l(\mathbf{p}) + \mathrm{M}_r(\mathbf{p}) + \mathrm{M}_u(\mathbf{p}) + \mathrm{M}_d(\mathbf{p})\Big)/4 \qquad (7)$$

If $\bar{\mathrm{M}}(\mathbf{p}) > R_{\mathrm{S}}$ then $\mathbf{p}$ is assumed to lie inside a homogeneous area. Hence, the large window of size $2R_{\mathrm{S}}$ x R_{S} is considered as the appropriate support window for $\mathbf{p}$, in order to contain more information. For pixels with $\bar{\mathrm{M}}(\mathbf{p}) \leq R_{\mathrm{S}}$, the appropriate support window is the one with size R_{S} x $\lceil R_{\mathrm{S}}/2 \rceil$. In Fig. 3b the pixels for which the appropriate support window has size $2R_{\mathrm{S}}$ x R_{S} are visualized with red.



(a) (b)

Fig. 3: (a) Arms lengths for a pixel $\mathbf{p}$ on a segmentation map and (b) Pixels with support region of $2R_{\mathrm{S}}$ x R_{S}

Let the filtered cost using the small window be denoted as $C_1^{'}(\mathbf{p}, d)$, while $C_2^{'}(\mathbf{p}, d)$ denotes the filtered cost using the large one. The final filtered cost $C^{'}(\mathbf{p}, d)$ at pixel $\mathbf{p}$ is set equal to the filtered cost that corresponds to the support window size that is appropriate for this pixel.

In subsection III-B2 the selection of rectangular support windows of two sizes is experimentally justified. Provably, more than two window sizes could be used. However, this would increase the computational cost of the algorithm. Moreover, two window sizes are enough to achieve high disparity estimation accuracy.

4) Comparison with related State of the art methods: A relevant work that uses adaptive guided image filtering is presented in [25]. However, there are significant differences between [25] and the proposed method regarding the selection of the support window for each pixel. In [25] the support window for each pixel is based on a skeleton that is built from four arms stretching in four directions, where the borders of the support window are determined directly by the endpoints of the arms. Therefore, for each pixel, a different support window is used, resulting into increased computational effort. On the contrary, our method that uses two window sizes, is expected to perform fast, as explained directly below.

A main advantage of the guided filter is that the computation cost is independent to the size of the selected support window [24]. This is because Eq. (5) can be expressed as a linear transform as follows:

$$C^{'}(\mathbf{p}, d) = \frac{1}{|w_{\mathbf{k}}|} \sum_{\mathbf{p} \in w_{\mathbf{k}}} (a_{\mathbf{k}} I(\mathbf{p}) + b_{\mathbf{k}}), \qquad (8)$$

where:

$$a_{\mathbf{k}} = (\Sigma_{\mathbf{k}} + \varepsilon U)^{-1} \left(\frac{1}{|w_{\mathbf{k}}|} \sum_{\mathbf{p} \in w_{\mathbf{k}}} I(\mathbf{p}) C(\mathbf{p}, d) - \mu_{\mathbf{k}} \overline{C}(\mathbf{k}, d) \right), \qquad (9)$$

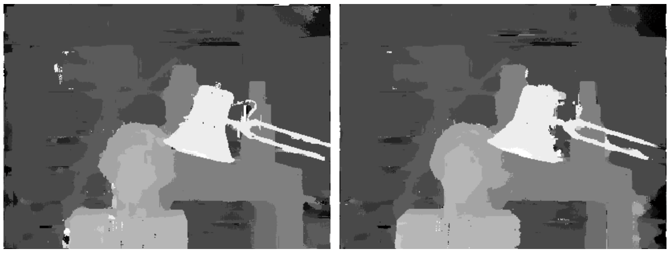
$$b_{\mathbf{k}} = \overline{C}(\mathbf{k}, d) - a_{\mathbf{k}}^{\mathrm{T}} \mu_{\mathbf{k}}. \qquad (10)$$

Here $\overline{C}(\mathbf{k}, d)$ is the mean of the d-th slice of C within $w_{\mathbf{k}}$. Moreover, a factor that increases the speed of the guided image filter is that the summations in equations Eq. (8), Eq. (9) and Eq. (10) can be computed using box filters with a fixed window size [24]. Our method runs the guided image filtering for two fixed support windows sizes, therefore it can use box filters. On the other hand, the method in [25] that uses support windows of random sizes needs to estimate the summations for each pixel separately, an operation that increases the computational cost of [25].

In the experimental results section (see subsection III-B4), evaluation results using our methodology with one modification, are provided. Instead of using our scheme for performing guided image filtering, the scheme from [25] is used. The results show that the exploitation of our scheme within the proposed methodology leads to better disparity estimation results than using the scheme from [25].

C. Disparity optimization relying on weighted semi-global optimization

1) Outliers detection: The left disparity map $d_{\mathrm{LR}}(\mathbf{p})$ (Fig. 4a) is acquired after applying Winner-Take-All (WTA) to the cost volume $C^{'}(\mathbf{p}, d)$, i.e. selecting for a pixel $\mathbf{p}$ the disparity d that minimizes $C^{'}(\mathbf{p}, d)$. If the right image is considered as reference, then the disparity map $d_{\mathrm{RL}}(\mathbf{p})$ of Fig. 4b is acquired. The computation of $d_{\mathrm{LR}}(\mathbf{p})$ and $d_{\mathrm{RL}}(\mathbf{p})$ is fully independent. The disparity maps $d_{\mathrm{LR}}(\mathbf{p})$ and $d_{\mathrm{RL}}(\mathbf{p})$ are taken into consideration to detect problematic areas, especially outliers in occluded regions and depth discontinuities. A prevalent strategy for detecting outliers is the Left-Right consistency check [20].



(a) (b)

Fig. 4: (a) $d_{\mathrm{LR}}(\mathbf{p})$ and (b) $d_{\mathrm{RL}}(\mathbf{p})$ disparity maps

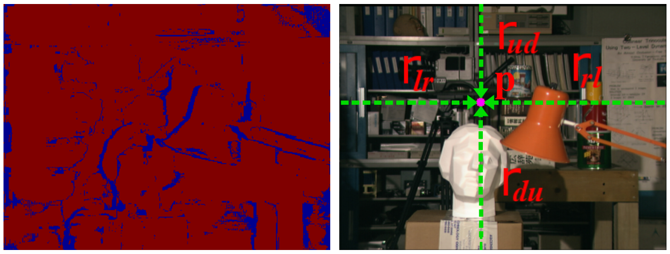
In this strategy, the outlier pixels have disparity values that are not consistent between the two disparity maps ($d_{\mathrm{LR}}(\mathbf{p})$ and $d_{\mathrm{RL}}(\mathbf{p})$) and therefore, they do not satisfy the relation:

$$|d_{\mathrm{LR}}(\mathbf{p}) - d_{\mathrm{RL}}(\mathbf{p} - d_{\mathrm{LR}}(\mathbf{p}))| \leq T_{\mathrm{LR}}, \qquad (11)$$

where $\mathbf{p}$ is the location of the considered pixel. The threshold for the outliers detection in Eq. (11) is set equal to $T_{\mathrm{LR}} = 0$. Fig. 5a shows the outliers map $\mathrm{O}_1^{T_{\mathrm{LR}}}(\mathbf{p})$ that is generated for $T_{\mathrm{LR}} = 0$. The blue regions in $\mathrm{O}_1^{T_{\mathrm{LR}}}(\mathbf{p})$ denote the outlier pixels for which the relation in Eq. (11) does not hold, while the red regions denote the inlier pixels, i.e. pixels for which Eq. (11) holds.

2) Weighted Semi-global Optimization: The optimization of the filtered cost volume $C^{'}(\mathbf{p}, d)$ is based on the semi-global optimization method of [31], which aggregates matching costs in 1D from multiple path directions. While the original semi-global method aggregates matching costs equally from multiple direction, in our approach a weighted aggregation, which improves the disparity estimation results, is used.

This work considers four path directions $\mathbf{r}$, namely left-to-right, right-to-left, up-to-down and down-to-up, which are denoted as $\mathbf{r}_{lr} = [+1, 0]^{\mathbf{T}}$, $\mathbf{r}_{rl} = [-1, 0]^{\mathbf{T}}$, $\mathbf{r}_{ud} = [0, +1]^{\mathbf{T}}$ and $\mathbf{r}_{du} = [0, -1]^{\mathbf{T}}$, respectively (see Fig. 5b).



(a)

(b)

Fig. 5: (a) Outliers map $\mathrm{O}_1^{T_{\mathrm{LR}}}(\mathbf{p})$ for threshold $T_{\mathrm{LR}} = 0$ and (b) Path directions used for cost volume optimization.

Let $L_{\mathbf{r}}$ be a path that is traversed in the direction $\mathbf{r} \in \{\mathbf{r}_{lr}, \mathbf{r}_{rl}, \mathbf{r}_{ud}, \mathbf{r}_{du}\}$. The path cost $L_{\mathbf{r}}(\mathbf{p}{,}d)$ of pixel $\mathbf{p}$ at disparity d is computed recursively from:

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}{,}d) = C^{'}(\mathbf{p}, d) + \min\Big\{ & L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}{,}d), L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}{,}d \pm 1) + \\ & \pi_1(\mathbf{p}),\ \min_{d_{\mathrm{i}}} L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}{,}d_{\mathrm{i}}) + \pi_2(\mathbf{p}) \Big\} - \\ & \min_{d_{\mathrm{i}}} L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}{,}d_{\mathrm{i}}) \end{aligned} \tag{12}$$

where $d_{\mathrm{i}} \in [\mathit{disparity\ range}]$ and **p - r** denotes the previous pixel along the path direction. $\pi_1(\mathbf{p})$ and $\pi_2(\mathbf{p})$ are two smoothness penalty terms (with $\pi_1(\mathbf{p}) \leq \pi_2(\mathbf{p})$) for penalizing disparity changes of neighboring pixels.

