Linear stereo matching

Leonardo De-Maeztu[1] Stefano Mattoccia[2] Arantxa Villanueva[1] Rafael Cabeza[1]
[1]Public University of Navarre
Pamplona, Spain
{leonardo.demaeztu,avilla,rcabeza}@unavarra.es
[2]University of Bologna
Bologna, Italy
stefano.mattoccia@unibo.it

Abstract

Recent local stereo matching algorithms based on an adaptive-weight strategy achieve accuracy similar to global approaches. One of the major problems of these algorithms is that they are computationally expensive and this complexity increases proportionally to the window size. This paper proposes a novel cost aggregation step with complexity independent of the window size (i.e. O(1)) that outperforms state-of-the-art O(1) methods. Moreover, compared to other O(1) approaches, our method does not rely on integral histograms enabling aggregation using colour images instead of grayscale ones. Finally, to improve the results of the proposed algorithm a disparity refinement pipeline is also proposed. The overall algorithm produces results comparable to those of state-of-the-art stereo matching algorithms.

1. Introduction

Dense stereo matching algorithms aim at determining correspondences in two or more images of the same scene taken from different viewpoints. This topic was exhaustively reviewed in [17] and [19]. According to both, most stereo algorithms can be categorized in two major classes: local methods and global methods. Local approaches use information within a finite region around the pixel whose disparity is being computed. Global approaches incorporate explicit smoothness assumptions and determine all disparities simultaneously by applying energy minimization techniques.

Traditional local stereo matching algorithms are typically faster than global approaches and have a lower memory footprint. However, they also have reduced accuracy compared to global state-of-the-art algorithms. Recent local algorithms based on adaptive-weight [6, 22] produce similar results to those obtained using global optimization techniques. Unfortunately, the computational complexity of this type of local algorithms is high, and is quadratically related to the window size used to aggregate the matching costs. Recently, two similar solutions have been proposed that render the computation complexity independent of the size of the aggregation window [7, 23] (also referred as O(1) complexity). However, the proposed methods have some limitations (i.e. aggregation can only be performed using grayscale images) and they have significantly reduced accuracy compared to state-of-the-art adaptive-weight methods arising from the asymmetric nature of the aggregation strategy (i.e. only the reference image information is used for aggregating costs).

This paper proposes a novel O(1) costs aggregation strategy that thanks to its symmetric nature outperforms state-of-the-art O(1) costs aggregation methods. Moreover, compared to previous O(1) solutions, our proposal relies on a completely different approach that allows to aggregate costs using colour input stereo pairs (previous O(1) solutions aggregate costs using grayscale images because of memory footprint limitations). The symmetric and colour-based O(1) aggregation strategy proposed not only outperforms state-of-the-art O(1) algorithms, but it also produces results comparable to non-O(1) adaptive-weight stereo matching [22]. Thanks to the O(1) nature of our algorithm, when the size of the input stereo pair grows, our method enables a dramatic improvement in execution time compared to adaptive-weight stereo matching. Finally, we also propose a pipeline that combines two disparity refinement techniques, that allows to obtain results comparable to top-performing stereo matching algorithms.

The rest of this paper is organized as follows. In Section 2, we review state-of-the-art O(1) cost aggregation strategies and other non-O(1) methods based on the adaptive-weight strategy. In Section 3 we present our novel constant time aggregation strategy and in Section 4 the overall algorithm proposed in this paper is described. We report experimental results in Section 5 and we draw conclusions in Section 6.

2. Related work

In this Section we split the review of state-of-the-art stereo matching algorithms in non-O(1) and O(1) approaches.

2.1. Non-O(1) cost aggregation strategies based on adaptive-weights

Traditional local stereo matching algorithms produce less accurate results compared to global ones. This gap has been reduced [18, 19] by recent local stereo algorithms [2, 21]. Even if the idea of aggregating costs using adapting weights for each pixel had been studied in several publications [8, 14], it was in [22] where Yoon and Kweon proposed the adaptive-weight aggregation method which clearly outperformed previous local stereo matching algorithms. It consists in aggregating costs over a fixed-size window (35×35 in [22]), where each pixel adds its cost to the total cost of the window with a different weight. The weight encodes the likelihood that each pixel q belongs to the same object than the central pixel of the window p. The pixel weight depends on two values: the colour difference and the distance between this pixel and the central pixel of the window. The weight is higher for pixels of a colour similar to the central one as well as for pixels closer to the central one. The colour difference (Δc_{pq}) is computed as the Euclidean distance between the values in the CIELab colour space of pixel p and pixel q. The distance (Δg_{pq}) is computed as the spatial Euclidean distance between pixel p and pixel q. Two constants, γ_c and γ_p, respectively, are used to modulate the relative relevance of Δc_{pq} and Δg_{pq}. The weight, w, assigned to the cost of pixel q when the disparity of pixel p is being computed is expressed as

$$w(p,q) = \exp\left(-\left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p}\right)\right) \tag{1}$$

