Noname manuscript No.
(will be inserted by the editor)

PREPRINT: On the performance of local methods for stereovision

A. Buades · G. Facciolo

the date of receipt and acceptance should be inserted later

Abstract We consider two problems associated with non-dense block-matching stereo algorithms. The first is to compute correct disparities at places where the pronto-parallel hypothesis is invalid: places like discontinuities and slanted surfaces. The second is to detect when a match is ambiguous or invalid. We advocate, in this work, for the use of oriented windows in order to deal with slanted surfaces and discontinuities. Unlike adaptive support windows the oriented windows permit to correctly estimate disparities on slanted surfaces. To improve the mismatch detection we propose two new parameter-less techniques, which are based on old ideas. One, based on the notion of *distinctiveness*, is used to detect ambiguous matches. The other, based on the *min-filter*, is used to detect occluded and dis-occluded pixels. The incorporation of these simple ingredients in a coarse-to-fine algorithm, allows to obtain fairly dense results with less mismatches than other algorithms with similar characteristics. Our work is motivated by photogrammetric applications from high-resolution aerospatial imagery. The automatic generation of urban digital elevation models from this type of images justifies our interest on obtaining sparse but reliable disparity maps.

Keywords stereovision · mismatch detection · multiple correlation windows · slanted surfaces

A. Buades
E-mail: toni.buades@uib.es

G. Facciolo
E-mail: gabriele.facciolo@cmla.ens-cachan.fr

1 Introduction

Stereovision consists in finding the depth of a scene from two or more images taken from slightly different viewpoints. This is one of the central problems in computer vision, and it has been an active research subject for the last thirty years. The principle of stereovision is that the apparent motion of a point, resulting from a camera movement, is related to its distance from the camera. The problem is then to find corresponding points in the images in order to compute their apparent motion (or parallax movement).

The correspondence problem or disparity computation reduces in the case of two stereo-pair images u and v to the search of a disparity function ϵ such that

$$u(\mathbf{x}) = v(\mathbf{x} + \epsilon(\mathbf{x})).$$

For the pinhole camera model it can be shown [12] that the parallax movement of a point occurs only along a line (called *epipolar*), and that the images can always be rectified so that the disparity of every points is horizontal.

Since the above equation is ambiguous many stereovision algorithms do not look for the function ϵ by matching the grey level of individual pixels but instead match the grey level of an entire window around each pixel. These are designated *local methods* [24] because they rely only on information present in the neighborhood of a pixel to determine its disparity. Conversely, the *global methods* disambiguate the matching by imposing some global regularity constraints on the output disparity function.

The local methods compute the disparity by correlating a small window (or patch) along the epipolar lines, being the *Sum of Squared Differences* (SSD) the most common matching cost used for this purpose. The

lack of texture or information in the image is the main challenge of local methods. That is, if an image patch lacks texture or its signal-to-noise ratio is too small, then the computed disparity will likely be incorrect. The *global methods* cope with ambiguities by restricting the disparities to some class of smooth functions, which usually permit to derive reasonable estimations even on textureless areas. Most global methods could even extend these estimations to half-occluded regions of the images (regions which are visible only in one of the images). However, the result of a global method makes no explicit distinction between bad and good matches, confusing where the disparity is factual or not. Moreover, the smoothness of the solution is always controlled by a parameter, whose optimal value usually depends on the scene being processed.

The choice between local and global methods is certainly application dependent. In some cases it may be critical to obtain dense disparity maps regardless of the possible extrapolation errors that a global method may introduce. However, since the main application we are considering is the generation of urban digital elevation models from satellite images, in order to keep only the factual correspondences we opt for a local method. Thus, as in [7,13], our algorithm will favor reliability over completeness by allowing for non matched regions. For that we propose a set of tests that permit to identify the reliable matches.

We distinguish two stages of the stereo estimation process: the computation of a candidate disparity map by a winner-take-all strategy [24]; and the classification (or rejection) of the mismatches which is done independently of the mechanism that generated the matches. Thus, a good algorithm should generate good candidate disparities and should be effective at detecting the mismatches. This modularization simplifies the exposition and the analysis of the individual choices. For computing the disparities we will consider classic block-matching costs with uniformly weighted windows. Concretely the Sum of Squared Differences (SSD) and the zero-mean SSD (ZSSD). The rejection tests are aimed at detecting mismatches due to factors like: lack of texture, repetitive patterns and occlusions.

Detect mismatches. Let us first comment on the main causes of matching errors and how to handle them.

Match ambiguity. As mentioned before the ambiguity due to lack of texture in the image is one of the main sources of matching errors for the local methods. Similar issues are observed in presence of the aperture problem, or when the patch belongs to an object with a repetitive pattern. The common denominator for all these cases is the ambiguity of the matches. Many methods [7,13,18] detect ambiguity by comparing the costs of the best and second best match candidates, but this usually requires to fix a parameter. Manduchi and Tomasi [19] observed that self-similar regions of the reference image yield ambiguous matches because the are not *distinctive*. Based on this notion of distinctiveness we will propose a new parameter-less criterion to detect these ambiguous matches.

Almost all local stereo matching algorithms use the left-right consistency check [8,9] for detecting mismatches. Although this test excels in the detection of occlusions (see below) it can also detect some ambiguous matches.

Inadequacy of the geometric model. Three-dimensional geometric discontinuities of the scene produce occlusions and dis-occlusions, which manifest as spurious matches or foreground fattened results. Figure 1 illustrates these concepts. A similar phenomenon is observed on slanted surfaces. The block-matching performance is usually poor on slanted surfaces (that are not occluded), and may even produce incorrect matches. The overall reason for these failures is the inadequacy of the fronto-parallel hypothesis for block-matching methods. That is, the assumption that any image patch can be found undistorted in the second image. The patches overlapping discontinuities are only partially visible in the second image. And the patches on slanted surfaces appear distorted in the second image (by an affinity in the case of rectified images).

The left-right consistency check [8,9] detects most occlusions, however the foreground fattening associated to the dis-occlusions is usually not detected by this technique. In this work we propose a new filter based on the *min-filter* [2,6] that permits to reject most of the foreground fattened regions.

Improve matching. Errors near depth discontinuities result from the incorrect assumption that any image patch appears translated in the second image. Let us see how these errors can be attenuated or prevented.