In more detail, in Eq. 12, $C^{'}(\mathbf{p}, d)$ is the filtered cost value of pixel $\mathbf{p}$ at disparity d (see Eq. 5), while the second term of the equation adds the lowest path cost of the previous pixel **p - r** of the path, including the appropriate smoothness penalty terms $\pi_1(\mathbf{p})$ and $\pi_2(\mathbf{p})$ for penalizing disparity discontinuities. Finally, the minimum path cost $\min\limits_{d_{\mathrm{i}}} L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}{,}d_{\mathrm{i}})$ of the previous pixel is subtracted from the whole term, so that the values of $L_{\mathbf{r}}$ do not permanently increase along the path. More details regarding Eq. 12 can be found in [31].

The smoothness penalty terms $\pi_1(\mathbf{p})$ and $\pi_2(\mathbf{p})$ are defined according to:

$$\Big(\pi_1(\mathbf{p}), \pi_2(\mathbf{p})\Big) = \begin{cases} (\Pi_1, \Pi_2), \text{ if } \Big(\nabla(\mathbf{p}) \leq \tau_l(I_l(\mathbf{p}))\ \& \\ \qquad \nabla(\mathbf{p}^d) \leq \tau_r(I_r(\mathbf{p}^d))\Big) \\ \left(\frac{\Pi_1}{4}, \frac{\Pi_2}{4}\right), \text{ if } \Big(\nabla(\mathbf{p}) \leq \tau_l(I_l(\mathbf{p}))\ \& \\ \qquad \nabla(\mathbf{p}^d) > \tau_r(I_r(\mathbf{p}^d))\Big) \\ \left(\frac{\Pi_1}{4}, \frac{\Pi_2}{4}\right), \text{ if } \Big(\nabla(\mathbf{p}) > \tau_l(I_l(\mathbf{p}))\ \& \\ \qquad \nabla(\mathbf{p}^d) \leq \tau_r(I_r(\mathbf{p}^d))\Big) \\ \left(\frac{\Pi_1}{10}, \frac{\Pi_2}{10}\right), \text{ otherwise,} \end{cases} \tag{13}$$

where $\Pi_1 = 0.002$, $\Pi_2 = 0.006$ are constant parameters. $\nabla(\mathbf{p})$ and $\nabla(\mathbf{p}^d)$ are the intensity differences between a considered pixel and the preceding one along the path direction, on the two images, respectively, and they are defined as:

$$\nabla(\mathbf{p}) = |I_l(\mathbf{p}) - I_l(\mathbf{p} - \mathbf{r})| \tag{14}$$

and

$$\nabla(\mathbf{p}^d) = \left|I_r(\mathbf{p}^d) - I_r(\mathbf{p}^d - \mathbf{r})\right|, \tag{15}$$

where I_l and I_r are the images in grayscale. The idea of using $\nabla(\mathbf{p})$ and $\nabla(\mathbf{p}^d)$ is based on the assumption that a disparity change (depth discontinuity) often coincides with an intensity edge [13], [20]. Therefore, the intensity difference between neighboring pixels is able to indicate the presence of an intensity edge.

The thresholds τ_l and τ_r in Eq. 13 are adaptive to $I_l(\mathbf{p})$ and $I_r(\mathbf{p}^d)$, respectively. The rationale behind using adaptive thresholds is that for areas with low intensity it is more difficult to discriminate regions that may belong to different depths, while for areas with high intensity this discrimination is more evident. Therefore, the intensity threshold τ, which denotes a depth discontinuity, should be low for a low intensity pixel and increase as the intensity of the considered pixel increases. More details on how the adaptive intensity threshold τ is defined are given in the following paragraph.

A fixed intensity threshold for identifying disparity edges, equal to $10/255$ (the pixel intensity range is [0, 1]), is used in [20]. The proposed methodology proposes an adaptive threshold $\tau(I(\mathbf{q}))$ whose value varies around the average value of $10/255$ depending on the intensity $I(\mathbf{q})$ of the considered pixel $\mathbf{q}$. The adaptive intensity threshold has a minimum value of $5/255$ and maximum value of $15/255$, so that its range is tight around the value of $10/255$. The adaptive threshold $\tau(I(\mathbf{q}))$ is set to $5/255$ when $I(\mathbf{q}) < 30/255$, while $\tau(I(\mathbf{q}))$ is set to $15/255$ when $I(\mathbf{q}) \geq 210/255$. For $30/255 \leq I(\mathbf{q}) < 210/255$ the adaptive threshold $\tau(I(\mathbf{q}))$ increases linearly with $I(\mathbf{q})$, from its minimum value $5/255$ to its maximum value $15/255$.

Based on the above, the equation of the intensity-based adaptive threshold $\tau(I(\mathbf{q}))$ is given from:

$$\tau(I(\mathbf{q})) = \begin{cases} 5/255, \text{ if } \ I(\mathbf{q}) < 30/255 \\ \left(\left(\frac{10}{180}\right) \cdot (I(\mathbf{q}) \cdot 255 - 30) + 5\right)/255, \text{ if } 30/255 \leq I(\mathbf{q}) < 210/255 \\ 15/255, \text{ if } \ I(\mathbf{q}) \geq 210/255, \end{cases} \tag{16}$$

while the graphical representation of Eq. (16) is displayed in Fig. 6.

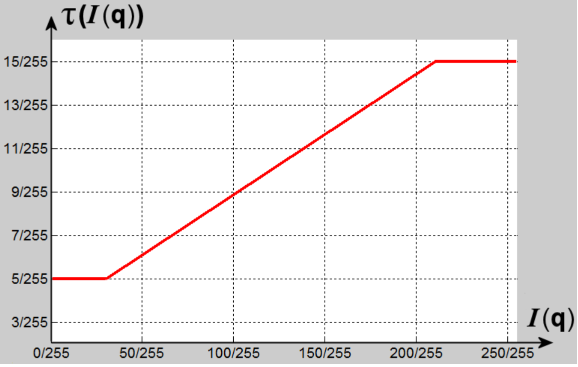


Fig. 6: Graphical representation of $\tau(I(\mathbf{q}))$.

After computing the four path costs $(L_{\mathbf{r}_{lr}}, L_{\mathbf{r}_{rl}}, L_{\mathbf{r}_{ud}}, L_{\mathbf{r}_{du}})$ using Eq. (12), the final cost volume $C''(\mathbf{p},d)$ is calculated from:

$$C''(\mathbf{p},d) = \frac{1}{4}\cdot\Big[w_{lr}(\mathbf{p})\cdot L_{\mathbf{r}_{lr}}(\mathbf{p},d)+w_{rl}(\mathbf{p})\cdot L_{\mathbf{r}_{rl}}(\mathbf{p},d)+ \\ + w_{ud}(\mathbf{p})\cdot L_{\mathbf{r}_{ud}}(\mathbf{p},d)+w_{du}(\mathbf{p})\cdot L_{\mathbf{r}_{du}}(\mathbf{p},d)\Big] \qquad (17)$$

where $w_{lr}(\mathbf{p})+w_{rl}(\mathbf{p})+w_{ud}(\mathbf{p})+w_{du}(\mathbf{p})=4$.

In the original approach of the semi-global optimization [31]: $w_{lr}(\mathbf{p}) = w_{rl}(\mathbf{p}) = w_{ud}(\mathbf{p}) = w_{du}(\mathbf{p}) = 1$, while in our approach these weights may not be equal. Practically, if along a path direction, the non-outlier pixels that belong to the same surface as the considered pixel $\mathbf{p}$, are much more than the non-outliers pixels of other directions, we assume that this direction should get a higher weight since it will give more accurate estimates. Therefore, for a pixel $\mathbf{p}$ and a specific direction, the total number of non-outlier pixels that precede $\mathbf{p}$ along this direction and at the same time they belong to the same surface as $\mathbf{p}$, is computed. The total number of non-outlier pixels for directions $\mathbf{r}_{lr}, \mathbf{r}_{rl}, \mathbf{r}_{ud}$ and $\mathbf{r}_{du}$ for pixel $\mathbf{p}$ is denoted as $\mathrm{M}_l^{'}(\mathbf{p})$, $\mathrm{M}_r^{'}(\mathbf{p})$, $\mathrm{M}_u^{'}(\mathbf{p})$ and $\mathrm{M}_d^{'}(\mathbf{p})$, respectively. $\mathrm{M}_l^{'}(\mathbf{p})$, $\mathrm{M}_r^{'}(\mathbf{p})$, $\mathrm{M}_u^{'}(\mathbf{p})$ and $\mathrm{M}_d^{'}(\mathbf{p})$ are computed as described in the next paragraph.

Let that the arms lengths of a pixel $\mathbf{p}$, as estimated in subsection II-B3, are $\mathrm{M}_l(\mathbf{p})$, $\mathrm{M}_r(\mathbf{p})$, $\mathrm{M}_u(\mathbf{p})$ and $\mathrm{M}_d(\mathbf{p})$. The number of the pixels across an arm, which are outliers according to the outliers map $\mathrm{O}_1^{T_{\mathrm{LR}}}(\mathbf{p})$ (see Fig. 5a), is subtracted from the size of the arm. The sizes of the arms, after subtracting the number of outlier pixels, are denoted as $\mathrm{M}_l^{'}(\mathbf{p})$, $\mathrm{M}_r^{'}(\mathbf{p})$, $\mathrm{M}_u^{'}(\mathbf{p})$ and $\mathrm{M}_d^{'}(\mathbf{p})$, respectively.

Let $\mathrm{M}_{\max}^{'}(\mathbf{p})$ denote the maximum value among $\mathrm{M}_l^{'}(\mathbf{p})$, $\mathrm{M}_r^{'}(\mathbf{p})$, $\mathrm{M}_u^{'}(\mathbf{p})$ and $\mathrm{M}_d^{'}(\mathbf{p})$, while $\mathrm{M}_{\sec}^{'}(\mathbf{p})$ denotes the second highest value. Based on $\mathrm{M}_{\max}^{'}(\mathbf{p})$ and $\mathrm{M}_{\sec}^{'}(\mathbf{p})$, the following conditions are defined:

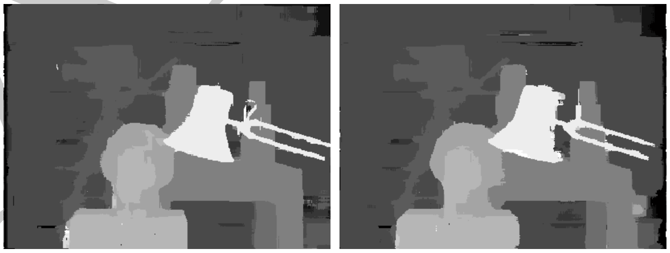
$$\mathrm{M}_{\max}^{'}(\mathbf{p})/\mathrm{M}_{\sec}^{'}(\mathbf{p}) > 2 \text{ and } \mathrm{M}_{\max}^{'}(\mathbf{p}) > R_{\mathrm{S}}/2 \qquad (18)$$

The first condition confirms that a direction has much more non-outlier pixels than the other directions, while the second condition confirms that there is a sufficient number of non-outlier pixels along this direction. In case both conditions in Eq. (18) are satisfied, then a higher weight is given to the path cost that corresponds to the direction from which $\mathrm{M}_{\max}^{'}(\mathbf{p})$ has been derived.