During the aggregation step, the weights computed for the pixels in the reference window and the pixels in the target window are combined (symmetric aggregation). Then the dissimilarity within a square window, E, between the pixels p and $\overline{p}_d$ can be expressed as

$$E(p,\overline{p}_d) = \frac{\sum_{q \epsilon N_p, \overline{q}_d \epsilon N_{\overline{p}_d}} w(p,q) w(\overline{p}_d, \overline{q}_d) e(q, \overline{q}_d)}{\sum_{q \epsilon N_p, \overline{q}_d \epsilon N_{\overline{p}_d}} w(p,q) w(\overline{p}_d, \overline{q}_d)} \tag{2}$$

where $\overline{p}_d$ and $\overline{q}_d$ are the corresponding pixels in the target image when the pixels p and q in the reference image have a disparity value of d, N_p and $N_{\overline{p}_d}$ are the aggregation windows, and $e(q, \overline{q}_d)$ represents the pixel-based raw matching cost of q and $\overline{q}_d$.

One of the main disadvantages of adaptive-weight aggregation is that it is computationally expensive. Moreover, as it was mentioned in Section 1, computation time is quadratically related to the window size. Several authors have proposed faster local methods inspired by the adaptive-weight algorithm. Gong et al. [2] proposed two modifications to the original adaptive-weight algorithm. First, they only use the weight term obtained using the target image. Second, a two-pass 1D algorithm is implemented instead of using a 2D square window for aggregation. Richardt et al. [16] recently proposed a new real-time local stereo algorithm based on the adaptive-weight method. They use a dual-cross-bilateral grid for costs aggregation, an extension of the bilateral grid used to speedup bilateral filtering [12]. However, the results of both algorithms [2, 16] are less accurate than those of the adaptive-weight algorithm [22]. Fast Bilateral Stereo [11] is an algorithm that combines the efficiency of *integral images*, deployed by traditional *correlative* approaches, with an adapting-weight strategy applied on a block basis. Compared to the original adaptive-weight algorithm, the execution time is significantly reduced with comparable results.

All the techniques described in the previous paragraph are faster than the adaptive-weight algorithm. However, the execution time of these techniques still depends on the aggregation window size.

2.2. O(1) cost aggregation strategies

Ju and Kang [7] propose an alternative O(1) implementation of adaptive-weight aggregation. They use integral histograms to render the computation time independent of the size of the aggregation window using an approach inspired by Porikli's O(1) technique [13]. Ju and Kang [7] define the integral histogram $H^d_{(I,\overline{I})}(p_x, p_y, i)$ for the difference image with disparity d between the grayscale versions of reference image I and target image $\overline{I}$ as

$$\begin{aligned} H^d_{(I,\overline{I})}(p_x,p_y,i) = {} & H^d_{(I,\overline{I})}(p_x-1,p_y,i) \\ & + H^d_{(I,\overline{I})}(p_x,p_y-1,i) \\ & - H^d_{(I,\overline{I})}(p_x-1,p_y-1,i) \\ & + \left|I(p_x,p_y) - \overline{I}(p_x-d,p_y)\right| \end{aligned} \tag{3}$$

where the point p is represented by its coordinates p_x and p_y. i represents all the possible bins of the integral histogram.

To obtain not only a constant time algorithm but also a faster algorithm than the original adaptive-weight algorithm, Ju and Kang [7] perform some simplifications. First, only the weights in the reference image are used. Second, grayscale input images are used for costs aggregation instead of colour ones due to memory footprint constraints [7]. Finally, spatial filtering is performed separately from the colour filtering. According to the definition of the integral histogram and the simplifications previously mentioned, the constant time aggregation [7] can be expressed as

$$E(p,\overline{p}_d) = \frac{\sum_{i=0}^{N-1} w(p,i) H^d_{(I,\overline{I})}(p,i)}{\sum_{i=0}^{N-1} w(p,i)} \tag{4}$$

where N is the number of bins of the histogram and $w(p,i)$ is the resulting weight of comparing the value of the pixel p and the value of the bin i. It can be observed in (4) that aggregation is performed using the values of the histogram instead of the input stereo pair. This is the reason why the computation complexity of the algorithm is independent of the window size.