The usual way to cope with depth discontinuities is to use adaptive windows (also known as adaptive support weights) that avoid image discontinuities as was first proposed by Kanade and Okutomi [15]. Similar works by Lotti and Giraudon [17] and more recently by Wang et al. [29] pre-compute the image edges and recursively grow a comparison window while avoiding the edges. Patricio et al. [21] and Yoon and Kweon [31]

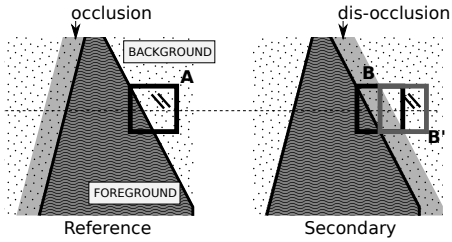


Fig. 1 Depth discontinuities and foreground fattening. The center of the window (A) is on the background but the window contains the depth boundary, thus some points in the window match at the foreground disparity (B), while others match at the background disparity (B'). The best match will be for the region containing the stronger horizontal texture, which is usually the object boundary itself. Since the object boundary is at the foreground disparity, a strong preference for the foreground disparity is created. Leading to the well-known "foreground fattening" effect.

select an adaptive window containing only pixels with a grey level similar to the reference one, like in neighborhood and bilateral filters [27,30]. All of these methods identify depth discontinuities with image discontinuities, which is not always the case.

Other approaches do not try to avoid explicitly the discontinuities of the image. Instead, they choose an adaptive window with a minimum matching cost criterion. The subjacent idea is that windows that do not contain depth discontinuities will be matched with a smaller matching cost. Fusiello et al. [10] choose among all the windows containing the reference pixel the one that yield the minimal matching cost. Veksler [28] applied the same strategy but considering square windows of different sizes. A more elaborated version by Hirschmüller et al. [13] adapts the shape of the window by dividing the matching window into small subwindows and by selecting those for which the minimum matching cost is attained.

The methods based on adaptive windows deal with the discontinuity issues by adapting to the shapes in the image. However they are still bound to the fronto-parallel hypothesis in the sense that they don't contemplate the case of slanted surfaces. To cope with the slanted surfaces some methods [5,11] propose to explore, for each image point, all possible surface orientations. For rectified images the transformations between planar patches are reduced to three dimensions: translation, horizontal tilt and shear. In [5] this over-parametrized space of transformations is efficiently explored using a random exploration algorithm.

In this work we use multi-window block-matching algorithm similar to [13] which chooses for each pixel among a group of oriented windows. Using elongated and oriented windows improves the matching on slanted surfaces and permits to match pixels closer the disparity discontinuities. On a slanted surface the chosen window will adapt to the direction of least variation of the disparity, thus matching with a low cost. Whereas near a disparity discontinuity the window oriented with the edge will match with lower cost, thus permitting to match and validate closer to the discontinuity. Moreover, as shown in the asymptotic analysis performed by Blanchet et al. [4], the disparity estimated by SSD minimization is in reality to the mean disparity within the matching window. The oriented windows are consistent with this analysis as they permit to minimizes the variation of disparity within the window, thus leading to more accurate results.

The proposed block-matching algorithm is melded with a principled coarse-to-fine strategy that allows to reduce the occurrence of ambiguous matches and also reduce the computation time. Moreover the multi-scale strategy interacts positively with the multi-window matching yielding denser results.

Contributions and plan of the paper. In this work we propose a principled block-matching algorithm for stereo, using a coarse-to-fine strategy. A multi-window algorithm with oriented windows is used to compute the disparities. This new approach permits to deal with non fronto-parallel surfaces and allow to match near the object's edges, leading to higher matching densities. We propose two new match classification techniques that are virtually parameter-less, that extend previous ideas on match validation [2,3]. Unlike other works [7,13] the proposed algorithm has no tredeoffs or parameters to be set beside the size of the patch. Compared to other non-dense block-matching methods the proposed algorithm yields state-of-the-art performances in terms of density and mismatch rates. To test the algorithm an on-line demo is also made available[1].

We summarize the plan of the paper as follows. Section 2 introduces the block-matching method that will be used for disparity estimation. Section 3 describes the criteria used to reject incorrect matches. Each criterion is designed to cope with one of the block-maching's drawbacks, namely lack of texture or information, strobe effect or fattening effect. In Section 4 we introduce the multi-window block-matching algorithm which chooses for each pixel among a group of oriented windows. Section 5 describes a multi-scale strategy that is performed by adapting at each pixel's disparity range depending on the estimation at a lower scale. It is shown that this strategy improves the results of the block-matching. Finally, Section 6 compares the proposed algorithm with

[1] On-line demo of the proposed method available at: `http://dev.ipol.im/~facciolo/ipol_demo/msmw/`

two state-of-the-art local methods for non-dense block-matching and a global method based on the graph cuts minimization. It is shown that the proposed method yields the lowest mismatch rates among all the considered algorithms while producing a fairly dense disparity map. It is illustrated by examples that global method fail in estimating correctly the disparity near depth discontinuities and detecting occlusions. A global fattening effect appears due to the regularity constraints of global methods. Thus, only local methods allow for a control of matches and thus a validation step.

2 Block-Matching by SSD

We denote by $\mathbf{x} = (x_1, x_2)$ the position of a point in a continuous image domain $\Omega \subset \mathbb{R}^2$, and by $u(\mathbf{x}) = u(x_1, x_2)$ and $v(\mathbf{x})$ the images of a rectified stereo pair. For non-integer positions $\mathbf{x} \in \Omega$, $u(\mathbf{x})$ is computed interpolating the discrete samples of u. The disparity is estimated by

$$\mathbf{d}(\mathbf{x}) := \underset{\mathbf{d}' \in p\mathbb{Z}}{\arg\min}\, c(\mathbf{x}, \mathbf{x} + \mathbf{d}'),$$

where $c(\cdot, \cdot)$ is a cost function computed by comparing patches of u and v centered respectively at $\mathbf{x}$ and $\mathbf{x}+\mathbf{d}'$. The factor $p \in \mathbb{Q}$ is the disparity step and it is set to $p = 1$ in most methods. We will consider here the following two cost functions:

- The sum of squared differences (SSD), which writes as the Euclidean distance between patches

$$\mathrm{SSD}(\mathbf{x}, \mathbf{y}) := \frac{1}{|B_r|} \sum_{\mathbf{t} \in B_r} |u(\mathbf{x} + \mathbf{t}) - v(\mathbf{y} + \mathbf{t})|^2,$$

  where B_r denotes the domain of a patch (or window) centered at $\mathbf{0}$ and parameter r. For squared windows r is the length of its side. For color images the cost is computed as the average among all the channels.
- The zero-mean SSD (ZSSD) [14] removes the average intensity of each patch rendering the comparison independent of the mean intensity. This cost is defined as

$$\mathrm{ZSSD}(\mathbf{x}, \mathbf{y}) := \frac{1}{|B_r|} \sum_{\mathbf{t} \in B_r} \Big| u(\mathbf{x} + \mathbf{t}) - v(\mathbf{y} + \mathbf{t}) - \overline{u|_{\mathbf{x}+B_r}} + \overline{v|_{\mathbf{y}+B_r}} \Big|^2,$$

  where $\overline{v|_{\mathbf{y}+B_r}}$ denotes the average of image v over the window centered at $\mathbf{y}$.

Figure 2 compares results using these costs for two stereo pairs. For the first pair, homologue regions have exactly the same color or intensity. This is not the case for the second pair where there is a small change of intensity between the images particularly in the dark areas. For the first pair, both costs lead to a similar result while for the second one the mean normalized cost gives much better results. For this reason, we will use the ZSSD cost.

