

Fast locally consistent dense stereo on multicore

Stefano Mattoccia

University of Bologna

Dipartimento di Elettronica Informatica e Sistemistica (DEIS)

Advanced Research Center on Electronic Systems (ARCES)

stefano.mattoccia@unibo.it

Abstract

Many computer vision applications require fast and accurate 3D measurements. However, despite the advent of powerful computing architectures (e.g., multicore CPU and GPU), most top-ranked dense stereo algorithms rely on global 2D disparity optimization methods that are often too slow for practical use. Moreover, their huge memory requirements are typically not suited to devices with constrained resources (e.g., FPGA). Nevertheless, algorithms based on 1D disparity optimization methods (i.e., *Dynamic Programming* and *Scanline Optimization*) provide a good trade-off between accuracy and efficiency with a limited memory footprint. In this paper, we show that enforcing a relaxed local consistency constraint to the disparity fields, provided by fast 1D disparity optimization methods, yields much more rapidly, results comparable to those of the top-ranked approaches. The simple and non-iterative computational structure of our proposal enables us to exploit coarse grained parallelism on multicore CPUs. Moreover, due to its limited memory footprint, our proposal could be potentially mapped on devices, such as FPGA, with constrained resources.

1. Introduction

Many computer vision applications require accurate and dense 3D measurements in real-time. Stereo vision, compared to active sensors (e.g., Time of Flight), is passive and holds the potential to obtain more accurate results. According to [17] most dense stereo algorithms perform: costs computation, costs aggregation, disparity optimization and disparity refinement.

In *local* algorithms the focus is on cost aggregation [20] and disparity optimization is typically a simple Winner Takes All (WTA) strategy. These algorithms have a simple and regular computational structure that allows, in some cases, for fast/real-time implementations on standard

PCs and their limited memory footprint renders these approaches suited to devices with constrained resources (e.g. FPGA, embedded devices). Moreover, because of their local computational structure, these algorithms can also easily take advantage of the multicore architectures that are currently available in most CPUs (e.g. Intel, AMD, ARM, IBM, SPARC, Analog Devices (Blackfin)) deployed in desktop computers, mobile devices, game consoles, etc. Although taking advantage of the parallel capabilities of multicore CPU is, generally, complicated, the simple computational structure of most local algorithms easily enables the exploitation of thread-level parallelism (e.g., distributing computations among different cores (CPU) or processing elements (GPU)) as well as data-level parallelism (i.e., executing the same operation on multiple data by means of SIMD instructions). Nevertheless, as reported in [16], in most cases, local algorithms are outperformed in terms of accuracy by *global* algorithms.

Global algorithms typically enforce the smoothness assumption in vertical and horizontal directions deploying a 2D disparity optimization method. Unfortunately, these methods are iterative and typically too slow for most practical applications. Moreover, their huge memory footprint typically renders these algorithms inappropriate for devices with constrained resources. Nevertheless, a subclass of global algorithms that enforce the smoothness term in 1D, such as *Dynamic Programming* (DP) [17, 23, 21] and *Scanline Optimization* (SO) [17, 8], allow the efficient realization of reasonably accurate results [17, 8, 23, 21, 4]. Moreover, some of these methods have a limited memory footprint suited to devices with constrained resources (e.g., [6]) and their computational structure may allow the exploitation of thread-level parallelism.

A different and non-iterative methodology to enforce the local coherence of disparity fields was proposed in [12]. This method, referred to as *Locally Consistent* (LC), explicitly models the mutual relationships among the disparity of neighboring points without performing disparity optimizations amenable to known approaches. In [13], it was

shown that enforcing the local consistency of the disparity fields provided by fast SO/DP algorithms by means of the LC technique allows significant improvement in the effectiveness of the original SO/DP algorithms, yielding results comparable to 2D disparity optimization methods. Unfortunately, enforcing local consistency using method [13] results in an execution time of approximately 15 seconds, which is not suitable for most practical applications.

In this paper, we propose to enforce a *relaxed* local consistency constraint that, according to our experiments, yields, in most cases, results comparable to those of [13] with a significantly reduced computational effort. Moreover, the computational structure of our proposal easily enables us to obtain a further significant speed-up, taking advantage of coarse-grained thread-level parallelism on multicore architectures. The paper is organized as follows. In the next section, we review relevant previous work concerned with dense stereo algorithms (focusing on approaches based on 1D disparity optimization methods), the LC technique [12], and the LC-based framework proposed in [13]. Afterwards, we describe the proposed optimization techniques. Finally, we assess the effectiveness of our proposal, providing detailed experimental results on the standard Middlebury dataset, deploying a standard multicore off-the-shelf architecture.