Recently, Zhang et al. [23] proposed a new structure, the joint integral histogram (JIH) , that accelerates weighted filtering and adaptive-weight stereo matching. Apart from this new structure, the proposed algorithm is similar to the previously described O(1) algorithm [7]. Finally, we report that in parallel to our activity, Rhemann et al. [15] developed an asymmetric aggregation strategy based on a similar linear model.

3. Proposed linear stereo aggregation step

In this Section we describe our O(1) aggregation step that, differently by other O(1) approaches does not rely on the Porikli's method [13]. For stereo matching algorithms, it is desirable that the edges of the input stereo pair of images are preserved during costs aggregation, as it is made in the adaptive-weight algorithms, where pixels with different colour values are aggregated with a low weight (1). Local linear models have been proven useful in image filtering [4], image matting [9], image super-resolution [24], and haze removal [3]. In these models, each pixel p of the output image O is supposed to be linearly related to the same pixel p of the input image I in a window N_q centered at the pixel q

$$O_p = a_q I_p + b_q, \forall p \epsilon N_q, \tag{5}$$

where a_q and b_q are constant parameters in each window N_q. This linear relationship ensures the conservation of edges between the input and the output image [4]. To model this behavior, in our linear stereo algorithm, the aggregated cost E is linearly related to the input stereo pair composed of the reference image I and the target image $\overline{I}$ as

$$E(p,\overline{p}_d) = \mathbf{a}_q \begin{bmatrix} \mathbf{I}(p)^T \\ \overline{\mathbf{I}}(\overline{p}_d)^T \end{bmatrix} + b_q \tag{6}$$

where $\mathbf{I}(p)$ and $\overline{\mathbf{I}}(\overline{p}_d)$ are intensity values if the algorithm is being applied to grayscale images or RGB components if the algorithm is being applied to colour images, $\mathbf{a}_q$ is a 2×1 or a 6×1 vector (depending on whether the input stereo pair contains grayscale or colour information, respectively) and $E(p,\overline{p}_d)$ and b_q are scalars.

Minimizing the difference between the input and the output of the linear model, the coefficients of the model are determined. According to [4] the parameters can be expressed as

$$\mathbf{a}_q = (\mathbf{\Sigma}_q + \epsilon \mathbf{U})^{-1} \left(\frac{1}{|N|} \sum_{p \epsilon N_q} \left[\mathbf{I}(p) \overline{\mathbf{I}}(\overline{p}_d) \right] e(p,\overline{p}_d) - \mu_q \overline{e}_q \right) \tag{7}$$

$$b_q = \overline{e}_q - \mathbf{a}_q^T \mu_q \tag{8}$$

where $\mathbf{\Sigma}_q$ is the 2×2 or 6×6 covariance matrix of the $\left[\mathbf{I}(p)\overline{\mathbf{I}}(\overline{p}_d)\right]$ vector in N_q and $\mathbf{U}$ is a 2×2 or 6×6 identity matrix. μ_q is the 2×1 or 6×1 mean vector of the $\left[\mathbf{I}(p)\overline{\mathbf{I}}(\overline{p}_d)\right]$ vector in N_q. $\overline{e}_q$ is the mean of e in N_q. $|N|$ is the number of pixels in N_q. ϵ is a regularization parameter preventing $\mathbf{a}_q$ from being too large. ϵ has a similar effect to σ_c in the adaptive-weight algorithm [22]. The larger the ϵ value, the stronger the filtering effect on edges.

After applying this linear model to the entire input stereo pair, a pixel p is involved in all the windows N_q that contain p. In other words, the aggregated cost of each pixel is modeled by $|N|$ different $\mathbf{a}_q$ and b_q values. Once computed all the $\mathbf{a}_q$ and b_q values, we average them within N_q. After this final modification, the proposed stereo linear model can be expressed as

$$\begin{aligned} E(p,\overline{p}_d) &= \frac{1}{|N|} \sum_{q \ni p \epsilon N_q} \left(\mathbf{a}_q \begin{bmatrix} \mathbf{I}(p)^T \\ \overline{\mathbf{I}}(\overline{p}_d)^T \end{bmatrix} + b_q \right) \\ &= \overline{\mathbf{a}}_p \begin{bmatrix} \mathbf{I}(p)^T \\ \overline{\mathbf{I}}(\overline{p}_d)^T \end{bmatrix} + \overline{b}_p \end{aligned} \tag{9}$$

The behavior of the proposed aggregation algorithm can be explained using Equations (7) and (8). In fact, in highly textured regions, the value of the $\mathbf{a}_q$ parameter is close to 1 and b_q tends to 0 (i.e. no filtering is performed). In low texture regions, $\mathbf{a}_q = 0$ and $b_q = \overline{e}_q$ (i.e. our linear model computes a simple square window averaging).