For example, if $\mathrm{M}_{\max}^{'}(\mathbf{p})$ is equal to $\mathrm{M}_u^{'}(\mathbf{p})$, which corresponds to direction $\mathbf{r}_{ud}$, then the weights used in Eq. (17) will be set as: $w_{ud}(\mathbf{p}) = w_\alpha$ and $w_{lr}(\mathbf{p}) = w_{rl}(\mathbf{p}) = w_{du}(\mathbf{p}) = w_\beta$, where $w_\alpha > w_\beta$ and $w_\alpha + 3\cdot w_\beta = 1$. That is, a higher weight is given to the direction that has much more pixels that belong to the same surface as $\mathbf{p}$, when compared to the other directions, which at the same time are non-outliers. If any of the conditions in Eq. (18) is not satisfied, then all weights are set equal to 1. Experiments on the selection of optimal values for w_α and w_β are given in subsection III-A.

To summarize, two novel ideas regarding the semi-global optimization have been introduced in this subsection. The first idea concerns the introduction of a scheme for defining the weights of each path cost, contrary to the methods in [13], [20], [31] that do not assign weights to the path costs. The second idea concerns the employment of an adaptive threshold for the detection of depth discontinuities, contrary to the methods in [13], [20] that use a static threshold.

The output of the Winner-Take-All (WTA) on $C''(\mathbf{p},d)$, which has been estimated from Eq. (17), gives the disparity map $d_{\mathrm{LR}}^{'}(\mathbf{p})$ (see Fig. 7a). If the right image is considered as the reference image, then the disparity map $d_{\mathrm{RL}}^{'}(\mathbf{p})$ (see Fig. 7b) is acquired.



(a) (b)

Fig. 7: (a) $d_{\mathrm{LR}}^{'}(\mathbf{p})$ and (b) $d_{\mathrm{RL}}^{'}(\mathbf{p})$ disparity maps after weighted semi-global optimization.

D. Disparity refinement in outlier regions

The disparity maps after cost volume optimization may contain a large number of outliers in occluded regions, uniform areas and near depth discontinuities. With the algorithmic steps, described through this section, these problematic areas can be handled efficiently in order to get a disparity map of high accuracy.

1) Outliers Handling:

a) Outliers detection: The disparity maps of the left image $d_{\mathrm{LR}}^{'}(\mathbf{p})$ (see Fig. 7a) and the right image $d_{\mathrm{RL}}^{'}(\mathbf{p})$ (see Fig. 7b) are taken into account so as to detect problematic areas. According to the Left-Right consistency check: $\left|d_{\mathrm{LR}}^{'}(\mathbf{p}) - d_{\mathrm{RL}}^{'}(\mathbf{p} - d_{\mathrm{LR}}^{'}(\mathbf{p}))\right| \leq T_{\mathrm{LR}}$, the outliers map $\mathrm{O}_2^{T_{\mathrm{LR}}}(\mathbf{p})$ (see Fig. 8) is acquired for $T_{\mathrm{LR}} = 0$. The blue regions in Fig. 8 denote the outlier pixels for which above relation does not hold, while the red regions denote the inlier pixels.

The detected outliers are filled with reliable disparities from neighboring areas by combining "Background outliers

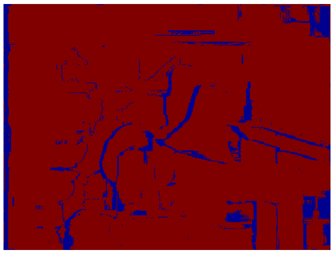


Fig. 8: Outliers map $O_2^{T_{\mathrm{LR}}}(\mathbf{p})$ for threshold $T_{\mathrm{LR}} = 0$.

handling” and “Generic outliers handling”. The filled outliers are then smoothed using bilateral filtering.

b) Background outliers handling: One of the simplest schemes for handling an outlier pixel $\mathbf{p}$, which may belong to the pixels of the occluded background, is to set its disparity $d(\mathbf{p})$ equal to the minimum disparity between the disparities of its spatially closest consistent (inlier) pixels on its left and its right side [15]. Practically, if $\mathbf{p}_l$ and $\mathbf{p}_r$ stand for the nearest consistent (inlier) pixels on the left and the right side of $\mathbf{p}$, respectively, the disparity value of $\min(d(\mathbf{p}_l), d(\mathbf{p}_r))$ is assigned to $d(\mathbf{p})$.

c) Generic outliers handling: The outliers may correspond to mismatches and not to background occlusion. In order to handle possible mismatches, we have introduced a straightforward scheme which precedes the “Background outliers handling” scheme. The “Generic outliers handling” scheme does not presume that an outlier pixel $\mathbf{p}$ belongs to the background, but it checks whether its left or right side has more similar (in term of intensity) pixels to that pixel, before assigning a disparity value to $\mathbf{p}$. In more detail, for an outlier pixel $\mathbf{p}$, separately for the left and right side, the inlier pixels, for which the condition in Eq. (19) is verified, are counted. The condition in Eq. (19) examines whether pixels $\mathbf{p}$ and $\mathbf{p}+\mathbf{s}$ are close in intensity.

$$|I_l(\mathbf{p}) - I_l(\mathbf{p}+\mathbf{s})| < \tau(I_l(\mathbf{p})), \tag{19}$$

In Eq. (19), $\tau(I_l(\mathbf{p}))$ is defined according to Eq. (16), $\mathbf{s} = (-\mathrm{s}_{\mathrm{x}_l}, 0)^{\mathbf{T}}$, $\mathrm{s}_{\mathrm{x}_l} \in [1, \mathrm{s}_{l_{\max}}(\mathbf{p}))$ stand for the left side and $\mathbf{s} = (\mathrm{s}_{\mathrm{x}_r}, 0)^{\mathbf{T}}$, $\mathrm{s}_{\mathrm{x}_r} \in [1, \mathrm{s}_{r_{\max}}(\mathbf{p}))$ stand for the right side. $\mathrm{s}_{l_{\max}}(\mathbf{p})$ and $\mathrm{s}_{r_{\max}}(\mathbf{p})$ are the integer values for which the condition of Eq. (19) fails for the first time when examining the left and the right sides, respectively. For the pixels on the left side of $\mathbf{p}$, the weights $\beta_l(\mathbf{p}+\mathbf{s})$ are calculated from:

$$\beta_l(\mathbf{p}+\mathbf{s}) = \begin{cases} 1, \text{ if } \mathbf{p}+\mathbf{s} \text{ is inlier} \\ 0, \text{ if } \mathbf{p}+\mathbf{s} \text{ is outlier.} \end{cases} \tag{20}$$

Afterwards, for the left side the following disparity histogram is generated:

$$\mathrm{H}_l(\mathbf{p}, d_{\mathrm{i}}) = \sum_{\mathbf{p}+\mathbf{s}:\ d(\mathbf{p}+\mathbf{s}) == d_{\mathrm{i}}} \beta_l(\mathbf{p}+\mathbf{s}), \tag{21}$$

where $d_{\mathrm{i}} \in [\mathit{disparity\ range}]$. Each bin in the $\mathrm{H}_l(\mathbf{p}, d_{\mathrm{i}})$ histogram corresponds to a specific disparity value $d_{\mathrm{i}} \in [\mathit{disparity\ range}]$, where the value (height) of the bin is equal to the total number of the left inlier pixels whose disparity is equal to the specific d_{i} disparity value.

In an analogous manner, the disparity histogram $\mathrm{H}_r(\mathbf{p}, d_{\mathrm{i}})$ is generated for the right side. Let now the maximum values of the left and the right histograms be $\mathrm{h}_{l_{\max}}(\mathbf{p}) = \max\limits_{d_{\mathrm{i}}} \left\{ \mathrm{H}_l(\mathbf{p}, d_{\mathrm{i}}) \right\}$ and $\mathrm{h}_{r_{\max}}(\mathbf{p})$, respectively and the disparity values, which correspond to the histogram bins with the maximum values, be $d_{l_{\max}}(\mathbf{p}) = \operatorname*{argmax}\limits_{d_{\mathrm{i}}} \left\{ \mathrm{H}_l(\mathbf{p}, d_{\mathrm{i}}) \right\}$ and $d_{r_{\max}}(\mathbf{p})$, respectively. Based on the above, the new disparity estimate $d(\mathbf{p})$ is given from:

$$d(\mathbf{p}) = \begin{cases} d_{l_{\max}}(\mathbf{p}), \text{ if } (\mathrm{h}_{l_{\max}}(\mathbf{p}) > \mathrm{h}_{r_{\max}}(\mathbf{p}) \ \&\ \mathrm{h}_{l_{\max}}(\mathbf{p}) > \mathrm{R}_{\mathrm{S}}/2) \\ d_{r_{\max}}(\mathbf{p}), \text{ if } (\mathrm{h}_{l_{\max}}(\mathbf{p}) < \mathrm{h}_{r_{\max}}(\mathbf{p}) \ \&\ \mathrm{h}_{r_{\max}}(\mathbf{p}) > \mathrm{R}_{\mathrm{S}}/2) \end{cases} \tag{22}$$