2. Related work

Although local algorithms [20, 17] based on the *adaptive weights* approach [28] yielded excellent results [28, 14, 9], according to [17, 16], most of the top-performing stereo algorithms rely on global approaches (e.g., [10, 24, 26, 25, 27, 2, 22, 19]). Given a rectified stereo pair, these latter methods solve the correspondence problem in terms of a pixel-labeling assignment of disparities, determining the disparity field D that minimizes

$$E(D) = E_{data}(D) + E_{smooth}(D) \quad (1)$$

The *data term* E_{data} in (1) encodes how well the disparity assignment fits with the stereo pair and, often, it is the sum of per-pixel data costs $C(D(p))$ between one point in the reference image R and the supposed correspondent point in the target image T

$$E_{data}(D(p)) = \sum_{p \in R} C(D(p)) \quad (2)$$

In some cases (e.g., [26]), the point-wise cost $C(D(p))$ is replaced by an aggregated cost computed over neighboring pixels deploying support regions that adapt their shape to image content. In (1), the *smoothness term* $E_{smooth}(D)$ enables to enforce that the disparity field D is piecewise smooth penalizing disparity changes. To this aim, the smoothness term models the interaction between each point

p and its neighboring points $q \in \mathcal{N}(p)$. In global approaches, $\mathcal{N}(p)$ includes points in vertical and horizontal directions (typically, the four nearest neighbors of p on the pixel grid) while in SO/DP-based approaches the smoothness term is enforced only in one direction (typically $\mathcal{N}(p)$ includes only one point along a scanline). The former disparity optimization methods are typically referred to as 2D, while the latter are referred to as 1D. In general, 2D methods perform better as they enable the enforcement of *inter* and *intra* scanline smoothness assumptions. Unfortunately, when deploying a 2D approach, the minimization of (1) turns out to be an \mathcal{NP} -hard problem. Therefore, global approaches typically rely, under particular hypotheses [18] on (1), on efficient energy minimization strategies based on Graph Cuts (GC) or Belief Propagation (BP). Nevertheless, in most cases, these methods are too slow for practical use. Moreover, their iterative computational structures and their huge memory requirements typically render these approaches inappropriate for devices with limited resources.

On the other hand, SO [17] and DP [1, 17]-based approaches perform a 1D optimization typically restricted to individual scanlines, which allows to efficiently minimize (1) in polynomial time. Moreover, compared to 2D disparity optimization methods, the memory footprint is quite low (proportional to $d_{max} - d_{min}$ for SO and to $L \times (d_{max} - d_{min})$ (L being the image width) for DP). Although often violated in practice, the DP-based approach also enables to enforce the *ordering constraint*. Unfortunately, the 1D optimization adopted by basic SO and DP approaches leads to the well-known *streaking* effect. To overcome this problem, the key idea is to enforce constraints incoming from regions outside the individual scanline. Basically, three approaches have been proposed so far to deal with this problem. The first approach [23], referred to as RTGPU, uses as data term the cost computed over a vertical support by means of the approximated *adaptive weight* approach [28] in order to include cues from neighboring scanlines. RTGPU runs in real-time, computing the aggregated cost on a GPU and the DP-based optimization on the CPU. A second approach [8] aims to reduce the streaking effect, combining individual SOs computed along multiple directions (typically 8 or 16). This method, referred to as C-Semiglobal, turns out to be the best performing SO/DP based approach on the Middlebury evaluation site and has a running time of only a few seconds. This algorithm has also been effectively mapped on FPGA [6] and GPU [5]. Compared to the original Semiglobal approach [7], C-Semiglobal includes a disparity refinement step based on the Mean Shift algorithm [3]. A third approach aims to avoid the streaking effect by performing 1D disparity optimization on a tree rather than on a scanline. In [21], DP-based optimization is performed on a tree structure made of the *most important* edges. This approach was improved by replacing edges with segments:

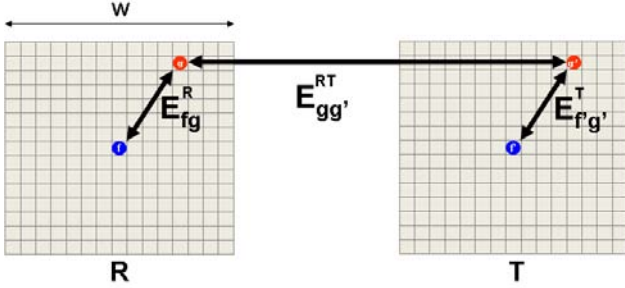


Figure 1. LC technique: in local algorithms once a disparity assumption for the central points (f and f' , blue) has been set, the plausibility of the same disparity assumption of each point within the *active* support (g and g' , red) is modeled according to the three events E_{fg}^R , $E_{f'g'}^T$ and $E_{gg'}^{RT}$ depicted in the figure. **[Best viewed with colors]**

[11] deploys the Mean Shift algorithm [3], while [4] deploys a fast line segmentation algorithm. This latter class of approaches is quite fast: [21] and [4] runs in a fraction of a second, while [11] runs in a few seconds on standard PCs.