As it was previously mentioned in this paper, Equation (9) used for costs aggregation, was inspired by guided filtering [4]. Similarly, as pointed out in [19, 22], the idea behind the adaptive-weight aggregation strategy [22] was related to bilateral filtering [20]. In fact, our proposal and the adaptive-weight technique aim to average costs only in regions with similar colours.

It is worth noting that all the window-based computations described in this Section can be computed in constant time by means of the integral image method [1]. As a consequence, the computation time of the linear stereo aggregation stage is independent of the size of the window.

4. Stereo matching algorithm

4.1. Disparity computation and disparity selection

In Section 3 a novel O(1) costs aggregation step for stereo matching was proposed. According to [17] and [19]

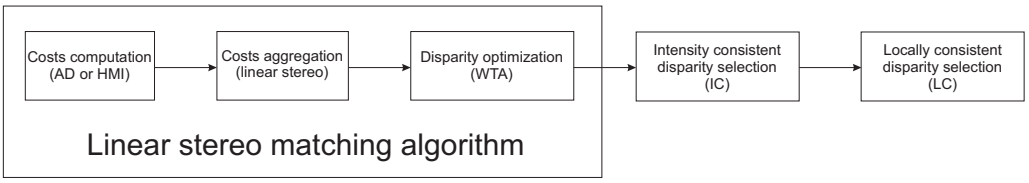


Figure 1. Structure of the proposed linear stereo algorithm and the disparity refinement pipeline

most dense stereo algorithms perform four steps: (a) *costs computation*, (b) *costs aggregation*, (c) *disparity optimization* and (d) *disparity refinement*. In this Subsection, we present steps (a) and (c), that combined with the linear stereo aggregation step described in Section 3 result in our overall proposed stereo matching algorithm.

For what concerns costs computation (a), two different measures are deployed in this paper: Absolute Difference (AD) and Hierarchical Mutual Information (HMI) [5]. In the case of colour images, AD is expressed as

$$e(q, \overline{q}_d) = \sum_{c \epsilon \{r,g,b\}} |I_c(q) - I_c(\overline{q}_d)| \qquad (10)$$

where I is the intensity value of pixel p in the reference image and $\overline{I}$ is the intensity value of pixel $\overline{p}_d$ in the target image. HMI is a matching measure defined from the entropy of two images (their information content). It is a more robust measure against radiometric distortion that AD [5]. The reader can find a detailed description of HMI in [5].

For what concerns disparity optimization (c), we use the simple winner-takes-all strategy (WTA). We obtain subpixel accuracy by fitting a parabola to the direct neighbors of the best disparity found [5].

4.2. Disparity refinement

In this paper, as depicted in Figure 1, we also propose a novel combination of two existing disparity refinement techniques [5] and [10]. The combination of these two techniques allows us to remove large errors and to enforce local consistency in the disparity results.

The intensity consistent (IC) disparity selection technique proposed in [5] (Section 2.5.2), relies on segmentation and is particularly effective to solve the problem of propagation of disparities from textured foregrounds to untextured backgrounds as well as to assign disparities to large untextured regions, one of the major problems of local stereo matching algorithms.

The locally consistent (LC) disparity selection technique [10], by enforcing local consistency between neighboring points, has proven to be effective in recovering wrong disparity assignments in uniform regions as well near depth discontinuities. Nevertheless, this technique is unable to recover from large erroneous areas (i.e. the erroneous patches typically caused by homogeneous regions in the stereo pair). Therefore, we propose a disparity refinement pipeline in which the resulting disparity maps of linear stereo are refined with method [5] to solve large erroneous areas, and then refined enforcing local consistency by means of [10].

A detailed description of these two disparity refinement techniques can be found in [5] and [10].

5. Experimental results

5.1. Accuracy comparison

In this Subsection we evaluate the performance of our proposal and the performance of constant time O(1) stereo matching [7, 23] and adaptive-weight stereo matching [22] according to the metric used in the Middlebury ranking [18] and in [17]. For the three algorithms we use our own C implementation. The parameters of the three algorithms are chosen according to an optimization process aiming at obtaining the minimum average percentage of erroneous pixels in the tested stereo dataset. In this Section we perform three different tests. The first uses colour stereo pairs for costs computation and grayscale stereo pairs for costs aggregation (the reason of this choice if the aforementioned intrinsic limitation of the constant time algorithms proposed by [7, 23], which due to memory limitations cannot be executed on colour images). In this case, the three algorithms are referred as G-LinearS, G-ConsT and G-AdaptW. The second test evaluates the performance of linear stereo and adaptive-weight with simple pre and post-processing techniques on colour stereo pairs (both for costs computation and costs aggregation). In this test, algorithms are referred as LinearS and AdaptW (we do not evaluate ConstT in this experiment because, as it was previously mentioned, it cannot be executed on colour images due to a very large memory footprint). Finally we perform a third test in which the algorithms in the second test are refined with the IC and LC techniques described in Subsection 4.2. Those algorithms are referred as P-LinearS and P-AdaptW.