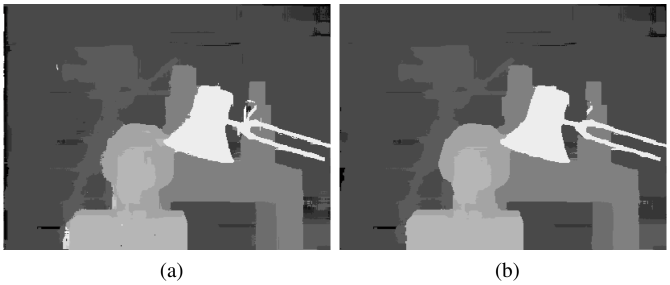
The second part of the conditions in Eq. (22) ($\mathrm{h}_{l_{\max}}(\mathbf{p}) > \mathrm{R}_{\mathrm{S}}/2$ or $\mathrm{h}_{r_{\max}}(\mathbf{p}) > \mathrm{R}_{\mathrm{S}}/2$) confirms that there is a sufficient number of inlier pixels on either the left or the right side of $\mathbf{p}$ that have the most frequent disparity value, before setting $d(\mathbf{p})$ equal to this most frequent disparity value.

d) Combination of “Generic outliers handling” and “Background outliers handling”: The disparity map of Fig. 7a, after applying “Generic outliers handling”, is visualized in Fig. 9a. The outlier pixels that have been handled using Eq. (22) are considered now as inliers. For the remaining outliers (i.e. none of the conditions in Eq. (22) holds), the scheme in paragraph “Background outliers handling” (II-D1b) is applied.

e) Bilateral smoothing of the handled outliers: The outliers handling, which is based on the combination of “Generic outliers handling” and “Background outliers handling”, may generate artifacts in the disparity map. Therefore, a bilateral filter is used to smooth the handled outliers. The bilateral filter weights are given by:

$$W_{\mathbf{p},\mathbf{q}} = \frac{1}{k} \cdot \exp\left(- \left(\frac{\Delta s_{\mathbf{p},\mathbf{q}}}{\gamma_s} + \frac{\Delta c_{\mathbf{p},\mathbf{q}}}{\gamma_c} \right) \right), \tag{23}$$

where k is a normalization factor, $\Delta s_{\mathbf{p},\mathbf{q}}$ and $\Delta c_{\mathbf{p},\mathbf{q}}$ denote the spatial distance and the color difference, respectively, between pixels $\mathbf{p}$, $\mathbf{q}$ and γ_s, γ_c are constant parameters that adjust the spatial and color distance. The parameters of the bilateral filter are set as in [15]: $\gamma_s = 9$, $\gamma_c = 0.1$ and the window size is $R_{\mathrm{S}} \times R_{\mathrm{S}}$. The disparity map of Fig. 9a after applying “Background outliers handling” and bilateral smoothing is visualized in Fig. 9b.



(a) (b)

Fig. 9: Disparity map after applying: (a) “Generic outliers handling” and (b) “Background outliers handling” plus bilateral smoothing.

2) Disparity edges refinement: Disparity edges, which correspond to depth discontinuities, may contain disparity outliers [13]. Therefore, a simple and efficient approach to refine the

Image	Resolution	Disp. levels	Proposed	TSGO[45]	JSOSP+GCP [46]	KADI [47]	ADCensus [13]	AdaptiveGF [25]	Appr. in [44]	VariableCross [21]	Appr. in [30]
Tsukuba	384 x 288	15	1.9	3	143.4	24.33	2.5	2.12	1.5	0.9	1.23
Venus	434 x 383	20	3.6	7	249.0	49.25	4.5	2.96	2.8	1.6	2.45
Teddy	450 x 375	60	9.7	20	262.8	154.23	15.0	8.78	7.3	2.4	7.23
Cones	450 x 375	60	9.6	20	306.6	154.78	15.0	8.76	7.2	2.4	7.05

TABLE I: Computational time in seconds.

disparity results at the disparity edges is introduced. Initially, the pixels that belong to disparity edges are assumed to have absolute disparity difference greater or equal to 1 with at least one of their 4-adjacent pixels. Fig. 10a shows with red the disparity edges extracted from the disparity map of Fig. 9b.

Around each pixel $\mathbf{p}_\mathrm{c}$ of the disparity edge, a circular region of radius 4 is defined. The color similarity between the center pixel $\mathbf{p}_\mathrm{c}$ and a pixel $\mathbf{q}$ within the circular region is estimated as:

$$w(\mathbf{p}_\mathrm{c},\mathbf{q}) = e^{\left(\frac{-\Delta I(\mathbf{p}_\mathrm{c},\mathbf{q})}{\gamma_\mathrm{c}}\right)}, \tag{24}$$

where

$$\Delta I\left(\mathbf{p}_\mathrm{c},\mathbf{q}\right) = \sqrt{\sum_{\mathrm{c}\in\{\mathrm{R},\mathrm{G},\mathrm{B}\}} |I^\mathrm{c}(\mathbf{p}_\mathrm{c}) - I^\mathrm{c}(\mathbf{q})|^2}. \tag{25}$$

A disparity histogram is generated for each $\mathbf{p}_\mathrm{c}$, where the values of its disparity bins are computed as follows:

$$\mathrm{H}_{\mathbf{p}_\mathrm{c}}(\mathbf{p}_\mathrm{c}, d_\mathrm{i}) = \sum_{\mathbf{q}:\ d(\mathbf{q})\ ==\ d_\mathrm{i}} w\left(\mathbf{p}_\mathrm{c},\mathbf{q}\right), \tag{26}$$

where $d_\mathrm{i} \in [disparity\ range]$. Let now the maximum and the second maximum value of $\mathrm{H}_{\mathbf{p}_\mathrm{c}}(\mathbf{p}_\mathrm{c}, d_\mathrm{i})$ be $\mathrm{h}_{\max}(\mathbf{p}_\mathrm{c}) = \max\limits_{d_\mathrm{i}}\left\{\mathrm{H}_{\mathbf{p}_\mathrm{c}}(\mathbf{p}_\mathrm{c}, d_\mathrm{i})\right\}$ and $\mathrm{h}_{\mathrm{sec}}(\mathbf{p}_\mathrm{c})$, respectively, and the corresponding disparity value for $\mathrm{h}_{\max}(\mathbf{p}_\mathrm{c})$ be $d_{\mathrm{h}_{\max}}(\mathbf{p}_\mathrm{c}) = \operatorname*{argmax}\limits_{d_\mathrm{i}}\left\{\mathrm{H}_{\mathbf{p}_\mathrm{c}}(\mathbf{p}_\mathrm{c}, d_\mathrm{i})\right\}$. If $\mathrm{h}_{\max}(\mathbf{p}_\mathrm{c})/\mathrm{h}_{\mathrm{sec}}(\mathbf{p}_\mathrm{c}) > 2$ then $d(\mathbf{p}_\mathrm{c}) = d_{\mathrm{h}_{\max}}(\mathbf{p}_\mathrm{c})$, otherwise the disparity value of $d(\mathbf{p}_\mathrm{c})$ does not change.

The disparity result after the disparity edges refinement is depicted in Fig. 10b. A median filter, using a 3x3 neighborhood, is applied to the disparity result of Fig. 10b in order to remove spurious disparities before acquiring the final disparity map, which is depicted in the upper-left image of Fig. 11.

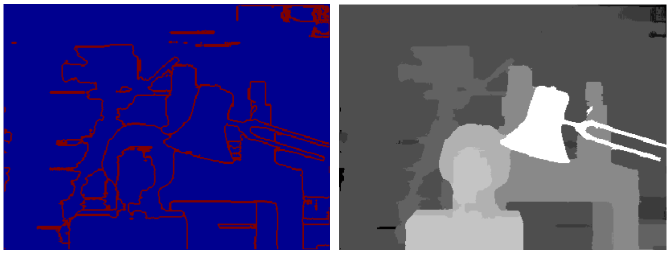

(a) (b)

Fig. 10: (a) Disparity edges and (b) disparity map after disparity edges refinement.

E. Computational cost

A C++ implementation of the algorithm is used to report on the required computational time. The algorithm was executed on a desktop PC with Core i7-3770 3.40 GHZ CPU and 8 GB RAM. The low computational time, using as input each of the four stereo pairs of the Middlebury evaluation benchmark [48], is indicated in the column "Proposed" of Table I. The measured time is the average of 10 separate runs and includes both the time for performing mean-shift segmentation (see subsection II-A) and the time for executing the algorithmic steps of the methodology (see subsections II-B to II-D).

The reason that the proposed approach has low computation time is fourfold. Firstly, the combined matching cost metric of Eq. (4) can be rapidly estimated, since it is formed by three terms which can be computed fast (see subsection II-B1). Secondly, guided image filtering, which is used for the cost filtering in subsection II-B2, has the advantage that its computational cost is independent to the size of the support window. This is because guided image filtering can be expressed as a linear transform according to subsection II-B4. While the computational complexity of guided image filtering is independent to the size of the support window, in the cost aggregation approaches of [13], [18], [19], [20], [21] the computational cost increases as the size of the support window increases. Thirdly, semi-global optimization (see subsection II-C2) helps to notably increase the disparity estimation accuracy at low computational cost as it is also verified in [13], [19], [32]. Fourthly, the outliers handling (see subsection II-D1) and the disparity edges refinement (see subsection II-D2) techniques, which belong to the disparity refinement step, have low computational complexity and at the same time they are applied just to outliers areas and disparity edges, respectively, and not to the complete disparity map.

The steps of the algorithm with increased computational cost include the content-based guide image filtering (see subsection II-B2) and the semi-global optimization (see subsection II-C2). However, these parts can be implemented in Graphics Processing Units (GPU) as can be verified in [15], [25] for the guided image filtering and in [13], [49] for the semi-global optimization. Therefore, the proposed methodology is appropriate for real-time GPU implementation.

III. Experimental results

In this section, the experimental results on multiple datasets are presented. In more detail, the four stereo pairs of the Middlebury online stereo evaluation benchmark [48] are used for the evaluation of this method. Furthermore, this section presents experimental results on 27 additional Middlebury stereo pairs, in order to verify the efficiency of the proposed approach.

	Best	Adapt. Windows as in [25]	S-G as in [13]	S-G as in [20]	Outliers handl. as in [15]
Avg. Rank	16.8	23.4	19.1	18.0	20.1
Nonocc (%)	1.91	2.12	1.94	1.95	1.95
All (%)	4.68	4.85	4.71	4.70	4.72
Disc (%)	6.28	6.24	6.29	6.26	6.32
APBP (%)	4.29	4.41	4.31	4.30	4.33

TABLE II: Evaluation results.