A different methodology to enforce local consistency of disparity fields was proposed in [12]. This method exploits the implicit assumptions made for cost aggregation. That is, deploying *fronto parallel* supports, as typically made by most local approaches, once a disparity has been set, each element of the support is implicitly assumed at the same disparity of the point under evaluation (*i.e.*, the central point). Therefore, the LC approach explicitly models, within the set of neighboring points referred to as the *active support*, the mutual relationships among neighboring pixels of a dense disparity field determining the accumulated posterior probability for each disparity hypothesis made for the points belonging to the support. This technique allows us to enforce that the disparity field is piecewise smooth by explicitly modeling this behavior according to the three events depicted in Figure 1. The two events, E_{fg}^R and $E_{f'g'}^T$, model the *intra-image* color and spatial proximity constraints while $E_{gg'}^{RT}$ models the *inter-image* color proximity constraint (see [12] for details). Therefore, once a disparity assumption for the central point of the *active support* is made, the plausibility (*i.e.*, the posterior probability of the three joint events) of the same implicit disparity assumption for each point within the support is computed and propagated towards neighboring points. For each point of the image, the accumulated plausibility received encodes the belief for each disparity hypothesis.

In [13], an LC-based framework was proposed, aimed at enforcing the local consistency of the disparity hypotheses provided by fast SO/DP-based algorithms [8] and [23]. An overview of this approach is depicted in Figure 2 and can be summarized as follows. The R and T images of the stereo

pair and the initial disparity field D are processed by means of the LC technique. This enables us to obtain, in a single iteration of the LC technique, two accumulated plausibility distributions: one concerned with reference image R and one concerned with target image T. Choosing, for each point and for each distribution, the label with the higher plausibility yields to the two disparity fields referred to as D_R and D_T . Afterwards, D_R and D_T are cross-checked to detect unreliable disparity assumptions and a simple interpolation/refinement step is carried out so as to fill in the missing values. In [13], it was shown that, according to [16], this approach yields results comparable to the top-ranked approaches deploying the initial disparity fields provided by [8] or [23] available in [16]. The key components in [13] are the LC technique and the algorithm that provides the initial disparity hypotheses D . In this paper, we focus our attention on the original LC technique proposing three methods to speed up its computation, which enables us to much more efficiently obtain results comparable to those reported in [13].

3. Improving the efficiency of the LC technique

In this section, we propose three strategies that allow us to efficiently enforce the local consistency of disparity fields by means of the LC technique [12], deploying the framework described in [13]. Compared to [13], our proposal enables us to much more efficiently obtain similar or equivalent results, enforcing a *relaxed* local consistency constraint. In order to further increase the efficiency of our proposal, we also effectively exploit coarse-grained thread-level parallelism on off-the-shelf multicore architectures.

3.1. Relaxing local consistency

In order to evaluate how well a certain disparity assumption fits with those of neighboring points, the original LC technique [12] computes, for each point of the active support, the posterior probability $\mathcal{P}(E_{fg}^R, E_{f'g'}^T, E_{gg'}^{RT}(d) \mid \Delta_{fg}^\psi, \Delta_{f'g'}^\psi, \Delta_{gg'}^\omega)$ for the three joint events E_{fg}^R , $E_{f'g'}^T$, $E_{gg'}^{RT}(d)$ depicted in Figure 1 according to

$$\mathcal{P}(E_{fg}^R, E_{f'g'}^T, E_{gg'}^{RT}(d) \mid \Delta_{fg}^\psi, \Delta_{f'g'}^\psi, \Delta_{gg'}^\omega) \propto e^{-\frac{\Delta_{fg}}{\gamma_s}} \cdot e^{-\frac{\Delta_{fg}^\psi}{\gamma_c}} \cdot e^{-\frac{\Delta_{f'g'}}{\gamma_s}} \cdot e^{-\frac{\Delta_{f'g'}^\psi}{\gamma_c}} \cdot e^{-\frac{\Delta_{gg'}}{\gamma_t}} \quad (3)$$

Although plausibility computation (see [12] for a detailed description) is independent of the disparity range, its computation turns out to be demanding. In fact, with standard stereo pairs, plausibility computation accounts for about 90% of the time required by the overall LC technique (*i.e.*, 15 seconds on our platform with the Teddy stereo pair). Thus, in order to improve its efficiency, we analyzed the behavior of the LC technique, observing that a good trade-off

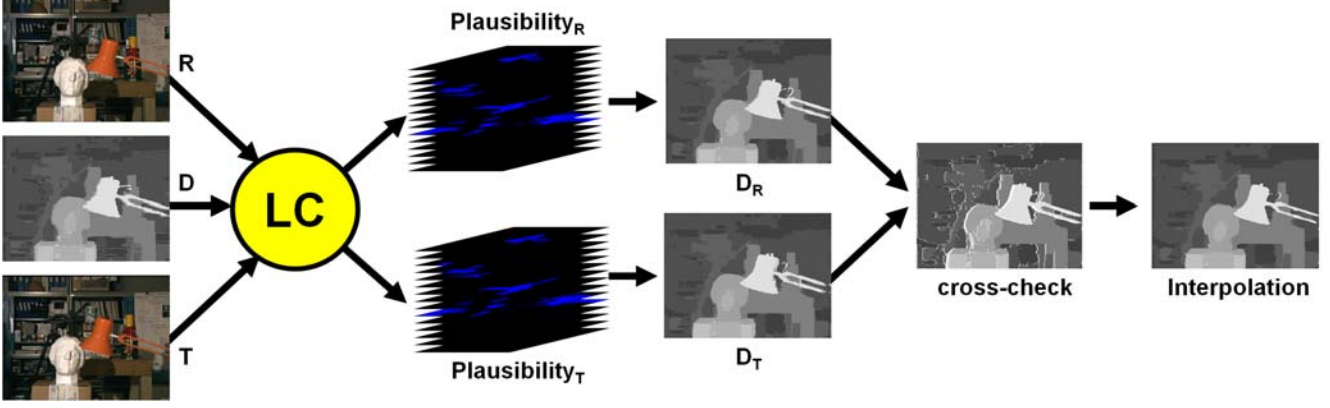


Figure 2. Overview of the LC-based approach proposed in [13].