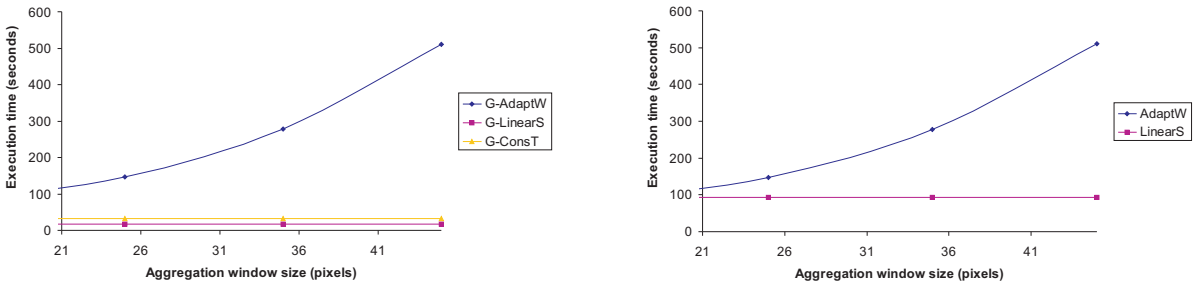
The Middlebury stereo evaluation website ([18]) provides a standard way to evaluate the accuracy of the reconstruction by the percentage of bad pixels in the computation of four disparity maps using four stereo pairs named Tsukuba, Venus, Teddy and Cones (see Figure 3). In fact, it is the implicit benchmark for stereo matching algorithms, as most papers evaluate their results according to these four stereo pairs.

Table 1. Performance comparison of aggregation methods using colour images for costs computation and grayscale images for costs aggregation and no pre or post-processing.

Algorithm	Tsukuba			Venus			Teddy			Cones			Average percent of bad pixels
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
G-AdaptW	3.94	5.83	16.0	1.73	3.31	11.2	11.2	20.2	22.5	5.83	16.0	13.7	10.9
G-LinearS	**3.93**	**5.74**	**15.4**	**3.23**	**4.78**	**22.1**	**12.2**	**21.1**	**25.5**	**4.35**	**14.8**	**11.3**	**12.0**
G-ConsT	5.70	7.18	23.2	4.77	6.15	25.5	13.7	22.3	28.8	6.68	16.0	17.0	14.8

Table 2. Performance comparison of aggregation methods using colour images for costs computation and costs aggregation, pre and post-processing.

Algorithm	Tsukuba			Venus			Teddy			Cones			Average percent of bad pixels
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
AdaptW	3.46	4.06	8.90	0.92	1.49	8.67	7.53	14.1	17.2	2.55	8.03	7.24	7.01
LinearS	**3.63**	**4.39**	**9.61**	**2.10**	**2.81**	**17.0**	**9.14**	**15.5**	**21.1**	**2.84**	**8.53**	**8.15**	**8.73**
P-AdaptW	1.62	2.09	5.78	0.18	0.36	2.16	6.37	11.6	14.9	2.87	8.80	7.14	5.33
P-LinearS	**1.10**	**1.67**	**5.92**	**0.53**	**0.89**	**5.71**	**6.69**	**12.0**	**15.9**	**2.60**	**8.44**	**6.71**	**5.68**



(a) (b)

Figure 2. Execution time of the studied algorithms on the Teddy stereo pair using: (a) grayscale images and (b) colour images for aggregation.

We have evaluated constant time O(1) stereo matching according to the Middlebury benchmark and compared it with linear stereo and adaptive-weight. For a fair comparison, we use the same experimental setup used in the papers [7, 23]: no pre or post-processing is implemented, colour stereo pairs are used for costs computation (AD matching metric), grayscale stereo pairs are used in the aggregation stage since the computation of the constant time algorithm using colour images requires a very large amount of memory that is not available in today's computers [7]. The optimal set of parameter values used in this test is 35×35 aggregation window with $\gamma_p = 10$ and $\gamma_c = 20$ for G-AdaptW, 35×35 aggregation window with $\gamma_c = 9$ for G-ConsT and $\epsilon = 10^{-2.75}$ and $r = 10$ for G-LinearS. Under these circumstances, the results of G-ConsT [7, 23], G-AdaptW [22] and the proposed G-LinearS are reported in Table 1. Table 1 reports that G-AdaptW and G-LinearS outperform on the whole dataset G-ConsT. This can be explained observing that the constant time algorithm computes weights asymmetrically deploying only the reference image. On the other end, linear stereo and adaptive-weight deploy the reference and the target image. Table 1 also reports that the performance of linear stereo performance is not as good as adaptive-weight on grayscale images with AD. Although the overall performance of G-AdaptW is better than G-LinearS, our proposal outperforms G-AdaptW in 6 out of 12 cases.