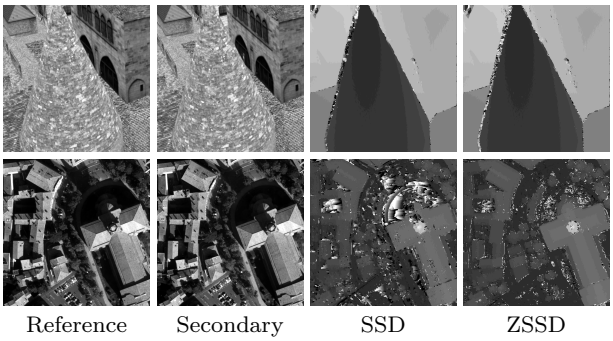


Fig. 2 Comparison SSD and ZSSD with square patch of size 5×5 on two stereo pairs. From left to right: stereo pair, disparity estimated by SSD and disparity estimated by ZSSD. Both methods give similar results when intensity does not change from one image to the other. Although it is not easily noticeable, small intensity differences in the second pair yield to SSD mismatches while are correctly matched by ZSSD.

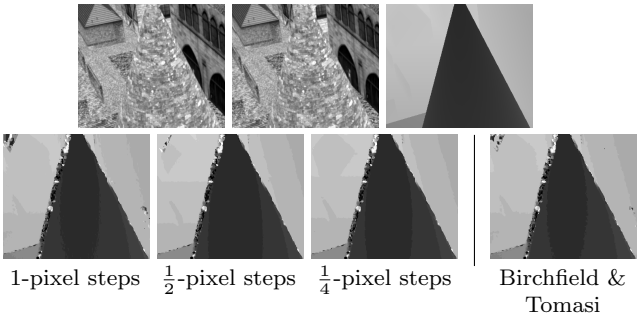


Fig. 3 Top: synthetic stereo image pair and ground truth disparity. Bottom, from left to right: disparity computed using SSD by exhaustive search with 1-, $\frac{1}{2}$- and $\frac{1}{4}$-pixel steps. Note that increasing the precision eliminates false matches from the upper-left corner and right portion of the image. The last image is computed using the sampling insensitive cost proposed by Birchfield and Tomasi [3] with integer disparity steps, note that the result is comparable to SSD at 1 pixel. All disparity images are quantified at 1 pixel for visual comparison.

Subpixel disparity computation. Due to the image sampling, if the true disparity is not integer, the reference patch may not exactly be present in the second image. That is, the exact patch will not exist in the second image because of the different sampling. As a result some matches will be incorrect if estimated at integer steps, even if the patches are well textured and not repeated. To cope with this issue, in this work we compute the disparities with a fixed subpixel step. As seen in Fig-

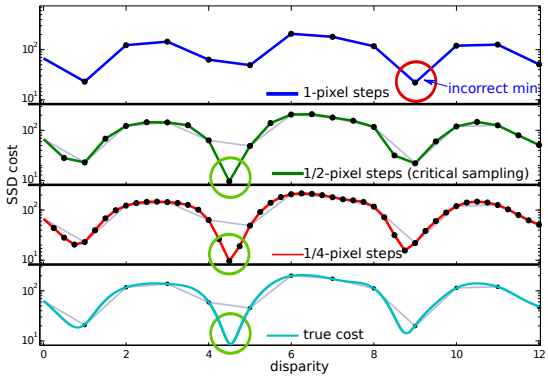


Fig. 4 Effects of subpixel sampling of the cost function. The graphs show the SSD cost profiles obtained by matching a region of the image shown in Figure 3. The curves show the costs $c(\mathbf{d})$ sampled with disparities steps of 1-, $\frac{1}{2}$-, $\frac{1}{4}$-pixel and the continuous cost. Clearly sampling the cost function at integer disparity steps can lead to incorrect results as the global minimum can hide between two samples. Sampling at $\frac{1}{2}$-pixel steps (which is the critical sampling of the SSD cost) the failures are less common but can still occur.

ure 3 the subpixel estimation improves the matching performance by avoiding correspondence mismatches.

To minimize the overhead involved in the subpixel estimation Birchfield and Tomasi [3] proposed a matching cost (based on the SSD) that is insensitive to the sampling of the patches. This cost is estimated at integer positions and is proven to be stable with respect to subpixel translations of the patches. It is particularly useful when used within global methods, where it can prevent the misclassification of some pixels as occlusions. However, as seen in Figures 3, for a winner-take-all method this cost yields a limited performance gain with respect to the standard SSD cost.

Szeliski and Scharstein [26] and Sabater et al. [23] have shown that in order to properly compute and interpolate the SSD cost $\mathbf{d} \mapsto c(\mathbf{d})$ (with $c(\mathbf{d}) := c(\mathbf{x}, \mathbf{x}+\mathbf{d})$ for a point $\mathbf{x}$), the input images and the cost function must be oversampled by at least a factor two. In a nutshell, this is because the squared difference (SSD) of two band-limited images has twice the bandwidth of the input images. In that case, perfect band-limited interpolation (sinc filter) could be used to interpolate the cost function from the samples taken at half-pixel disparity steps, allowing to accurately compute the disparity with arbitrarily subpixel precision.

The proposed oversampling of the cost function $\mathbf{d} \mapsto c(\mathbf{d})$ is important because it guarantees that the disparity can be exactly interpolated from the samples. The oversampling (zooming) of the images, on the other hand, is rather technical and is needed to guarantee that the discrete cost coincides with its continuous generalization [23]. Despite of that we will neglect the image oversampling as we observed that it rarely affects the matching results.

Figure 4 illustrates a situation in which sampling the cost function at 1 pixel steps is insufficient to obtain the correct match. Sampling the cost function at integer disparity steps can produce incorrect results as the global minimum of the underlying cost may remain hidden between samples. Failures are less common when sampling at 1/2 pixel intervals but can still occur, while we never observed such errors when sampling at 1/4 pixel intervals.

Although more expensive, sampling the cost function at rates higher than the critical sampling (steps smaller than 1/2 pixel intervals) permits to sample closer to the minimums and at the same time make possible to interpolate the costs using compact interpolation filters [26] (recall that at the critical sampling rate only sinc should be used). Looking at Figure 3 we can confirm that at least for finely textured objects sampling the cost function at quarter-pixel steps significantly improves the matching performance. Thus, in this work we estimate the disparities with quarter-pixel precision.

3 Rejection criteria

The rejection tests are applied to the disparity maps independently of the mechanism that generated them. These tests decide based on the matching costs and the content of the images. The modularization simplifies the evaluation of the stereo chain. As a good stereo algorithm is made of a good mechanism for generating candidate disparity maps and a good set of rejection tests. Several rejection criteria are proposed and will be described in what follows.