(α_1, α_2)	$(0.75, 0.20)$	$(0.70, 0.20)$	$(0.75, 0.15)$	$(0.70, 0.25)$	$(0.80, 0.15)$	$(0.80, 0.20)$	$(0.75, 0.25)$
Avg. Rank	16.8	21.4	18.8	21.5	17.8	29.5	30.2
Nonocc (%)	1.91	2.01	1.95	1.98	1.97	2.16	2.11
All (%)	4.68	4.83	4.77	4.74	4.75	4.94	4.87
Disc (%)	6.28	6.39	6.40	6.29	6.39	6.91	6.70
APBP (%)	4.29	4.41	4.37	4.33	4.37	4.67	4.56

TABLE III: Balance weights α_1, α_2 testing.

(w_α, w_β)	$(1.6, 0.8)$	$(1.3, 0.9)$	$(1.45, 0.85)$	$(1.75, 0.75)$	$(1.9, 0.7)$	$(1.0, 1.0)$
Avg. Rank	16.8	18.1	17.3	17.1	17.6	19.4
Nonocc(%)	1.91	1.92	1.91	1.91	1.91	1.96
All(%)	4.68	4.70	4.69	4.68	4.68	4.72
Disc(%)	6.28	6.28	6.29	6.29	6.32	6.31
APBP (%)	4.29	4.30	4.30	4.30	4.31	4.33

TABLE IV: w_α, w_β weights testing.

A. *Set of optimal parameters*

The parameters used for the experiments are the same for all tested stereo pairs. More specifically, the parameters used for the estimation of the cost term (see subsection II-B1) are selected equal to: $\{\alpha_1, \alpha_2, T_{\text{gra}}, T_{\text{gab}}, T_{\text{BT}}\} = \{0.75, 0.20, 2/255, 4/255, 7/255\}$. The variables used for the cost filtering are the smoothness parameter ε (see subsection II-B2), which is set to $\varepsilon = 0.0001$ (ε has the same value as in [15]) and the parameter R_{S} that defines the size of the rectangular window (see subsection II-B3), which is set to $R_{\text{S}} = 17$. The selection of $R_{\text{S}} = 17$ is based on the experiments described in subsection III-B2. The weights w_α, w_β that are used to define the weights $w_{lr}(\mathbf{p}), w_{rl}(\mathbf{p}), w_{ud}(\mathbf{p}), w_{du}(\mathbf{p})$ in Eq. (17) (see subsection II-C2) are set to $(w_\alpha, w_\beta) = (1.6, 0.8)$.

In the column "Best" of Table II, the numeric results from the Middlebury Stereo evaluation for the disparity maps extracted using these optimal parameters, are given. The results include the overall performance measure ("Avg. Rank"), the error in non-occluded regions ("Nonocc (%)"), the error in all regions ("All (%)"), the error near depth discontinuities ("Disc (%)") and the average percent of bad pixels ("APBP (%)").

In the following, the results that justify the selection of optimal values for (α_1, α_2) and (w_α, w_β) are presented.

Table III presents the results of testing the balance weights α_1 and α_2 of Eq. (4) (see subsection II-B1). The results show that optimal parameters $(\alpha_1, \alpha_2) = (0.75, 0.20)$ give better results compared to the results obtained using balance parameters with values that are slightly below or slightly above the optimal balance parameters. Additionally, from the last two columns $\big((\alpha_1, \alpha_2) = (0.80, 0.20)$ and $(\alpha_1, \alpha_2) = (0.75, 0.25)\big)$, where the values of the balance parameters cause the elimination of the Birchfield-Tomasi dissimilarity term from Eq. (4), it can be deduced that the disparity estimation accuracy reduces considerably after eliminating the Birchfield-Tomasi dissimilarity term. This fact shows the importance of using the Birchfield-Tomasi dissimilarity term in Eq. (4).

Table IV presents the results of testing weights w_α and w_β. The weights $w_{lr}(\mathbf{p}), w_{rl}(\mathbf{p}), w_{ud}(\mathbf{p})$ and $w_{du}(\mathbf{p})$ of Eq. (17) are defined according to w_α and w_β (see subsection II-C2). The results show that optimal weights $(w_\alpha, w_\beta) = (1.6, 0.8)$ give better results compared to the results obtained using weights with values that are close to the optimal path direction weights. The last column of Table IV gives results for weights $(w_\alpha, w_\beta) = (1.0, 1.0)$. For $(w_\alpha, w_\beta) = (1.0, 1.0)$, the weights in Eq. (17) are uniformly set as $w_{lr}(\mathbf{p}) = w_{rl}(\mathbf{p}) = w_{ud}(\mathbf{p}) = w_{du}(\mathbf{p}) = 1$. Obviously, the last column of Table IV gives worse results than the rest of its columns. Therefore, it can be deduced that the use of different values for weights $w_{lr}(\mathbf{p}), w_{rl}(\mathbf{p}), w_{ud}(\mathbf{p})$ and $w_{du}(\mathbf{p})$ in Eq. (17) helps to achieve better disparity estimation results than using uniform values.

B. *Middlebury Online Stereo Evaluation Benchmark*

1) Disparity results: The disparity results of the proposed method, for the optimal parameters set, accompanied with the disparity error maps as extracted by the Middlebury evaluation system are visualized in Fig. 11. Errors in non-occluded and occluded regions are marked in black and gray respectively.

Table V displays the Middlebury online evaluation results of several approaches, for error threshold equal to 1. The "nonocc (%)" error measure stands for the average estimation error only at the non-occluded regions of the left image, i.e. the regions of the left image that are also visible in the right image. The "disc (%)" measure considers the error only at the regions close to depth discontinuities. Finally, the "all (%)" error measure is calculated considering the whole left image. The subscript blue values, next to the percentages of error pixels, give the relative ranks in each column. The "Avg. Rank" is given by the average of the blue subscript values. Finally, the "APBP (%)" acronym stands for the average percent of bad pixels in "all" regions and is given by the average of the error percentages that are provided in the "all (%)" column. For further details, the reader is referred to [1]. The first seven approaches (which are displayed above the dashed line in Table V) correspond to the top ranked methods of the Middlebury evaluation benchmark. The rest four approaches (which are displayed below the dashed line) do not rank among top methods, but they are mentioned for comparison purposes.

The ranking results in the "Avg. Rank" column of Table V indicate that the proposed method is 5th out of 164 methods that are included in the Middlebury Stereo Evaluation. The 2nd ranked method TSGO [45] proposes a two-step energy minimization procedure: first a fully connected model is used for cost filtering and then a locally connected model is applied to compute the final disparity maps. The 3rd ranked method JSOSP+GCP [46] presents a stereo framework that handles a scene as a set of 3D entities with compact and smooth disparity distributions. The 3D entity-based representation enables the exploitation of a GCPs-plane constraint, a joint second-order smoothness prior and a soft segmentation constraint to estimate the 3D entities. The 4th ranked method KADI [47] presents a two-phase strategy for combining separate cost volumes, a mean-shift segmentation-driven approach for handling disparity outliers and disparity histogram analysis for fostering low-textured area plane fitting. Notice that [50] was under review at the time of this paper's writing. Therefore, the

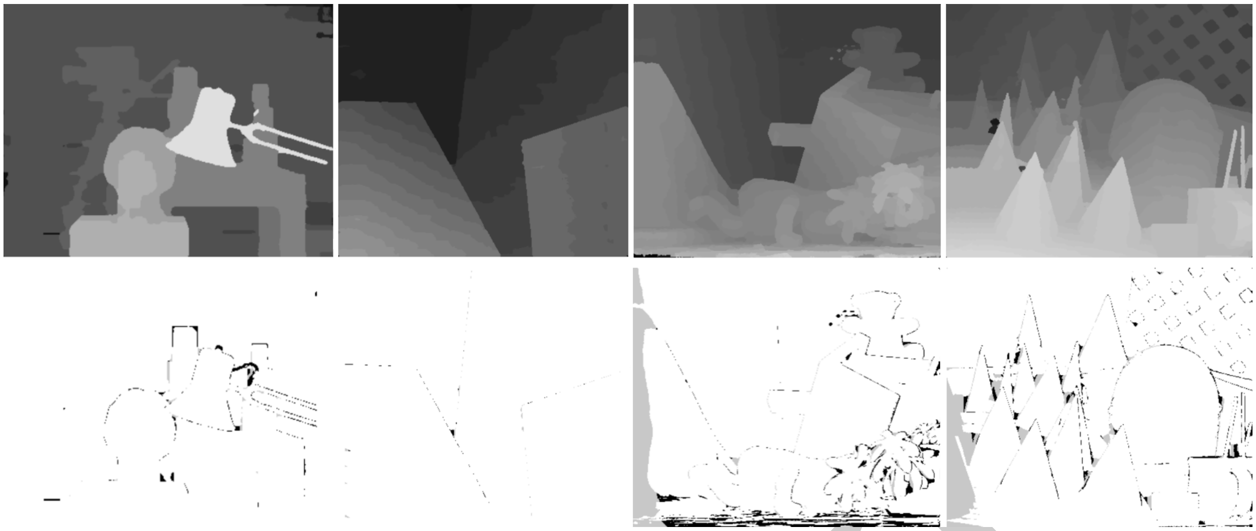


Fig. 11: Disparity maps generated with the proposed algorithm and their corresponding disparity error maps for error threshold equal to 1.