between efficiency and accuracy could be obtained by relaxing plausibility computation (3). In particular, we observed that in (3) the spatial consistency constraints encoded by $e^{-\frac{\Delta_{f,g}}{\gamma_s}}$ and $e^{-\frac{\Delta_{f',g'}}{\gamma_s}}$ and the intra-image color proximity constraint encoded by $e^{-\frac{\Delta_{fg}^\psi}{\gamma_c}}$ and $e^{-\frac{\Delta_{f'g'}^\psi}{\gamma_c}}$ can be modeled more roughly, on a block basis, without significantly degrading the effectiveness of the original LC technique. Conversely, our analysis suggested that the posterior probability associated with event $E_{gg'}^{RT}(d)$ (i.e., encoded by the inter-image color proximity constraint $e^{-\frac{\Delta_{gg'}^\omega}{\gamma_t}}$) should be modeled deploying the original method, on a point basis, so as to not significantly reduce the effectiveness of the original LC approach. Thus, as depicted in Figure 3, we evaluate E_{fg}^R and $E_{f'g'}^T$ events on a block basis deploying the same cues $\Delta_{f,g}$ and $\Delta_{f',g'}$ within each block (top of Figure 3). Conversely, evaluation of $E_{gg'}^{RT}$ is carried out on a point basis, as originally proposed in [12] (bottom of Figure 3).

Therefore, by observing Figure 3, the posterior probability associated with the *approximated* events E_{fg}^R and $E_{f'g'}^T$, referred to as E_{*fg}^R and $E_{*f'g'}^T$, respectively, is obtained by

1. approximating for reference image R the spatial proximity constraints between points f and $g \in R$ with the Euclidean distance between f and the center of the block that contains g . The same approach applies to points f' and $g' \in T$, belonging to target image T.
2. approximating for reference image R the intra-image color proximity constraint between f and $g \in R$ according to the following cue:

$$\Delta_{*fg}^\psi = \sqrt{\sum_{c \in R,G,B} (I_c(f) - \bar{I}_c(g))^2} \quad (4)$$

where $\bar{I}_c(g)$ represents the average photometric intensity of channel c computed within a block of size $w \times w$

that contains g (see Figure 3). Similarly, for target image T, the approximated intra-image color proximity constraint between f' and $g' \in R$ is computed according to the following cue:

$$\Delta_{*f'g'}^\psi = \sqrt{\sum_{c \in R,G,B} (I_c(f') - \bar{I}_c(g'))^2} \quad (5)$$

The proposed block-basis approximation enables us to significantly reduce the number of expensive exponential computations. Moreover, it is noteworthy that average intensities $\bar{I}_c(g)$ and $\bar{I}_c(g')$ can be computed very efficiently by means of integral images. As previously indicated, we observed that modeling $E_{gg'}^{RT}(d)$ (i.e., the inter-image color proximity constraint) on a point basis enables us to limit the loss of accuracy due to the approximated computation described so far. A detailed analysis of this approximation will be provided in the experimental results section.

Therefore, according to our proposal, the relaxed plausibility of disparity d for points $g \in R$ and $g' \in T$ results in

$$\begin{aligned} & \mathcal{P}_*(E_{*fg}^R, E_{*f'g'}^T, E_{gg'}^{RT}(d) \mid \Delta_{*fg}^\psi, \Delta_{*f'g'}^\psi, \Delta_{gg'}^\omega) \\ & \propto e^{-\frac{\Delta_{*fg}}{\gamma_s}} \cdot e^{-\frac{\Delta_{*fg}^\psi}{\gamma_c}} \cdot e^{-\frac{\Delta_{*f'g'}}{\gamma_s}} \cdot e^{-\frac{\Delta_{*f'g'}^\psi}{\gamma_c}} \cdot e^{-\frac{\Delta_{gg'}^\omega}{\gamma_t}} \quad (6) \end{aligned}$$

3.2. Reducing the size of the active support

The second strategy that we propose to reduce the computational cost of the original LC technique consists of reducing the size of the active support deployed. Although optimal results were obtained [12, 13] deploying rather large active supports (i.e., 39×39 pixels), we observed experimentally that smaller active supports (e.g., 15×15 pixels) can be deployed without dramatically sacrificing the effectiveness of the original LC technique. Deploying smaller active supports (e.g., from 39×39 to 15×15) leads to a