In Figure 2a, the execution time of G-AdaptW, G-LinearS and G-ConsT on the Teddy stereo pair is represented versus the window size. Execution time measurements were performed on a Intel Core 2 6420 processor using only one core. It can be observed that the proposed G-LinearS algorithm is much faster than G-AdaptW and also it outperforms G-ConstT in terms of execution time. As

proposed in [23], G-ConsT could be speeded up by quantizing the grayscale levels used for aggregation. However, according to the results presented in [23], the accuracy of the algorithm would further decrease, being a much less accurate algorithm than G-AdaptW or G-LinearS.

In order to provide a more detailed comparison of linear stereo and adaptive-weight, we evaluate their performance according to the Middlebury benchmark using colour images both for costs computation and costs aggregation. Experimentally we found that for both algorithms preprocessing the input images with a 5×5 bilateral filter [20] with parameters $\sigma_c = 10$ and $\sigma_s = 10$ only for costs aggregation provides optimal results. Moreover we postprocess the resulting disparity maps of both algorithms applying cross-checking, a 3×3 median filtering to the resulting disparity maps and eliminating constant disparity blobs smaller than 80 pixels. The disparities in these invalidated areas are filled in with the content of the first non-invalidated pixel to the left or to the right (according to the nature of the occlusion). For both algorithms we use the optimal set of parameters empirically found, 35×35 aggregation window with $\gamma_p = 26$ and $\gamma_c = 6$ for AdaptW and $\epsilon = 10^{-4}$ and $r = 9$ for LinearS. HMI with $\sigma = 1$ is implemented for costs computation, as it produces better results than AD. Table 2 reports the experimental results for linear stereo (LinearS) and adaptive-weight (AdaptW) according to this setup. The resulting disparity maps are shown in Figure 4.

Table 2 shows that AdaptW produces slightly better results than LinearS. However, we can observe more significant differences on the discontinuity regions of Venus and Teddy. Despite this small difference, it is worth noting that the complexity of the costs aggregation step of linear stereo is independent of the window size. According to Figure 2b LinearS outperforms AdaptW in terms of execution time. Moreover, this advantage grows when the size of the support is increased.

To test the effectiveness of the disparity refinement technique proposed in Section 4.2, we perform a third experiment with LinearS and AdaptW implemented as described previously in this paper, deploying the disparity refinement step. These algorithms are referred as P-LinearS and P-AdaptW. For IC, the parameter values proposed in [5] are used. For LC, the chosen values are a 39×39 window with $\gamma_s = 22$, $\gamma_c = 23$, $\gamma_m = 5$ and $T = 60$ for P-LinearS and a 39×39 window with $\gamma_s = 13$, $\gamma_c = 35$, $\gamma_m = 8$ and $T = 50$ for P-AdaptW. The results of these algorithms can be found in Table 2. The resulting disparity maps are reported in Figure 5. Comparing the non refined (LinearS and AdaptW) and the refined (P-LinearS and P-AdaptW) versions of the algorithms, the results are notably improved showing that the disparity refinement pipeline is very effective. The disparity refinement pipeline is more effective for the LinearS algorithm, since we can observe that the results of P-LinearS and P-AdaptW are very similar (P-LinearS even outperforms P-AdaptW in 5 out of 12 results).

Observing the overall results, the proposed P-LinearS method has results comparable to state-of-the-art algorithms according to the Middlebury ranking [18] [1]. The complete proposed pipeline is composed of a local stereo matching algorithm whose execution time is independent of the size of the window (15 seconds for Tsukuba and 94 seconds for Teddy) and a disparity refinement step based on the intensity consistent technique (IC) [5] (which runs in 10 second on Tsukuba and 30 seconds on Teddy) and the locally consistent technique (LC) [10] (which takes 8 seconds for Tsukuba and 20 seconds for Teddy). Execution time has been measured in our own implementation of both methods. However, both techniques as well as the proposed linear cost aggregation method could be implemented much more efficiently.