Left-Right consistency. The Left-Right consistency test [8,9] predicts stereo mismatch where the left-based disparity values are not the inverse mapping of the right-based disparity values. The test rejects a match if $|\mathbf{d}_R(\mathbf{x}+\mathbf{d}_L(\mathbf{x})) + \mathbf{d}_L(\mathbf{x})| > \tau$, where the threshold τ is usually set to 1 and $\mathbf{x}+\mathbf{d}_L(\mathbf{x})$ denotes the position in the right image of the homologous point of $\mathbf{x}$ (in the left image).

This test consistently detects occluded pixels and some, but not all, dis-occluded pixels. In some cases it also detects some repetitive and textureless areas. The foreground fattening associated to dis-occlusions is not detected by this test.

Self-similarity. Intuitively a match is uncertain or ambiguous if the same local structure appears repeatedly. A common criterion used to detect these situations consists in comparing the costs of the best and second best

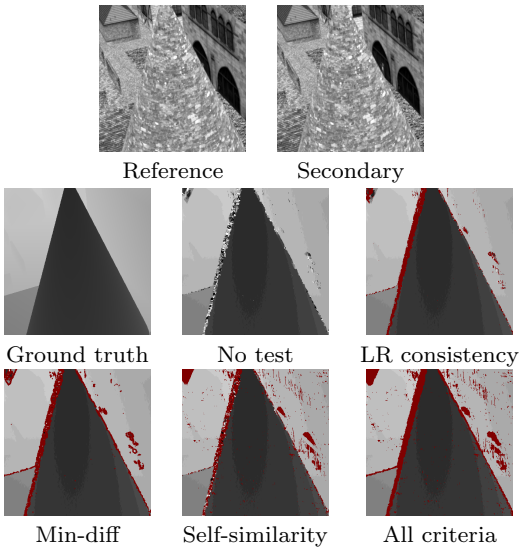


Fig. 5 Application of reject criteria with a 5×5 patch. The disparity is displayed with rejected pixels by each criterion marked in red. From top to bottom and left to right: pair images, ground truth, disparity with no criteria applied, LR consistency, min-diff, self-similarity and all criteria at once.

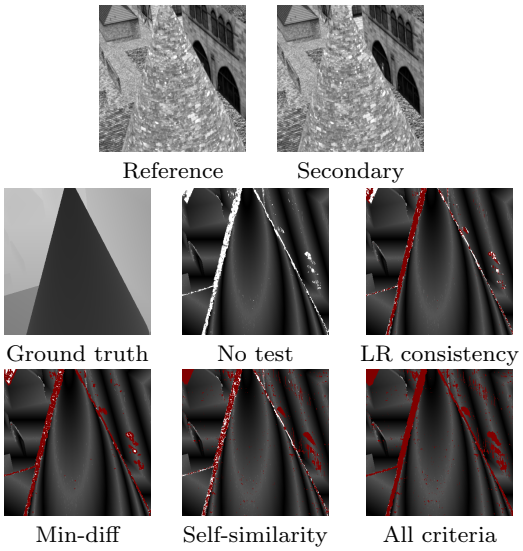


Fig. 6 Application of reject criteria with a 5×5 patch. The committed error is displayed with rejected pixels by each criterion marked in red. Error image displayed with range $[0, 1.5]$. From top to bottom and left to right: pair images, ground truth, disparity with no criteria applied, LR consistency, min-diff, self-similarity and all criteria at once.

match candidates in the secondary image [7,13], let c_1 and c_2 be respectively these costs. This test is also used in correspondence algorithms such as SIFT [18] which rejects matches if the ratio c1/c2 exceeds a fixed 80% threshold.

Manduchi and Tomasi [19] observed that the content of the reference image can be used to predict how ambiguous the match would be. For that they define

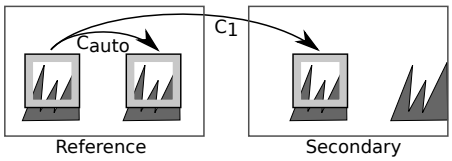


Fig. 7 Self-similarity. This criterion compares the cost c_1 of the best match candidate in the secondary image and the cost c_{auto} of the best match in the reference image itself. If $c_{auto} < c_1$ then the match is considered ambiguous.

the *distinctiveness* of an image position as the perceptual distance to the most similar other position in the same image within a search window. Distinctive points are not necessarily rich in texture, but its features are unique as they look like nothing else in the picture (at least along the epipolar line). Particularly, in [19] the authors studied the case of the auto-SSD function (Sum of Squared Differences computed in the same image), and proposed to reject all pixels with auto-SSD above the average cost value for the entire image.

Here we propose a new criterion inspired by [19], one that compares the costs of the best candidate in the secondary image c_1 and the cost of the best match in the reference image itself c_{auto} (see Figure 7). The idea is that when the auto-similarity cost of a patch (c_{auto}) is smaller than the cost of its best match in the second image (c_1), then patch is likely to be part of a portion of a repetitive pattern and thus non-distinctive. The differences due to the samplings in both images might interfere with this comparison. That is, these costs will be comparable only if all patches are sampled at the same sub-pixel position, which is not usually the case. For this reason, we compensate for this fact by introducing a sampling term. The final criterion rejects a match if

$$c_1 > c_{auto} - \max(c_{auto}(\mathbf{x}, \mathbf{x} + p/2), c_{auto}(\mathbf{x}, \mathbf{x} - p/2))$$

where $c_{auto}(\mathbf{x}, \mathbf{x}+p/2)$ and $c_{auto}(\mathbf{x}, \mathbf{x}-p/2)$ denote the cost of matching a window of the reference image with itself but shifted by $p/2$ pixels, in our case $p = 1/4$ is the disparity step. Note that c_{auto} acts as an adaptive threshold for the match, thus if the patch belongs to a repetitive structure, then the test is more restrictive. Unlike the tests based on the second best match, this test needs no parameter tuning.

Textureless regions are also discarded by this method since they are detected as ambiguous.

Min-diff. The method of shiftable correlation windows [2,6] computes the disparity at each position by considering the costs of all the possible windows that contain it (not only the one centered at it). This well known technique significantly reduce the *foreground fattening*

near the occlusion boundaries of the scene. When the window is uniformly weighted, the shiftable correlation can be computed by the so-called *min-filtering*, which is a post-process of the costs computed with centered windows [24]. *Min-filter* replaces the disparity of the current pixel by the disparity of the pixel that realizes the minimum cost within the window. However the *min-filter* post-process creates shocks on regions with smooth disparities and sometimes it may also propagate incorrect disparities near texture-less areas. Here we will take advantage of *min-filter* for detecting foreground fattened pixels.