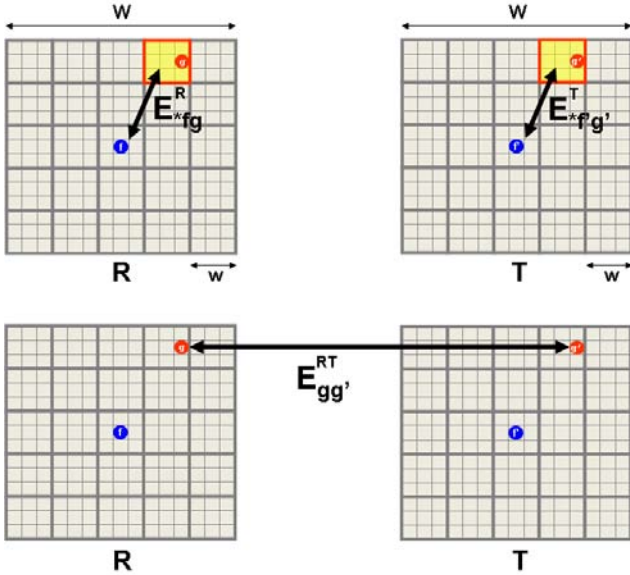


Figure 3. Relaxing the local consistency constraint. (Top) E_{fg}^R , $E_{fg'}^T$: approximated evaluation of the spatial consistency constraints and intra-image color proximity constraints. (Bottom) $E_{gg'}^{RT}(d)$: exact evaluation of the inter-image color proximity constraint. [Best viewed with colors]

significant reduction of the computational burden. On the other hand, this choice also implies that local consistency is enforced only within closer neighboring points, thus reducing the effectiveness of the original LC technique. Although, reducing the size of the active support provides a good trade-off between efficiency and accuracy, our experimental analysis highlights that the performance degradation is less prominent when dealing with more accurate initial disparity hypotheses. To this aim, we report experimental results deploying the initial disparity hypotheses provided by two fast algorithms that lead to significantly different results in terms of accuracy (and execution time). Finally, it is noteworthy that reducing the size of the active supports leads to a significant reduction of the memory footprint. In fact, given a stereo pair of width L , an active support of size W and a disparity range of A pixels, the memory requirements of the LC technique are proportional to $L \times A \times W$. Therefore, the memory footprint is more than halved when deploying active supports of 15×15 in place of 39×39 . This fact would be particularly relevant when dealing with computing platform with extremely constrained resources (e.g., FPGA).

3.3. Exploiting thread-level parallelism

A further proposed improvement in efficiency relies on exploiting coarse-grained thread-level parallelism on multicore architectures. Different from the other proposed strate-

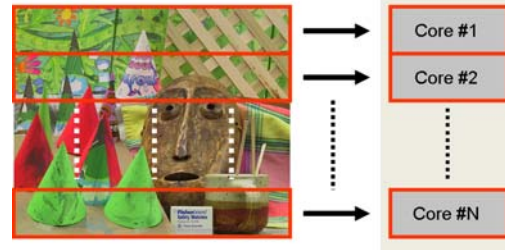


Figure 4. Exploiting coarse-grained thread-level parallelism on multicore architectures.

gies described so far and based on approximations of the original LC technique, thread-level parallelism enables a significant performance improvement without any loss in accuracy. For this purpose, we deployed the OpenMP API [15] that enables multi-platform (e.g., Linux and Windows) shared-memory parallel programming. It is noteworthy that the *local* computational structure of the LC approach allows effective exploitation of coarse-grained parallelism, avoiding the expensive thread synchronization. In fact, although the OpenMP API provides specific instructions for thread synchronization, our experiments suggest that this feature should be avoided to obtain optimal results. To this aim, as shown in Figure 4, we split the stereo pair according to the number of available cores. Then, these portions of the stereo pair are distributed to the available cores and processed independently. This strategy does not require thread synchronization, as the height of each portion is typically significantly larger than the size of the active support; thus threads that process adjacent portions of the stereo pair do not interfere (i.e., they do not write in the same portion of memory). We exploited thread-level parallelism for plausibility computation, cross-checking plausibility (see [12] for details), and left-right consistency check. Finally, we point out that thread-level parallelism could be effectively coupled with data-level parallelism (i.e. executing the same operation on multiple data by means of SIMD instructions), currently available in most off-the-shelf processors and GPUs, so as to obtain even further speed-ups.

4. Experimental results

In this section¹, we validate the effectiveness of our approach within the framework proposed in [13] and summarized in Figure 2. For our experiments, we deploy the initial disparity hypotheses provided by two fast dense stereo algorithms. As usual in stereo, we evaluated our proposal on the Middlebury dataset [17, 16], according to ALL, NOCC, and DISC parameters (see [16] for a detailed description). The initial disparity hypotheses processed by the LC tech-