5.2. Scalability

The time measurements described in the previous Subsection are associated to the execution of algorithms using the Middlebury stereo pairs. The most time consuming stereo pairs in the Middlebury dataset are Teddy and Cones (450×375 resolution and 60 possible disparities). However, these two stereo pairs were obtained downsampling the original 1800×1500 images (240 disparities). Although the Middlebury benchmark uses relatively low resolution images, images of larger size are already quite common nowadays. When the resolution of the input stereo pairs is increased, the size of the aggregation window must be increased accordingly using the same scale factor. As a consequence, the use of O(1) aggregation techniques becomes even more advantageous. In Figure 6, it can be observed how the execution time ratio of AdaptW and LinearS increases as the resolution of the input stereo pair is increased. For a resolution of 1800×1500 pixels and 240 disparity levels, LinearS is around 50 times faster than AdaptW.

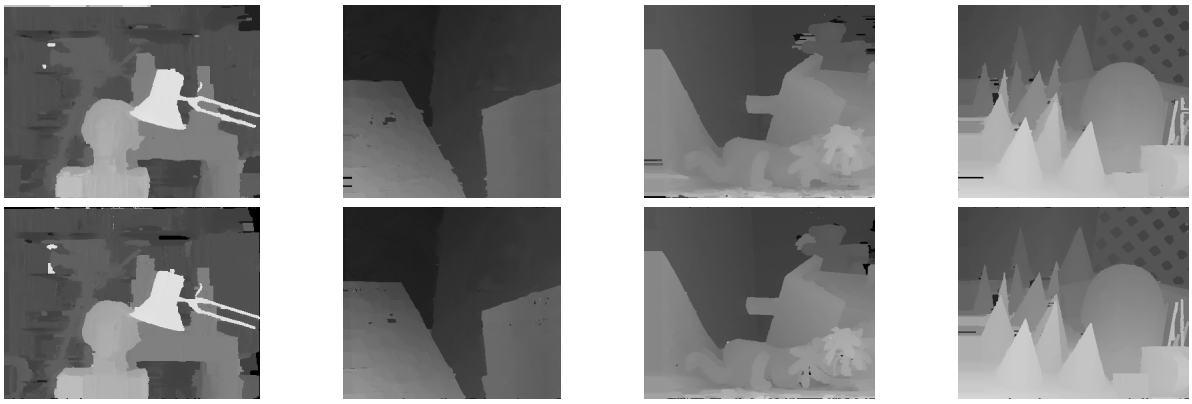
6. Conclusions

In this paper, we propose a new O(1) costs aggregation method based on a linear model. Compared to existing O(1) methods, our proposal does not rely on the Porikli's method [13]. To our knowledge, it is the first O(1) costs aggregation method which relies on a symmetric strategy. Moreover, unlike previous O(1) algorithms, aggregation based on colour images can be performed with a reasonable computational cost. Thanks to these two improvements (symmetric and colour-based aggregation) our linear stereo algorithm clearly outperforms previous O(1) stereo algorithms

[1] February 23^{rd} 2011 our P-LinearS algorithm classifies in position number 15 in the Middlebury ranking.

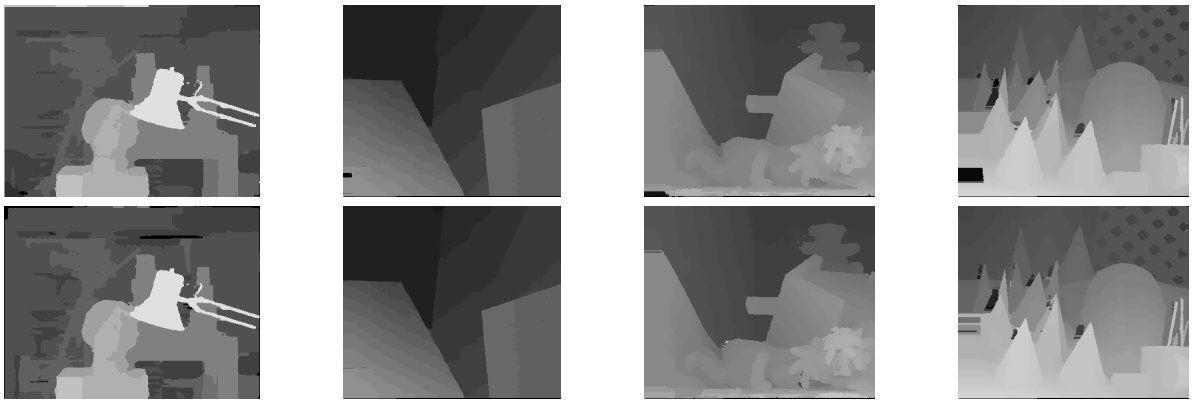
(a) (b) (c) (d)

Figure 3. Left image of each one of the Middlebury datasets. (a) "Tsukuba" images. (b) "Venus" images. (c) "Teddy" images. (d) "Cones" images.