The *min-diff* criterium is based on the observation that the foreground fattened disparities at occluded and dis-occluded pixels are significantly modified by *min-filter*. Thus, *min-diff* compares the output of *min-filter* ($\mathbf{d}_{MF}$) with the original disparity ($\mathbf{d}$), and if the difference exceeds a threshold the pixel is rejected. The test is summarized as:

$$|\mathbf{d}_{MF}(\mathbf{x}) - \mathbf{d}(\mathbf{x})| > s.$$

We fix the threshold to $s = 1$ which allows for small changes due to the shocks introduced by *min-filter*. Indeed the shocks on smooth areas are usually small in magnitude, one pixel at most. When *min-filter* modifies significantly the disparity at a position this suggests that a better disparity hypothesis can be found by using a shifted window. This is usually associated to mismatches at the dis-occlusions and occlusions, in which case this criterion marks the position as invalid.

The *min-diff* test has a tendency to under-estimate the fattened regions. This happens particularly at the occlusions boundaries where there is no real alternative to the foreground fattening because the area is not visible in the second image. We found that dilating the rejection mask by 1 pixel improves the detection of the occlusions at the expense of few false rejections.

Removal of isolated matches. This last criterion rejects all accepted points that are "isolated", meaning that they are surrounded by many rejected points. This criterion rejects a point if more than 75% of its matching window has been rejected already. The rationale of this test is that an isolated valid point is more likely to be a mismatch than a proper match. In the eventuality of being a proper match, an isolated point would still be a too small feature for the current window size.

Comparison of rejection criteria. Figure 5 show the disparity computed with ZSSD and the rejection masks for each of the proposed criteria. Figure 6 show the errors committed by the matching algorithm and the points rejected by each criteria.

As expected the Left-Right consistency test rejects points on the occlusion and dis-occlusion parts of the image because in those regions the same patch doesn't exist in both images. However, some of the occlusion and dis-occlusion points may produce consistent matches because of the fattening effect. In those cases the correspondence match is dominated by the occluding edge, which is visible in both images. In Figure 6 can also be seen that this consistency check may detect some incorrect matches due to the lack of texture or auto similarity of patches.

The min-diff criterion rejects points mainly in dis-occlusions parts of the scene as shown in Figure 5. Usually, in a dis-occlusion part there exists a shifted window visible in both images that matches with a smaller cost. Some incorrect matches are also detected in the occlusion parts, this is because occluded patches usually match with a random disparity but a high cost, thus they are rejected if any neighboring pixel matches with smaller cost and significantly different disparity. The biggest limitation of the *min-diff* test is that it is biased towards the detection of fronto-parallel surfaces. Indeed the threshold $s = 1$ limits the maximum surface inclination it can accept, which leads to many unnecessary rejections. Nonetheless, for the current work we keep it with its limitations and leave its refinement for future work.

The self-similarity criterion correctly rejects points in auto similar or ambiguous zones (see the rooftop in the top left corner of the image in Figure 5). It also might reject pixels in occluded parts since it might happen that the auto-similarity cost is smaller than the matching cost just because the patch does not exist in the secondary image. Moreover, as mentioned before, the self-similarity test also detects almost completely the textureless regions, leaving only some isolated points which are later removed by the isolated pixel criterion.

It is important to note that the rejection criteria must be applied in a specific order, so that once a point has been rejected by a test it is no longer considered in the subsequent tests. Concretely the removal of isolated matches should be applied last, and min-diff should be applied after the self-similarity and left-right consistency check.

4 Multiple window orientations

A drawback of the block-matching methods is that the fronto-parallel assumption is often invalid. That is, if the disparity function is not constant within a window, then the corresponding region in the second image appears distorted, thus the two image patches could

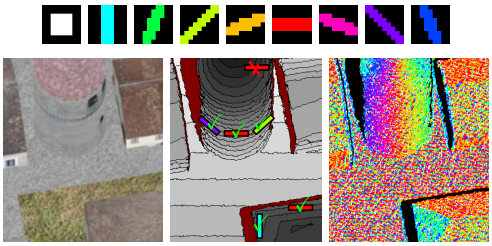


Fig. 8 Correlation windows with different orientations better adapt to different slanted planes, because with the correct window orientation the depth of the underlying surface is almost constant. Top row: oriented windows used in the experiments (with color code), corresponding to a 5×5 window. Bottom row from left to right. The reference image from a stereo pair. The computed disparity map with some isolines highlighted. Overimposed are shown some windows whose orientation best fit the surfaces. An illustration of the actual window orientation as determined by the algorithm.

not be compared directly. The SSD minimization is robust against slight deformations, but fails in presence of stronger ones. An extreme case of this failure occur when the window contains a discontinuity (two different depths). In this case it is impossible to find a unique correspondence leading to the occlusions and dis-occlusions artifacts.

The consequences of the fronto-parallel assumption's failure are two fold. First, the algorithm is unable to correctly match points near disparity discontinuities, thus the reject criteria invalidates these matches. This is illustrated by previous experiences (Figures 5 and 6) where a band of invalid points (with the width of matching window) covers the disparity discontinuities. Second, mismatches appear on highly slanted surfaces because of the strong distortions suffered by the image within the window.

To cope with both problems we propose to perform the block-matching for each pixel using elongated windows with different orientations and then choosing the window yielding the minimum matching cost. This approach differs from the shiftable window proposed in [2,6,24]. We propose to use only windows centered in the reference pixel but with different shapes. The shape of the window giving the minimum matching cost intuitively will be the one for which the disparity function varies the least. That is, the window being as fronto-parallel as possible (see Figure 8).

Using elongated and oriented windows permits to improve the matching on slanted surfaces and to match pixels closer the disparity discontinuities. Near a disparity discontinuity the chosen window will be the one wider in the direction of the discontinuity and narrower in the direction orthogonal to the discontinuity, thus permitting to match and validate closer to the discontinuity. Whereas on a slanted surface the chosen window will adapt to the direction of least variation of the disparity, thus matching with a lower cost.

Algorithm 1: Multi-window block-matching algorithm. $dMin$ and $dMax$ are images indicating the search ranges for all the pixels. $disp$ contain both disparity maps LR and RL.

$MultiWindowMatching(u_0,\ u_1,\ dMin,\ dMax)$;

Input: image pair u_0 and u_1.
Input: min & max disp. range images $dMin$ and $dMax$, minimum & maximum disparity for each pixel.

Output: Fixed precision disparity $disp$.

for each window $\mathbf{w}$ **do**
  `// Compute left and right disparity maps`
  $disp_{\mathbf{w}} = ZSSDMatching(u_0,\ u_1, dMin, dMax, \mathbf{w})$

  `// Apply the rejection criteria to` $disp_{\mathbf{w}}$
  Update $disp_{\mathbf{w}}$ applying the LR criterion
  Update $disp_{\mathbf{w}}$ applying the MIN-DIFF criterion
  Update $disp_{\mathbf{w}}$ applying the Self-similarity criterion
  Update $disp_{\mathbf{w}}$ removing the ISOLATED MATCHES
end for

$disp$ = Combine all the $disp_{\mathbf{w}}$

`// Ensure consistency of` $disp$ `after combining`
Update $disp$ applying the LR criterion
Update $disp$ removing the ISOLATED MATCHES

The rejection criteria of Section 3 are straightforwardly adapted to any window shape. Moreover, we observed that the matches are much easily validated for the correct window shape, thus improving the overall performance of the rejection stage. For each window shape, the matching and rejection stages are applied independently. Then for each pixel the choice for the optimal window shape is made considering only the window orientations validated at that pixel. Algorithm 1 details the complete process. Although in the algorithm the matching process is repeated for each window orientation, all the orientations can be efficiently computed by combining smaller support windows as proposed by Hirschmüller et al. [13].