Algorithm	Pos.	Avg.	Tsukuba			Venus			Teddy			Cones			APBP(%)
		Rank	nonocc(%)	all(%)	disc(%)	nonocc(%)	all(%)	disc(%)	nonocc(%)	all(%)	disc(%)	nonocc(%)	all(%)	disc(%)	
IGSM[50]	1	10.5	0.93_{10}	1.37_{13}	5.05_{12}	0.07_{2}	0.17_{5}	1.04_{2}	4.08_{20}	5.98_{10}	11.4_{21}	2.14_{9}	6.97_{14}	6.27_{8}	3.79
TSGO[45]	2	13.8	0.87_{4}	1.13_{1}	4.66_{6}	0.11_{11}	0.24_{15}	1.47_{14}	5.61_{47}	8.09_{21}	$13.8_{;40}$	1.67_{2}	6.16_{3}	4.95_{2}	4.06
JSOSP+GCP[46]	3	15.2	0.74_{1}	1.34_{10}	3.98_{1}	0.08_{4}	0.16_{1}	1.15_{4}	3.96_{18}	10.1_{41}	11.8_{22}	2.28_{20}	7.91_{38}	6.74_{23}	4.18
KADI[47]	4	15.6	1.02_{17}	1.23_{4}	5.51_{18}	0.08_{3}	0.20_{8}	1.11_{3}	5.16_{38}	9.43_{35}	13.0_{34}	2.07_{4}	7.16_{19}	5.97_{4}	4.33
Proposed	5	16.8	1.01_{16}	1.32_{9}	5.17_{15}	0.08_{5}	0.21_{11}	1.17_{6}	4.35_{24}	9.83_{38}	12.3_{27}	2.19_{14}	7.35_{22}	6.43_{15}	4.29
SSCBP[51]	6	18.2	1.05_{20}	1.39_{15}	5.57_{20}	0.10_{8}	0.16_{2}	1.39_{11}	3.44_{14}	8.32_{26}	9.95_{15}	2.60_{35}	7.13_{18}	7.23_{34}	4.03
ADCensus[13]	7	18.8	1.07_{24}	1.48_{22}	5.73_{27}	0.09_{6}	0.25_{19}	1.15_{4}	4.10_{21}	6.22_{11}	10.9_{18}	2.42_{26}	7.25_{21}	6.95_{27}	3.97
AdaptiveGF[25]	21	36.8	1.04_{19}	1.53_{25}	5.62_{22}	0.17_{34}	0.41_{47}	1.98_{32}	5.71_{51}	11.3_{57}	14.3_{47}	2.44_{28}	8.22_{48}	7.05_{31}	4.98
Approach in [44]	23	37.5	1.66_{80}	2.01_{72}	6.58_{51}	0.11_{9}	0.30_{26}	1.50_{15}	5.05_{33}	10.4_{42}	13.9_{43}	2.39_{23}	7.71_{31}	6.85_{25}	4.87
VariableCross[21]	115	109.9	1.99_{101}	2.65_{99}	6.77_{57}	0.62_{107}	0.96_{105}	3.20_{78}	9.75_{132}	15.1_{129}	18.2_{111}	6.28_{140}	12.7_{132}	12.9_{128}	7.60
Approach in [30]	Undef.	Undef.	6.18	6.88	13.70	5.69	6.43	12.20	13.60	20.90	28.20	6.80	13.30	14.00	12.30

TABLE V: This table provides the average rank position and the average rank on the online Middlebury stereo evaluation benchmark and the percentages of error pixels in non-occluded regions ("Nonocc (%)"), all regions ("All (%)") and regions near depth discontinuities ("Disc (%)"), respectively, for the four evaluated stereo image pairs. The subscript blue values, next to the percentages of error pixels, give the relative ranks in each column. The last column gives the average percent of bad pixels ("APBP (%)").

proposed method ranks 4th among already published methods. This is an important achievement bearing in mind the reduced computational complexity of this algorithm and its suitability to be implemented in GPU. Moreover, although our method is less accurate than TSGO, JSOSP+GCP and KADI, it is faster than them. This fact is evident in Table I that shows the computational times of the proposed method and the approaches in TSGO, JSOSP+GCP and KADI (the respective computational times were obtained from [45], [46] and [47]). Table V shows that the proposed approach has better ranking SSCBP[51]. The computational complexity of SSCBP[51] is not available to compare it against our approach. Table I also includes the CPU computation time of the ADCensus [13] approach that is listed in Table V.The proposed method has better ranking than ADCensus [13] and at the same time it is faster than this approach. The approach in [25], which also exploits the guided image filter, is slightly faster than our approach as confirmed from Table V. However, it ranks 21th in the Middlebury online evaluation benchmark, while our approach ranks 5th. The error rates of [25] are given in Table V. As it is evident in Table I, our preceding work in [44] is about 25% faster than the current approach, mainly due to the fact that the work in [44] computes the guided image filter for just one support window, while the current approach computes the guided image filter for two different support windows. However, the work in [44] ranks 23rd and it is notably less accurate than the approach presented in this article with respect to all columns of Table V. There are other approaches that have evidently lower computational cost than the presented approach, such as VariableCross [21] and the approach in [30] in Table I, but they have significantly lower disparity estimation accuracy as it is evident in Table V. The value for the average ranking is undefined for [30] in Table V, since this approach is not included in Middlebury online evaluation system. The values in the rest columns of Table V, regarding [30], were adopted from the corresponding paper.

For the stereo pairs of the Middlebury Stereo Evaluation benchmark, the proposed method ranks: 12th for the "Tsukuba" image pair, 6th for the Venus image pair, 29th for the Teddy image pair and 15th for the "Cones" image pair.

2) Experiments on the definition of support windows: In order to prove why the exploitation of rectangular support windows of two sizes (as suggested in subsection II-B3) enhances the disparity estimation results, we have performed experiments using support windows of either rectangular or square shape.

In specific, the following cases of defining support windows have been examined:

- Case 1: Use one rectangular support window with size R_{S} x $\lceil R_{\mathrm{S}}/2 \rceil$.
- Case 2: Use one square support window with size R_{S} x R_{S}.

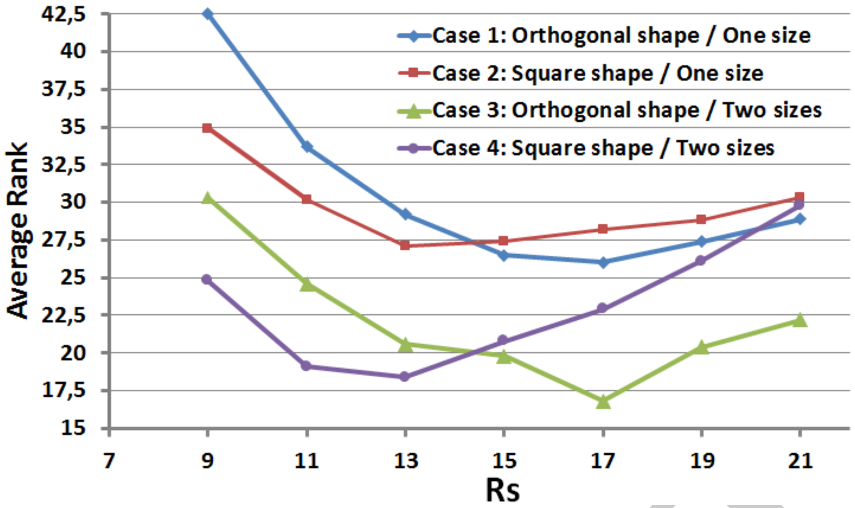


Fig. 12: Average Rank against R_{S} for four different cases of defining support windows sizes.

- Case 3: Use two rectangular support windows with sizes R_{S} x $\lceil R_{\mathrm{S}}/2 \rceil$ and $2R_{\mathrm{S}}$ x R_{S}.
- Case 4: Use two square support windows with sizes R_{S} x R_{S} and $2R_{\mathrm{S}}$ x $2R_{\mathrm{S}}$.

Those four cases affect subsection II-B3. In more detail, when using just one support window (“Case 1” or “Case 2”) the selection of the appropriate support window size for each pixel is not required, while when using two support windows (“Case 3” or “Case 4”) the condition “If $\bar{\mathrm{M}}(\mathbf{p}) > R_{\mathrm{S}}$” (see subsection II-B3) is examined to decide which of the two windows is more appropriate to determine the filtered cost of each pixel.

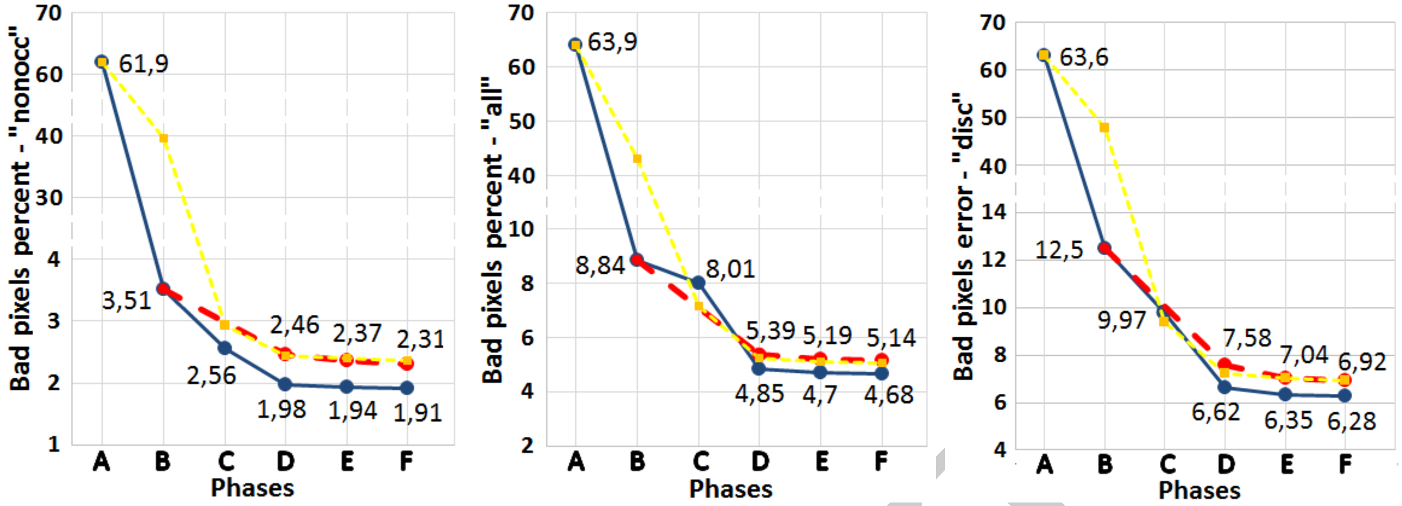
The curves in Fig. 12 show the Average Rank (as estimated according to the online Middlebury evaluation) for each of the above four cases, for different values of R_{S}. An important finding is that between “Case 1” and “Case 2”, “Case 1” (rectangular support window) gives better Average Rank than “Case 2” (square support window). Moreover, by comparing “Case 1” and “Case 2” with “Case 3” and “Case 4”, it is evident that the use of two support windows sizes gives a better Average Rank. Finally, it is shown that “Case 3” (this is the case proposed in subsection II-B3) gives the best disparity estimation results among all cases. The value of R_{S} for which the best Average Rank is accomplished is $R_{\mathrm{S}} = 17$.