¹Additional experimental results available at:

www.vision.deis.unibo.it/smatt/RLC_stereo.htm

Algorithm	Rank	Tsukuba			Venus			Teddy			Cones		
		NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC
C-Semiglobal [8]	14th	2.61	3.29	9.89	0.25	0.57	3.24	5.14	11.8	13.0	2.77	8.35	8.20
LC(C-Semiglobal) W=39, w=1 [13]	5th	1.06	1.56	5.71	0.13	0.28	1.84	5.53	11.2	13.9	2.85	8.34	7.53
RLC(C-Semiglobal) W=39, w=3	5th	1.04	1.55	5.62	0.13	0.27	1.88	5.39	10.9	13.5	2.84	8.31	7.56
RLC(C-Semiglobal) W=35, w=5	5th	1.09	1.65	5.90	0.13	0.27	1.88	5.72	11.3	14.1	2.87	8.43	7.65
RLC(C-Semiglobal) W=15, w=1	5th	1.10	1.59	5.98	0.13	0.23	1.76	5.52	11.0	13.9	2.90	8.37	7.74
RLC(C-Semiglobal) W=15, w=3	5th	1.09	1.57	5.88	0.12	0.24	1.70	5.44	11.0	13.6	2.88	8.46	7.73
RLC(C-Semiglobal) W=15, w=5	5th	1.05	1.64	5.66	0.12	0.28	1.69	5.86	11.4	14.5	2.86	8.46	7.66
RLC(C-Semiglobal) W=9, w=1	7th	1.23	1.75	6.61	0.14	0.30	1.99	5.20	10.8	13.3	3.12	8.72	8.29
RLC(C-Semiglobal) W=9, w=3	5th	1.07	1.70	5.81	0.14	0.31	1.86	5.43	11.0	13.7	2.87	8.35	7.70

Table 1. Experimental results, according to the Middlebury evaluation site, deploying the initial disparity hypotheses of the C-Semiglobal [8] algorithm.

nique were those provided by the C-Semiglobal [8] and the RTGPU [23] algorithms available on the Middlebury web site. The former is a quite accurate algorithm based on multiple SOs and a post-processing step aimed at improving the raw results. This method is currently ranked 14th in [16] and has an execution time of 3-4 seconds, according to [8]. It is also noteworthy that the core of this algorithm has been effectively mapped on GPU [5] and FPGA [6]. In contrast, the RTGPU algorithm relies on DP and aggregated costs (computed by means of an approximated *adaptive weight* approach). Due to CPU/GPU mapping, RTGPU delivers, on the Middlebury dataset, disparity fields in real-time. The RTGPU approach is currently ranked 58th in [16]. For our experiments, the benchmarking platform was an off-the-shelf Intel Core 2 Quad CPU at 2.49 GHz with 4 physical cores.

The first row of Table 1 reports the results obtained by the original C-Semiglobal algorithm [8] (referred to as C-Semiglobal) according to the Middlebury metrics (ALL, NOCC, and DISC). The second row reports the result obtained enforcing the local consistency of the disparity field provided by the original C-Semiglobal approach with parameters $W=39$ and $w=1$, according to the framework proposed in [13]. This method is referred to as LC(Semiglobal). The other rows in the table report the results obtained enforcing the *relaxed* local consistency constraint proposed in this paper. This method, for different parameters (see Figure 3) W and w , is referred to as RLC. It can be seen from the table that enforcing the local consistency of the C-Semiglobal disparity field leads, in most cases, to a dramatic improvement (especially for Tsukuba and Venus) in the overall ranking (according to the current Middlebury ranking, from 14th to 5th). Even more interestingly, if we look at the results provided by the RLC approach proposed in this paper, we can see that, in most cases, its effectiveness is comparable to the original approach proposed in [13]. In particular, even deploying small active support (e.g., $W=15$ or $W=9$) always yields equiva-

lent results compared to LC(Semiglobal) [13]. Only with the smaller active support (i.e., $W=9$ and $w=1$) our proposal yields a ranking of 7. Nevertheless, even in this worst case, we notice a significant improvement by enforcing relaxed local consistency, which leads to very accurate results. The table clearly reports that evaluating E_{fg}^R , $E_{f'g'}^T$ events on a block basis and reducing the size of the active supports are both cost-effective strategies. In fact, if we look at Figure 5, where we report the execution times for LC [13] and RLC deploying 1, 2, 3, and 4 cores, we can see that our proposal allows for a notable improvement in terms of reduced execution time. By focusing on the experiments with a single core, it can be seen that results comparable to those of [13] (i.e., 15 seconds for 39(1)) can be obtained with our proposal (i.e., 15(3), 15(5), and 9(1)) in less than 2 seconds. Moreover, the figure also shows that exploiting multithreading, as described in section 3.3, leads to a notable further speed-up, especially with larger active supports. Deploying 4 cores allows us to enforce local consistency by means of [13] in about 4 seconds (vs. the 15 seconds required with a single core) and by means of RLC with parameters $W=15$ and $w=1$, in about 1.3 seconds with comparable results.