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

Figure 4. Disparity maps computed by LinearS (upper row) and AdaptW (lower row). (a) "Tsukuba" images. (b) "Venus" images. (c) "Teddy" images. (d) "Cones" images.



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

Figure 5. Disparity maps computed by P-LinearS (upper row) and P-AdaptW (lower row). (a) "Tsukuba" images. (b) "Venus" images. (c) "Teddy" images. (d) "Cones" images.

in terms of accuracy of the results. Not only our algorithm outperforms previous O(1) solutions, but experimental results show that the proposed method produces accurate disparity maps comparable to those of adaptive-weight aggregation. However, compared to the adaptive-weight aggregation approach, our proposal has a computational complexity

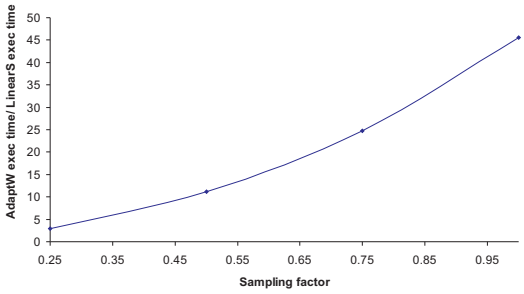


Figure 6. Execution time ratio of AdaptW and LinearS versus the sampling factor of the Teddy stereo pair (original resolution 1800×1500 pixels).

independent of the size of the window.

In this paper we have also proposed a disparity refinement strategy based on a combination of two already existing techniques. Each one of the techniques aim at solving common problems that arise in local stereo algorithms. The combination of these techniques provides, according to the Middlebury ranking, a notable improvement in accuracy of the resulting disparity maps. In fact, the overall proposed algorithm provides disparity maps comparable to state-of-the-art algorithms.

Finally, it has been proven that the O(1) nature of the algorithm makes it highly scalable for future applications using higher resolution stereo pairs. According to our tests, for high resolution images, our algorithm is around 50 times faster than adaptive-weight based stereo matching.

References

[1] F. Crow. Summed-area tables for texture mapping. *Proc. Special Interest Group on Graphics*, pages 217–212, 1984. 3

[2] M. Gong, R. Yang, L. Wang, and M. Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. *Intl J. Computer Vision*, 75(2):283–296, 2007. 2

[3] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In *Proc. IEEE Workshop Computer Vision and Pattern Recognition*, pages 1956–1963, 2009. 3

[4] K. He, J. Sun, and X. Tang. Guided image filtering. In *European Conf. Computer Vision*, pages 1–14, 2010. 3

[5] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 4, 6

[6] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *Proc. IEEE Intl Conf. on Image Processing*, pages 2093–2096, 2009. 1

[7] M. Ju and H. Kang. Constant time stereo matching. In *Machine Vision and Image Processing Conf.*, pages 13–17, 2009. 1, 2, 3, 4, 5

[8] R.-D. Lan and P. Remagnino. Robust matching by partial correlation. In *Proc. British Machine Vision Conf.*, pages 651–660, 1995. 2

[9] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008. 3

[10] S. Mattoccia. A locally global approach to stereo correspondence. In *Proc. IEEE Int. Workshop on 3D Digital Imaging and Modelling*, pages 1763–1770, 2009. 4, 6

[11] S. Mattoccia, S. Giardino, and A. Gambini. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering. In *Asian Conf. Computer Vision*, pages II: 371–380, 2009. 2

[12] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *Intl J. Computer Vision*, 81(1):24–52, 2009. 2

[13] F. Porikli. Constant time o(1) bilateral filtering. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2, 3, 6

[14] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52:93–99, 1985. 2

[15] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011. 3

[16] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *European Conf. Computer Vision*, volume 6313, pages 510–523. Springer Berling / Heidelberg, 2010. 2

[17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Intl J. Computer Vision*, 47(1-3):7–42, 2002. 1, 3, 4

[18] D. Scharstein and R. Szeliski. *Middlebury stereo evaluation - version 2*, http://vision.middlebury.edu/stereo/eval. 2, 4, 6

[19] R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2010. 1, 2, 3

[20] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. IEEE Intl Conf. on Computer Vision*, pages 839–846, 1998. 3, 6

[21] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008. 2

[22] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006. 1, 2, 3, 4, 5

[23] K. Zhang, G. Lafruit, R. Lauwereins, and L. Van Gool. Joint integral histograms and its application in stereo matching. In *Proc. IEEE Intl Conf. Image Processing*, pages 817–820, 2010. 1, 3, 4, 5, 6

[24] A. Zomet and S. Peleg. Multi-sensor super resolution. In *Proc. IEEE Workshop Applications of Computer Vision*, pages 27–31, 2002. 3