3) Evaluation of the methodology: The initial disparity map that is generated from the initial cost volume (which is computed via the matching cost computation step (Phase A)) is heavily corrupted with noisy disparities. Based on the Middlebury online benchmark, this subsection examines how the initial disparity map is improved after applying sequentially: (1) content-based guided image filtering (Phase B), (2) disparity optimization relying on weighted semi-global optimization(Phase C), (3) outliers handling (Phase D), (4) disparity edges refinement (Phase E) and (5) median filtering (Phase F). Outliers handling (Phase D) and disparity edges refinement (Phase E) constitute the disparity refinement step.

The blue lines in Fig. 13 depicts how the average percent of bad pixels in the disparity map decreases after applying each of the above phases. Fig. 13 includes results for non-occluded regions (see Fig. 13a), all regions (see Fig. 13b) and regions near depth discontinuities (see Fig. 13c).

The content-based guided image filtering in “Phase B” significantly enhances the initial disparity map. This is evident in the results of Fig. 13, where the average percent of bad pixels drastically reduces from “Phase A” to “Phase B”. The disparity map is further improved in “Phase C” after applying disparity optimization. The improvement is stronger for the regions near depth discontinuities. Outliers handling in “Phase D” further reduces the average percent of bad pixels. The decrease is more pronounced for all regions, which is rational since all regions include the occluded regions. The disparity edges refinement in “Phase E” helps to further lower the average percent of bad pixels, but in a less degree than the preceding steps, since it is applied locally to disparity edges. Finally, median filtering in “Phase F” slightly reduces the average percent of bad pixels.

In the following, the improvement in the disparity map quality, introduced by the aforementioned steps, is visually demonstrated. The initial disparity map, which is acquired via the matching cost computation step, is heavily corrupted with estimation-error noise, as it is obvious in Fig. 2. After applying content-based guided image filtering the noise is removed, as it is evident in Fig. 4a. Disparity optimization further improves the disparity results. This is clearly seen from the comparison between Fig. 4a and Fig. 7a. Fig. 9b shows the disparity map after performing outliers handling to the disparity map of Fig. 7a. The outlier regions (which are denoted with blue in the outliers map of Fig. 8) of the disparity map in Fig. 7a have been efficiently filled with reliable disparities in the disparity map of Fig. 9b. Fig. 10b displays the disparity map after performing disparity edges refinement to the disparity map of Fig. 9b. The disparity edges of Fig. 9b (which are shown with red in Fig. 10a) have been effectively refined in Fig. 10b. After applying median filtering to the disparity map of Fig. 10b the disparity map of the upper-left image of Fig. 11 is generated. The slight improvement in the disparity map quality, introduced by the median filtering, is subtly obvious at locations where the disparity value changes. In the following,



(a)

(b)

(c)

Fig. 13: Average percent of bad pixels after applying sequentially the proposed method phases for (a) non-occluded regions, (b) all regions and (c) near depth discontinuities regions.

two modifications in the proposed workflow are also examined.

The red lines show how the average percent of bad pixels in the disparity map decreases after applying each of the above phases, except for “Phase C” which has been excluded from the workflow. Thought, the percent of bad pixels has been increased compared to the results of the blue lines, it still remains in low levels, even without applying “Phase C”. Indeed, the final disparity maps acquired for the “Tsukuba”, “Venus”, “Teddy” and “Cones” image pairs, without applying “Phase C”, rank 19th in the Middlebury online evaluation benchmark. This rank proves that our algorithm is able to keep high standards of disparity estimation accuracy (this is mainly due to the outstanding performance of the proposed content-based guided image filtering (Phase B)), even after removing the disparity optimization step and thus reducing the computational cost of our approach.

The yellow dashed lines show how the average percent of bad pixels in the disparity map increases after applying each of the above phases, with the difference that “Phase C” is applied before “Phase B” (when referring to the yellow dashed line, the letter “B” corresponds to “Phase C” and the letter “C” corresponds to “Phase B” on the x-axis of Fig. 13 (a), (b) and (c), respectively). Evidently, the results of the yellow dashed lines after applying “Phase D”, “Phase E” and “Phase F” are very close to the corresponding results of the red lines. For example, after applying “Phase F” the bad pixels error for non-occluded regions, all regions and near depth discontinuities regions are 2.36%, 5.05% and 6.93%, respectively (due to space reasons, numeric values have not been included for the markers of the yellow lines). Therefore, it is proved that the case of switching “Phase B” and “Phase C” in the workflow, gives similar results with the case of excluding “Phase C” from the workflow. Here, it has to be clarified that when applying “Phase C” before “Phase B” the weights of the path costs in the weighted semi-global optimization (Phase C) are equal to 1 ($w_{lr}(\mathbf{p}) = w_{rl}(\mathbf{p}) = w_{ud}(\mathbf{p}) = w_{du}(\mathbf{p}) = 1$), since it is not feasible to rely on the noisy initial disparity map (see Fig. 2) to perform the methodology required for deciding on the weights of the path costs, which is described in subsection II-C2.

Concluding, the workflow proposed in this paper gives better disparity estimation accuracy compared to the two workflow modifications described before. However, it worths noting that the removal of “Phase C” from the workflow leads to the reduction of the computational cost, while keeping the disparity estimation accuracy in high standards. On the other hand, the case of switching “Phase B” and “Phase C” in the workflow gives similar results with the case of excluding “Phase C”. However, with increased computational cost compared to the latter case.

4) Further testing: As mentioned in section III-A, the column “Best” of Table II gives the numeric disparity estimation results using the proposed methodology with the optimal parameters. In the rest of the columns of Table II, we provide experimental results after making some modifications to the methodology.

More specifically, the corresponding comparative quantitative evaluation results, where the scheme of [25] (see subsection II-B4) has been used instead of our scheme for performing guided image filtering, are given in the column of Table II entitled “Adapt. Windows as in [25]”. The comparison between the columns “Best” and “Adapt. Windows as in [25]” demonstrates that the proposed guided-image filtering helps the overall methodology to achieve better accuracy than when using the scheme of [25]. Additionally, our scheme has lower computational cost than the one of [25], as it has been explained in II-B4. Concluding, the proposed “Content-based guided image filtering” scheme helps our algorithm to achieve better accuracy at a lower computational cost, compared to the filtering scheme of [25].

In order to evaluate how the proposed improvements with respect to the semi-global optimization step (these improvements include the weighted average of path costs and an adaptive threshold for identifying depth discontinuities according to section II-C2) ameliorate the disparity results, we have included numeric results for the cases where the semi-global approaches of [13], [20] have been used, instead of our weighed semi-global approach. The approaches in [13], [20] use a simple average of path costs (i.e. the weights used in Eq. (17) are $w_{lr}(\mathbf{p}) = w_{rl}(\mathbf{p}) = w_{ud}(\mathbf{p}) = w_{du}(\mathbf{p}) = 1$), while their thresholds used for the identification of depth discontinuities are constant. There are two differences between [13], [20], regarding the semi-global optimization step: a) the constant thresholds for identifying the depth discontinuities are $\tau = 15/255$ and $\tau = 10/255$ in [13] and [20], respectively; b) the difference between a pixel and its previous pixel, along a path direction, is estimated using color and grayscale image information in [13] and [20], respectively. The numeric disparity results, which haven been estimated using the approaches of [13] and [20] for performing semi-global (S-G) optimization, are given in the columns of Table II with the annotations “S-G as in [13]” and “S-G as in [20]”, respectively. Except for the semi-global optimization step the other steps of our methodology are applied as they are. The differences between the column “Best” and the columns “S-G as in [13]” and “S-G as in [20]”, prove that without using the weighted semi-global optimization the disparity estimation accuracy decreases. On the other hand, the additional computational cost introduced by the weighted semi-global optimization is small.

In order to assess how our contribution with respect to the outliers handling step improves the disparity estimation results, we have included numeric results for the case where the scheme of [15] is used for the outliers handling, instead of our scheme. In particular, the scheme in [15] comprises the “Background outliers handling” (see subsection II-D1b) and the “Bilateral smoothing of the handled outliers” (see subsection II-D1e). The “Generic outliers handling” (see subsection II-D1c), which has been proposed in this work, is not included in the outliers handling scheme of [15]. The numeric disparity results, which have been estimated using the scheme of [15] for performing outliers handling, are given in the column of Table II with the annotation “Outliers handl. as in [15]”. The differences between the columns “Best” and “‘Outliers handl. as in [15]” prove that the integration of “Generic outliers handling” in the outliers handling scheme of [15] helps to improve the disparity estimation accuracy. Moreover, the additional computational cost introduced by “Generic outliers handling” is negligible.

The mean-shift segmentation map (subsection II-A) is exploited for selecting the appropriate support window size of each pixel (see subsection II-B3). Therefore, it is important to verify that small variations to the optimal parameters $(\sigma_s, \sigma_R) = (3, 3)$ that adjust the segmentation result do not affect significantly the performance of this method. Table VI shows the error results for different values of the spatial radius and space feature radius. For the pairs of $(\sigma_s, \sigma_R) = (2, 3)$, $(\sigma_s, \sigma_R) = (3, 4)$ and $(\sigma_s, \sigma_R) = (4, 4)$ our approach remains in the fifth position , while for the pair of $(\sigma_s, \sigma_R) = (2, 2)$ the method ranks seventh. Hence, it is deduced that even varying the segmentation parameters the method remains in the top performers.

	$(\sigma_s, \sigma_R) = (2, 2)$	$(\sigma_s, \sigma_R) = (2, 3)$	$(\sigma_s, \sigma_R) = (3, 4)$	$(\sigma_s, \sigma_R) = (4, 4)$
Avg. Rank	22.4	16.9	16.9	17.5
Nonocc (%)	2.03	1.94	1.91	1.90
All (%)	4.77	4.71	4.68	4.69
Disc (%)	6.45	6.31	6.27	6.26
APBP (%)	4.42	4.32	4.29	4.29

TABLE VI: Segmentation parameters testing.