Although we have shown so far that reducing the size of the active support leads, in most cases, to equivalent results more efficiently compared to [13], it is noteworthy that this approach also reduces the effectiveness of the LC approach to recover from wrong initial disparity hypotheses. This behavior can be perceived by observing Figure 6, where we report, for Tsukuba, the distribution of plausibility for the disparity hypothesis $d=14$ obtained when processing the disparity field (top-right in the figure) provided by the C-Semiglobal approach with (center of the figure) a large active support (i.e., $W=39$) and (bottom of the figure) a small active support (i.e., $W=9$). Examining the area highlighted in the circles, we can see that C-Semiglobal wrongly set a large portion of the lamp as background. By deploying a large active support (center of the figure), the RLC approach is able to recover most of the wrong disparities, propagating

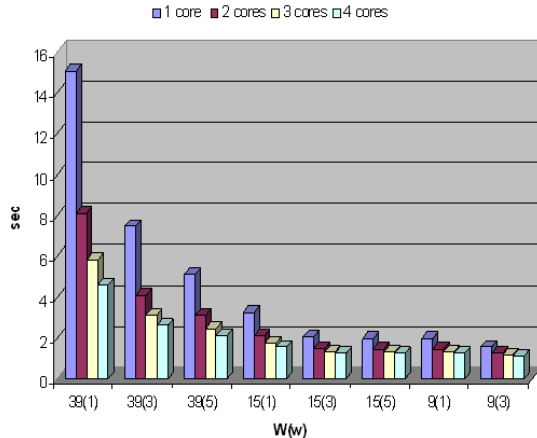


Figure 5. Execution time (for the Teddy stereo pair) on our benchmarking platform (Intel Core2 Quad (Q9300) CPU at 2.49 GHz) for the original LC approach [13] (*i.e.*, $W=39$ and $w=1$, referred to as 39(1)) and the RLC approach proposed in this paper for different configurations of W and w (*i.e.*, 39(3), 35(5), 15(1), 15(3), 15(5), 9(1) and 9(3)). For each approach, we also report the measured execution time deploying 1, 2, 3, and 4 cores.

correct disparity hypotheses. However, with smaller active supports (bottom of the figure), this feature of the RLC approach is significantly reduced.

In Table 2 we report, with the same methodology adopted so far, the experimental results deploying the initial disparity hypotheses provided by the RTGPU [23] algorithm. Obviously, the same results, in terms of the execution time, reported in Figure 5 apply for these experiments. Table 2 shows that the LC-based approach enables dramatic improvement to the disparity fields provided by RTGPU (its ranking increases from 58th to 13th, according to Middlebury). With large active supports (*i.e.*, $W=39$ and $W=35$), evaluating E_{fg}^R , $E_{f'g'}^T$ on a block basis leads to equivalent results much more efficiently. However, in this case, reducing the size of the active support to 15×15 leads to slightly worse results. This behavior can be explained by the fact that the RTGPU disparity fields are significantly less accurate than those provided by C-Semiglobal. Therefore, enforcing local consistency on larger active supports is more effective when the uncertainty of the initial disparity hypotheses is higher.

We conclude this section by observing that our overall proposal enables us to very efficiently obtain disparity fields comparable to top-ranked approaches. On our off-the-shelf computing platform, by deploying the C-Semiglobal algorithm, the overall approach proposed would take about 5-6 seconds (3-4 for C-Semiglobal, less than 2 seconds for RLC, and 160 ms for interpolation). By deploying the disparity hypotheses provided by RTGPU, the overall ap-

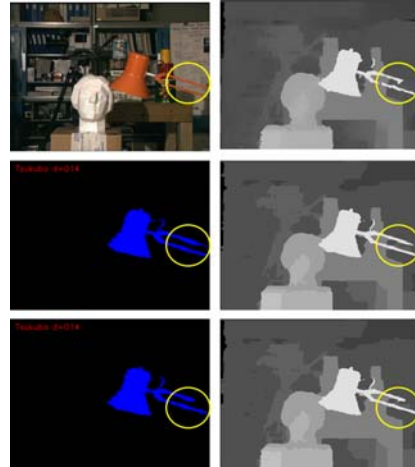


Figure 6. From top to bottom: (Top) Reference image and disparity field provided by C-Semiglobal [8], (Center) Plausibility map for disparity hypothesis $d = 14$ and disparity field obtained by our proposal RLC(C-Semiglobal) with parameters $W=39$ and $w=3$, (Bottom) Plausibility map for disparity hypothesis $d = 14$ and disparity field obtained by our proposal RLC(C-Semiglobal) with parameters $W=9$ and $w=3$.

proach proposed would take about 2 seconds. Finally, it is noteworthy that further speed-up can be obtained by exploiting SIMD capabilities, which are available in most recent off-the-shelf computing platforms. Moreover, we also point out that, due to its limited memory footprint, our overall proposal can be effectively mapped on devices with constrained resources so as to quickly obtain locally consistent disparity fields on low power/low cost devices (*e.g.*, FPGA).

5. Conclusions

In this paper, we have shown that the disparity fields provided by two fast dense stereo algorithms can be efficiently improved by deploying a relaxed local consistency constraint. Our proposal can be effectively mapped on off-the-shelf multicore architectures and, according to standard datasets, allows us to rapidly obtain results comparable to top-ranked approaches. Moreover, our overall proposal has a limited memory footprint, which would render it useful for mapping on devices with constrained resources.