The performance of ZSSD block-matching using 5×5 windows and the proposed multi-window algorithm using 5 and 9 window orientations is compared in Figure 9. We use the vertical, horizontal and diagonal windows shown in Figure 8, all windows have roughly the same area. Figure 9 illustrates how pixels are correctly matched nearer the occlusions and dis-occlusions and more pixels are correctly matched and validated in slanted parts of the scene. The window orientations, illustrated with the color code of Figure 8, coincide as expected with the direction of least variation of the disparity. We observe in the same figure that there is little difference between using 5 and 9 window orientations. In

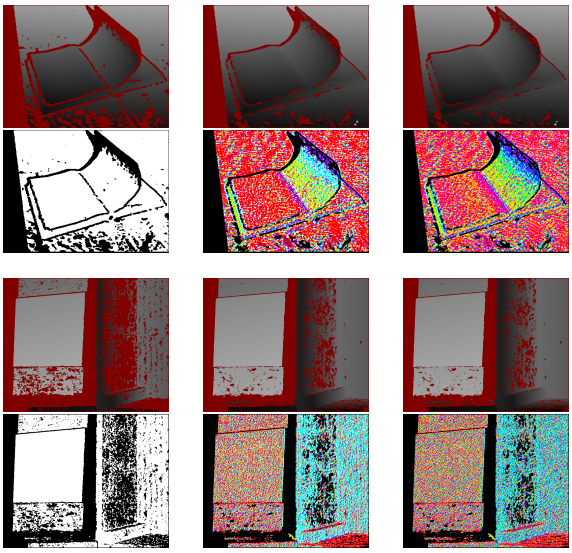


Fig. 9 Comparison of single and multiple window orientations (*book* and *wood* pairs from Figure 11). Using respectively 1, 5 or 9 windows orientations with support equivalent to 5×5 windows, using all filters and exhaustive search with precision of 1/4 of pixel. This figure illustrates how the elongated windows adapt to the surface orientations, thus improving the matching and leading to denser results.

spite of that we will use 9 window orientations in the rest of the paper.

5 Multi-scale algorithm

The range of disparities considered by the block-matching might be fixed globally depending on scene and the camera parameters (extrinsic and intrinsic). The disparity range has a direct impact in the computation time but also affects the ambiguity of the matching. That is, as the range increases so does the probability of matching by chance due to repetitive patterns or noise. So when considering larger search ranges we should use larger patches in order to keep the ambiguities from growing. A patch must contain enough information to match meaningfully, but it should also be small enough to avoid fattening artifacts.

In order to satisfy above requirements a multi-scale approach seems to be the most suitable solution. The proposed multi-scale approach uses the disparity computed at coarser scale in order to restrict the search range at the current scale. This permits to reduce the computation time and to maintain a small window size while retaining meaningfulness of the matches. All rejection criteria are applied at each scale, and the points rejected at the previous (coarser) scale are searched over the entire disparity range at the current scale. While the validated points are assigned a disparity range that depends on the minimum and maximum disparity of validated points in the window. Algorithm 2 details this strategy.

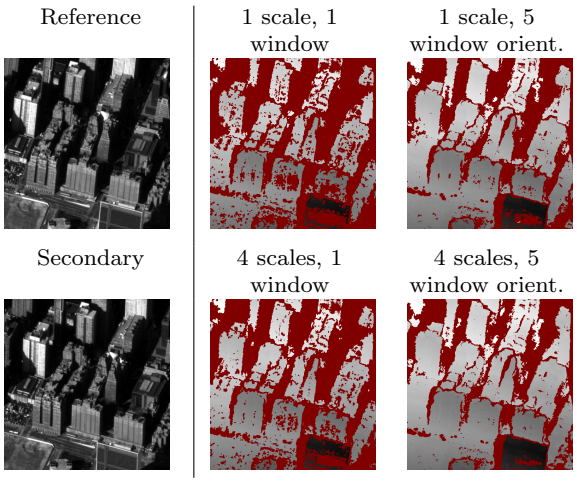


Fig. 10 Interaction between multi-scale and multi-window matching. In the left-most column is shown the image pair. The second column shows results obtained using a 5×5 pixels window with 1 and 4 scales, note that the multi-scale result is denser. The third column shows results obtained using windows with 5 orientations with 1 and 4 scales. With 5 orientations the single scale result is already denser than the one obtained with a single squared window, and the multi-scale result is even denser. The multi-scale algorithm is also faster than the single scale one. With a single window orientation the computation times for the single scale and multi-scale are respectively 149s and 99s (for a 1000×1000 image on a 2.3GHz CPU without multithreading). With the current implementation, the running time of the algorithm with 5 windows roughly increases by a factor 5. However, in addition to parallelization, the algorithm can be accelerated by simultaneously computing all the window orientations using a scheme like the one proposed by Hirschmüller et al. [13].

Figure 10 compares the performance of multi-scale versus single-scale algorithms using windows of 5×5 pixels. It is observed that more pixels are correctly matched with the multi-scale strategy and that the computation time is also significantly reduced.

Each scale of the multi-scale process invokes the multi-window matching algorithm described in the previous section. We denote the resulting algorithm multi-scale multi-window (MSMW). In Figure 10 we see that the advantages of the multi-scale add to those of the multi-window matching: the results are denser and there are fewer errors. Indeed the narrow windows allow to pick up narrow features at coarser scales, thus reducing the corresponding search ranges and considerably improving the matching.

Algorithm 2: Recursive multi-scale multi-window matching algorithm. Initially $dMin$ and $dMax$ are constant images set to the global minimum and maximum ranges and $s = 0$.

$MultiscaleChain(u_0,\ u_1,\ dMin,\ dMax,\ nScales,\ s)$;

Input: image pair u_0 and u_1.
Input: min & max disp. range images $dMin$ and $dMax$.
Input: number of scales $nScales$.
Input: current scale s.