Error %	Δd>1	Δd>1	Δd>2	Δd>2
	All	Visible	All	Visible
Proposed	12.07	7.71	8.32	5.07
TSGO[45]	12.79	10.05	8.92	7.24
ADCensus[13]	14.89	10.98	9.41	6.42
Inf. Permeability[12]	14.15	7.98	10.34	6.46
Guided Filter[15]	15.06	8.40	11.82	6.80
Geodesic Support[27]	16.49	9.85	11.76	8.04
Var. Cross[21]	17.13	8.81	12.69	7.04
Adapt. sup.[18]	16.94	9.54	13.10	7.42

TABLE VII: The error results for the extended stereo datasets.

C. Extended Comparison

Evaluation on just the four stereo pairs from the Middlebury online stereo database is not adequate to give a clear picture of the overall performance of an algorithm, since the average error rates of the best performing techniques are close to each other. Hence, our approach has also been evaluated on two additional Middlebury datasets in order to assess more extensively the performance of the proposed methodology. The 2005 and 2006 datasets, presented in [52], include 27 stereo pairs with their ground truth. The error percentage is measured for both non-occluded and all regions.

Table VII gives for multiple methods the percentages of erroneous pixels having 1 or 2 disparity level difference with respect to the ground truth. The tested methods include TSGO[45] and ADCensus[13] approaches, which are among the top performing methods in Table V. The results regarding TSGO[45] have been estimated using the source code provided by the authors, while the results regarding ADCensus[13] have been estimated based on our implementation of this approach. The results regarding the rest of methods in Table VII have been copied from the very recent work of [12]. The proposed work gives better results for the case of “All” and “Visible” regions than the rest of the methods. More specifically, for the case of “All” regions and $\Delta d > 1$, $\Delta d > 2$ the disparity errors of our approach are 0.72% and 0.6% less than the second best TSGO[45] method, respectively. While, for the case of “Visible” regions and $\Delta d > 1$, $\Delta d > 2$ the disparity errors of our approach are 0.27% and 1.35% less than the second best Inf. Permeability[12] and ADCensus[13] approaches, respectively.

The disparity maps for the 27 stereo pairs, with their respective disparity error maps for $\Delta d > 1$, can be found in the supplementary material that accompanies this paper.

IV. Conclusion

In this paper an approach that gives accurate disparity results for stereo image pairs was presented. The approach uses an efficient cost term, composed of three individual pixel-based cost terms, in order to estimate the initial cost volume. The filtered cost volume is acquired after applying image guided filtering to the initial cost volume, using rectangular support regions of two sizes. The optimization of the filtered cost volume is performed using weighted semi-global matching, where an adaptive threshold to identify depth discontinuities is used. Outliers handling is improved by introducing a straightforward scheme. The high performance of the proposed method is verified experimentally using the Middlebury evaluation benchmark and an extended stereo dataset.

Acknowledgment

This work is supported by the LASIE EU-funded IP project under Contract 607480.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

[2] Z. Xiaozhou, L. Huimin, Y. Xingrui, L. Yubo, and Z. Hui, "Stereo vision based traversable region detection for mobile robots using u-v-disparity," in *Chinese Control Conference*, pp. 5785–5790, 2013.

[3] E. Izquierdo and J.-R. Ohm, "Image-based rendering and 3d modeling: A complete framework," *Signal Processing: Image Communication*, vol. 15, no. 10, pp. 817–858, 2000.

[4] M. Sizintsev, S. Kuthirummal, S. Samarasekera, R. Kumar, H. S. Sawhney, and A. Chaudhry, "GPU accelerated realtime stereo for augmented reality," in *International Symposium 3D Data Processing, Visualization, and Transmission*, 2010.

[5] Z. Zhang, X. Ai, N. Canagarajah, and N. Dahnoun, "Local stereo disparity estimation with novel cost aggregation for sub-pixel accuracy improvement in automotive applications," in *IEEE Intelligent Vehicles Symposium*, pp. 99–104, 2012.

[6] E. Izquierdo and S. Kruse, "Image analysis for 3d modelling, rendering and virtual view generation," *Computer Vision and Image Understanding*, vol. 71, no. 2, pp. 231–253, 1998.

[7] E. Izquierdo, "Stereo image analysis for multi-viewpoint telepresence applications," *Signal Processing: Image Communication*, vol. 11, no. 3, pp. 231–254, 1998.

[8] S. Hermann and T. Vaudrey, "The gradient - a powerful and robust cost function for stereo matching," in *International Conference of Image and Vision Computing New Zealand*, pp. 1–8, 2010.

[9] H. Liu, Y. Liu, S. OuYang, C. Liu, and X. Li, "A novel method for stereo matching using gabor feature image and confidence mask," in *Visual Communications and Image Processing*, pp. 1–6, 2013.

[10] W. Fife and J. Archibald, "Improved census transforms for resource-optimized stereo vision," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 60–73, 2013.

[11] X. Sun, X. Mei, S. Jiao, M. Zhou, and H. Wang, "Stereo matching with reliable disparity propagation," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp. 132–139, 2011.

[12] C. Cigla and A. Alatan, "Information permeability for stereo matching," *Signal Processing: Image Communication*, vol. 28, no. 9, pp. 1072–1088, 2013.

[13] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *IEEE International Conference on Computer Vision Workshops*, pp. 467–474, 2011.

[14] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *International Conference on Pattern Recognition*, pp. 15–18, 2006.

[15] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013.

[16] Z. Lee, J. Juang, and T. Nguyen, "Local disparity estimation with three-moded cross census and advanced support weight," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1855–1864, 2013.

[17] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[18] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650–656, 2006.

[19] F. Tombari, S. Mattoccia, and L. Di Stefano, "Segmentation-based adaptive support for accurate stereo correspondence," in *IEEE Pacific-Rim Symposium on Image and Video Technology*, 2007.

[20] S. Mattoccia, F. Tombari, and L. Di Stefano, "Stereo vision enabling precise border localization within a scanline optimization framework," in *Asian Conference on Computer Vision*, pp. 517–527, 2007.

[21] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1073–1079, 2009.

[22] Q. Yang, "Recursive bilateral filtering," in *European Conference on Computer Vision*, pp. 399–413, 2012.

[23] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *International Conference on Computer Vision*, pp. 839–846, 1998.

[24] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European Conference on Computer Vision*, pp. 1–14, 2010.

[25] Q. Yang, P. Ji, D. Li, S. Yao, and M. Zhang, "Fast stereo matching using adaptive guided filtering," *Image and Vision Computing*, vol. 32, no. 3, pp. 202–211, 2014.

[26] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *IEEE International Conference on Computer Vision*, pp. 49–56, 2013.

[27] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *IEEE International Conference on Image Processing*, pp. 2093–2096, 2009.

[28] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *IEEE International Conference on Computer Vision*, vol. 2, pp. 508–515, 2001.

[29] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[30] Y.-C. Wang, C.-P. Tung, and P.-C. Chung, "Efficient disparity estimation using hierarchical bilateral disparity structure based graph cut algorithm with a foreground boundary refinement mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 5, pp. 784–801, 2013.

[31] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[32] S. Hermann and R. Klette, "Iterative semi-global matching for robust driver assistance systems," in *Asian Conference on Computer Vision*, pp. 465–478, 2012.

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

[34] C. Banz, S. Hesselbarth, H. F. Blume, and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation," *High-Performance Embedded Architectures and Compilers*, 2011.

[35] A. Akin, I. Baz, A. Schmid, and Y. Leblebici, "Dynamically adaptive real-time disparity estimation hardware using iterative refinement," *Integration, the VLSI journal*, vol. 47, no. 3, pp. 365–376, 2014.

[36] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[37] J. Kowalczuk, E. T. Psota, and L. C. Prez, "Real-time stereo matching on cuda using an iterative refinement method for adaptive support-weight correspondences," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 1, pp. 94–104, 2013.

[38] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza, "Efficient aggregation via iterative block-based adapting support-weights," in *International Conference on 3D Imaging*, 2011.

[39] I. Jung, T. Chung, J. Sim, and C. Kim, "Consistent stereo matching under varying radiometric conditions," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 56–69, 2013.

[40] G. Saygili, L. van der Maaten, and E. A. Hendriks, "Feature-based stereo matching using graph cuts," in *ASCI-IPA-SIKS Tracks*, 2011.

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

[42] T. Liu, P. Zhang, and L. Luo, "Dense stereo correspondence with contrast context histogram, segmentation-based two-pass aggregation

and occlusion handling," in *IEEE Pacific-Rim Symposium on Image and Video Technology*, 2009.

[43] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, pp. 269–293, 1999.

[44] G. Kordelas, D. Alexiadis, P. Daras, and E. Izquierdo, "Revisiting guided image filter based stereo matching and scanline optimization for improved disparity estimation," in *IEEE International Conference on Image Processing*, pp. 3803–3807, 2014.

[45] M. Mozerov and J. van de Weijer, "Accurate stereo matching by two-step energy minimization," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1153–1163, 2015.

[46] J. Liu, C. Li, F. Mei, and Z. Wang, "3d entity-based stereo matching with ground control points and joint second-order smoothness prior," *The Visual Computer*, vol. 31, no. 9, pp. 1253–1269, 2015.

[47] G. Kordelas, D. Alexiadis, P. Daras, and E. Izquierdo, "Enhanced disparity estimation in stereo images," *Image and Vision Computing*, vol. 35, pp. 31–49, 2015.

[48] Middlebury Stereo Evaluation:, "http://vision.middlebury.edu/stereo/eval/."

[49] C. Banz, H. Blume, and P. Pirsch, "Real-time semi-global matching disparity estimation on the gpu," in *IEEE International Conference on Computer Vision Workshops*, pp. 514–521, 2011.

[50] Y. Zhan, Y. Gu, K. Huang, C. Zhang, and K. Hu, "Accurate image-guided stereo matching with efficient matching cost and disparity refinement," *submitted to IEEE Transactions on Circuits and Systems for Video Technology*, 2015.

[51] Y. Peng, G. Li, R. Wang, and W. Wang, "Stereo matching with space-constrained cost aggregation and segmentation-based disparity refinement," in *SPIE, Three-Dimensional Image Processing, Measurement*, 2015.

[52] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.