References

- [1] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *Int. J. Comput. Vision*, 35(3):269–293, 1999. 2
- [2] M. Bleyer, M. Gelautz, C. Rother, and C. Rhemann. A stereo approach that handles the matting problem via image warping. In *CVPR09*, pages 501–508, 2009. 2

Algorithm	Rank	Tsukuba			Venus			Teddy			Cones		
		NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC
RTGPU [23]	58th	2.05	4.22	10.06	1.92	2.98	20.3	7.23	14.4	17.6	6.41	13.7	16.5
LC(RTGPU) W=39, w=1 [13]	13th	1.10	1.70	5.94	0.28	0.57	2.89	6.37	12.1	14.3	4.14	10.2	10.7
RLC(RTGPU) W=39, w=3	13th	1.06	1.73	5.60	0.30	0.57	2.80	6.29	12.1	14.1	4.10	10.0	10.6
RLC(RTGPU) W=35, w=5	13th	1.10	1.74	5.86	0.25	0.56	2.43	6.63	12.5	15.0	4.33	10.2	10.9
RLC(RTGPU) W=15, w=1	19th	1.01	1.69	5.47	0.49	0.91	5.71	6.46	12.4	14.7	4.67	10.5	11.8
RLC(RTGPU) W=15, w=3	19th	1.04	1.74	5.59	0.43	0.82	4.84	6.45	12.4	14.6	4.70	10.6	11.8
RLC(RTGPU) W=15, w=5	21th	1.07	1.87	5.73	0.40	0.86	4.62	6.72	12.6	15.3	4.56	10.5	11.3
RLC(RTGPU) W=9, w=1	28th	1.15	2.09	6.12	0.73	1.27	8.98	6.38	12.5	14.5	4.92	10.8	12.5

Table 2. Experimental results, according to the Middlebury evaluation site, deploying the initial disparity hypotheses of the RTGPU [23] algorithm.

- [3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24:603–619, 2002. 2, 3
- [4] Y. Deng and X. Lin. A fast line segment based dense stereo algorithm using tree dynamic programming. In *Proc. European Conf. on Computer Vision (ECCV 2006)*, volume 3, pages 201–212, 2006. 1, 3
- [5] I. Ernst and H. Hirschmüller. Mutual information based semi-global stereo matching on the gpu. In *ISVC '08: Proceedings of the 4th International Symposium on Advances in Visual Computing*, pages 228–239, 2008. 2, 6
- [6] S. Gehrig, F. Eberli, and T. Meyer. A real-time low-power stereo vision engine using semi-global matching. In *CVS09*, pages 134–143, 2009. 1, 2, 6
- [7] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proc. Conf. on Computer Vision and Pattern recognition (CVPR 2005)*, volume 2, pages 807–814, 2005. 2
- [8] H. Hirschmüller. Stereo processing by semi-global matching and mutual information. *IEEE Trans. on PAMI*, 2(30):328–341, 2008. 1, 2, 3, 6, 7
- [9] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *ICIP*, 2009. 2
- [10] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR '06*, pages 15–18, 2006. 2
- [11] C. Lei, J. Selzer, and Y. Yang. Region-tree based stereo using dynamic programming optimization. In *CVPR06*, pages II: 2378–2385, 2006. 3
- [12] S. Mattoccia. A locally global approach to stereo correspondence. In *3DIM2009*, pages 1763–1770, Kyoto, Japan, 2009. 1, 2, 3, 4, 5
- [13] S. Mattoccia. Improving the accuracy of fast dense stereo correspondence algorithms by enforcing local consistency of disparity fields. In *3DPVT2010*, Paris, France, 2010. 1, 2, 3, 4, 5, 6, 7, 8
- [14] S. Mattoccia, S. Giardino, and A. Gambini. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering. In *ACCV2009*, Xian, China, 2009. 2
- [15] OpenMP. The openmp api specification for parallel programming. <http://openmp.org/wp/>. 5
- [16] D. Scharstein and R. Szeliski. Middlebury stereo vision. <http://vision.middlebury.edu/stereo/>. 1, 2, 3, 5, 6
- [17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Jour. Computer Vision*, 47(1/2/3):7–42, 2002. 1, 2, 5
- [18] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. PAMI*, 30(6):1068–1080, 2008. 2
- [19] Y. Taguchi, B. Wilburn, and C. Zitnick. Stereo reconstruction with mixed pixels using adaptive over-segmentation. In *CVPR08*, 2008. 2
- [20] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *CVPR08*, pages 1–8, 2008. 1, 2
- [21] O. Veksler. Stereo correspondence by dynamic programming on a tree. In *CVPR '05*, pages 384–390, 2005. 1, 2, 3
- [22] W. W. Chen, M. Zhang, and Z. Xiong. Segmentation-based stereo matching with occlusion handling via region border constrains. *CVIU (submitted)*, page na, na. 2
- [23] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In *3DPVT '06*, pages 798–805, 2006. 1, 2, 3, 6, 7, 8
- [24] Z.-F. Wang and Z.-G. Zheng. A region based stereo matching algorithm using cooperative optimization. In *CVPR*, 2008. 2
- [25] L. Xu and J. Jia. Stereo matching: An outlier confidence approach. In *ECCV 2008*, pages 775–787, 2008. 2
- [26] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Trans. PAMI*, 31(3):492–504, 2009. 2
- [27] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. In *Proc. of CVPR2007*, pages 1–8, 2007. 2
- [28] K. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. PAMI*, 28(4):650–656, 2006. 2