Output: Fixed precision disparity $disp$.
Output: Updated $dMin$ and $dMax$.

if $s < nScales$ **then**
  $su_0 = u_0 \downarrow 2$; $su_1 = u_1 \downarrow 2$;
  $sdMin = dMin \downarrow 2$; $sdMax = dMax \downarrow 2$;
  $sdMin = 0.5 * sdMin$; $sdMin = 0.5 * sdMin$;
  $s++$;
  disp =
  $MultiscaleChain(su_0,\ su_1,\ sdMin,\ sdMax,\ nScales,\ s)$;
  $dMin = sdMin \uparrow 2$; $dMax = sdMax \uparrow 2$;
  $dMin = 2 * dMin$; $dMin = 2 * dMin$;
end if

$disp = MultiWindowMatching(u_0,\ u_1, dMin, dMax)$

for each pixel p **do**
  if p is rejected **then**
    Set $dMin(p)$ and $dMax(p)$ to the global minimum and maximum range.
  end if

  if p is validated **then**
    Set $dMin(p)$ and $dMax(p)$ to the minimum and maximum of validated disparities inside the correlation windows.
  end if
end for

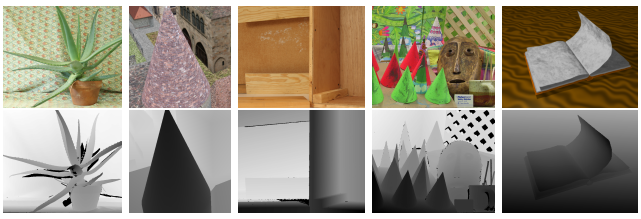


Fig. 11 Reference image and ground truth for stereo pairs used in the experiments section. From left to right the image name, size and disparity range are: *aloe*, 641 × 555, 135; *village2*, 512 × 512, 22; *wood*, 686 × 555, 105 ; *cones*, 900 × 750, 120 and *book*, 400 × 300, 61.

6 Experiments

We compare the results of the proposed multi-scale multi-window algorithm (MSMW) with other two non dense matching methods, namely Čech and Šára's *growing correspondence seeds* (GCS) [7] and the *five regions correlator* (5REG) proposed by Hirschmüller et al. [13] (as implemented in BoofCV [1]). The GCS algorithm,

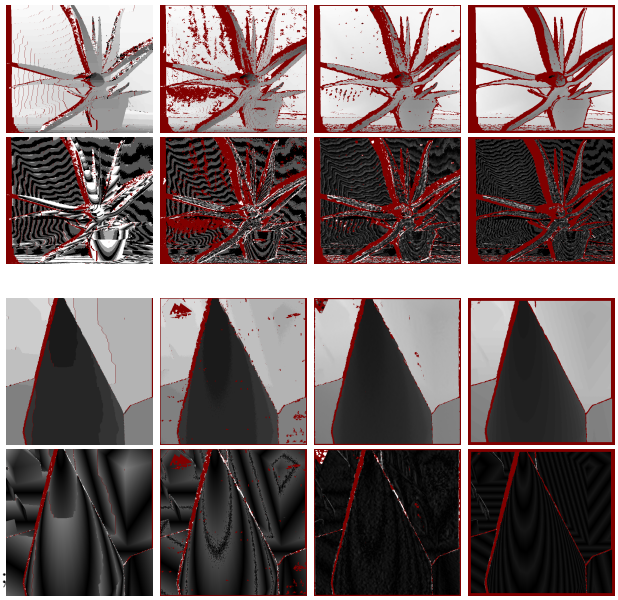


Fig. 12 Top, from left to right: disparity computed by the graph cut method, the GCS, the 5REG and the MSMW. Below: image error between computed disparities and ground truth displayed in range [0, 1.5].

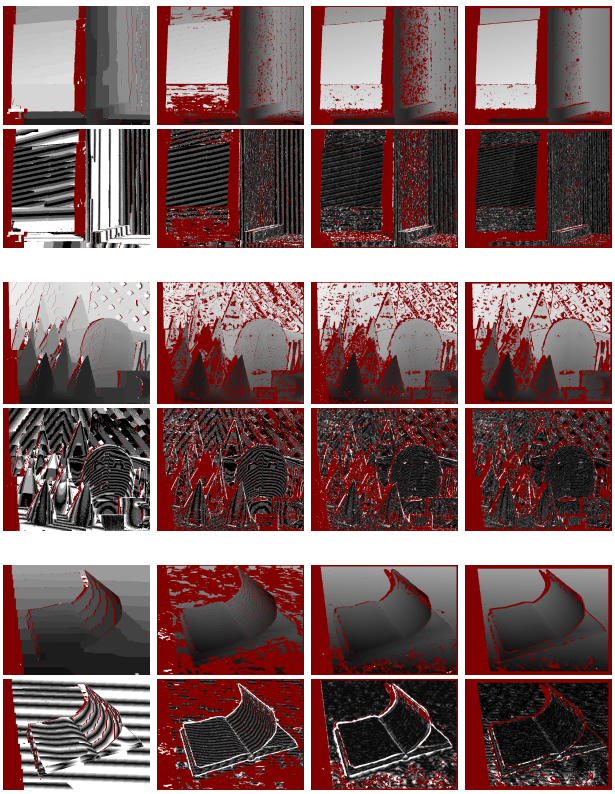


Fig. 13 Top, from left to right: disparity computed by the graph cut method, the GCS, the 5REG and the MSMW. Below: image error between computed disparities and ground truth displayed in range [0, 1.5].

as ours, favors reliability over completeness of the results. The algorithm performs a region growing from "sure" matches selected using Harris interest points. The 5REG algorithm considers five overlapping 5×5 sub-windows for each pixel and the matching cost is determined adding the cost of the center sub-window plus the two sub-windows yielding the smallest costs. The matches are then filtered applying the Left-Right consistency test and comparing the matching cost with the cost of the second local minimum. The 5REG implementation we used [1] does not include the border correction algorithm described by Hirschmüller et al. [13]. For MSMW we use 9 window orientations with areas comparable to a 5×5 pixel window (shown in Figure 8).

We also compare with the *graph cuts* method (GC) introduced in [16]. The algorithm writes as the minimization of a discrete energy containing three terms,

$$E(f) = E_{data}(f) + E_{occ}(f) + E_{smooth}(f),$$

where E_{data} results from the differences in intensity between corresponding pixels, the occlusion term E_{occ} imposes a penalty for making a pixel occluded and the smoothness term E_{smooth} forces neighboring pixels to have similar disparities. Two parameters must be set in the graph cuts method, one for the regularity demanded to the solution and one for the amount of occlusions to be detected. By default these parameters are tuned automatically.

Figures 12 - 14 compare the performance of the four methods mentioned above for different images with ground truth. The computed disparities and the errors with respect to the ground truth are displayed. The reference images and ground truth [14, 20, 22, 25, 24] are shown in Figure 11. Both, the GCS and 5REG algorithms produce fairly dense results. However, they contain many spurious matches and false detections. The disparity computed by the MSMW algorithm is dense and contains fewer errors. The improved rejection criteria significantly reduce the mismatches while the multi-scale scheme permits to obtain denser results by disambiguating some of the matches. Since GCS and 5REG do not manage fattening many errors are located around the discontinuities. The graph cut algorithm fails at detecting the occlusions in most of the examples. We also observe an over-quantization of the disparity computed by graph cut, which has disparity steps of two pixels instead of one. This is due to the fact that in the regularity term discontinuities of disparity are penalized independently of its amplitude, see [16] for more details.

The methods compared here produce disparity maps with different quantizations. GC and GCS operate at pixel precision, 5REG interpolates the disparities to subpixel precision, while MSMW compute the maps at quarter pixel. Instead of normalizing the results to pixel precision we illustrate each method at its native precision (Figures 12 - 14). The evaluation below will contemplate different precisions to cope with this variability.

To compare the algorithms we evaluate the density of the disparity maps and the number of mismatches produced by each method. Note that a density of 100% cannot be attained unless the disparity maps are interpolated. A mismatch is a point validated by the algorithm but whose disparity differs from the ground truth by more than a certain threshold. We don't distinguishing between occluded and non-occluded pixels when computing the mismatches. We consider here three mismatch thresholds: 0.5 pixels, 1 pixel and 2 pixels.

Table 1 summarizes the obtained results. It is observed that the proposed algorithm attains the lowest mismatch rate of the four, while keeping a relatively high density. Beside the GC method which is a global, the highest densities are attained 5REG and MSMW with small differences between the two. The 5REG method yields high densities but at the expense of higher mismatch rates. This can be explained by the filter using the second local minimum of the correlation function. The test checks if the cost of the best and the second best match verify $\frac{c_2 - c_1}{c_1} > 0.2$, but this default choice of the threshold allows many mismatches. Increasing it may reduce the mismatch rate but at the expense of a much lower density. The graph cuts method generates dense disparity maps. Although it also detects occlusions, we see in Table 1 that its mismatch rates are consistently one order of magnitude above the other methods (except for the Tsukuba image [24]).

Lastly, Figure 14 compares the performance of the four algorithms applied to a couple of stereo pairs kindly provided to us by the CNES *(Centre national d'études spatiales)*. The first one is an airborne pair taken over Toulouse, while the second is a view of Manhattan acquired by a *Pléiades satellite*. The unique agility of the Pleiades satellites allow it to capture small-baseline pairs in a single passage.

This type of high-resolution quasi-simultaneous imagery is leading to an increase in the demand of automatically generated and reliable digital elevation models with reduced number of outliers. The automatic construction of high quality digital elevation models justifies our interest on obtaining sparse and reliable disparity maps. In the figure we observe that among the considered algorithms the proposed one produces the least outliers, while producing fairly dense results.

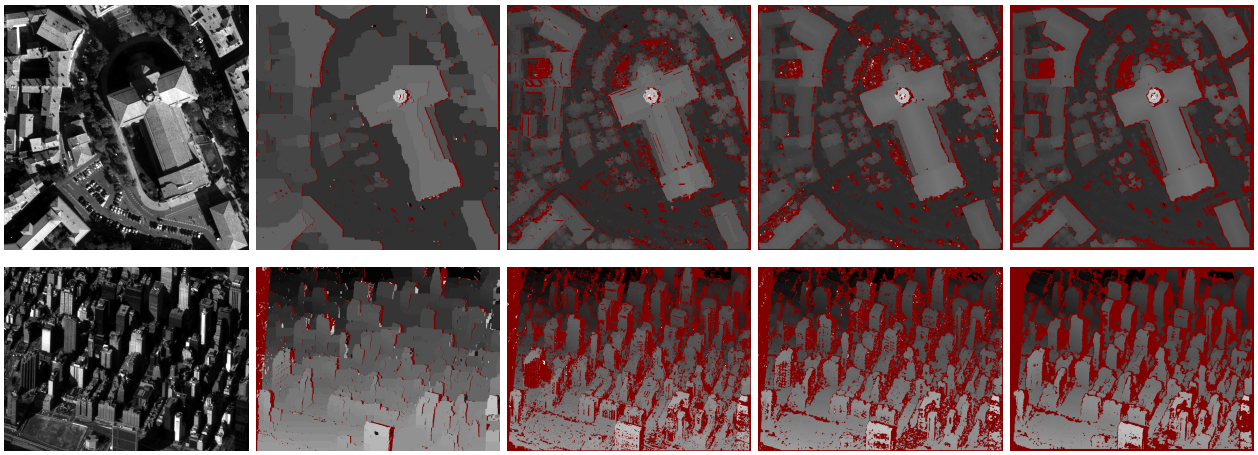
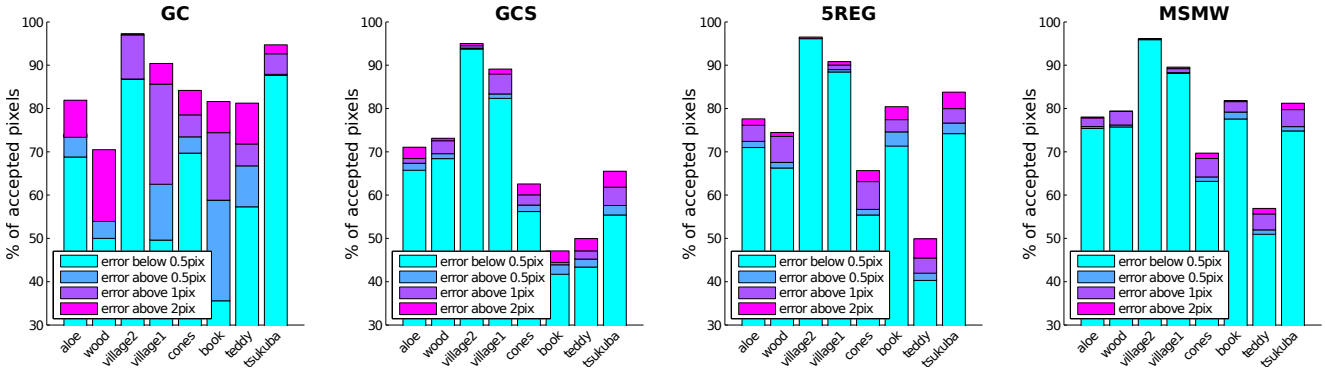


Fig. 14 Experiments on airborne and satellite stereo pairs kindly provided by the CNES *(Centre national d'études spatiales)*. From left to right are shown: the reference image, the disparities computed by the graph cut method, GCS, 5REG and MSMW. The first row corresponds to an airborne pair taken over Toulouse (size 900 × 900 pixels), the second row corresponds to a satellite view of Manhattan (size 1200 × 900 pixels).

Table 1 Evaluation of the results. The table below compares the outputs of the proposed algorithm (MSMW) with: the graph cuts algorithm (GC) from Kolmogorov and Zabih [16], Čech and Šára's [7] growing correspondence seeds (GCS) and the 5 regions algorithm (5REG) from Hirschmüller et al. [13]. We report the Density (D) of the obtained disparity maps and the Mismatches as percentage of pixels yielding errors above two pixels (M2), above one pixel (M1) and above 0.5 pixels (M05).

	GC [16]				GCS [7]				5REG [13]				MSMW			
Image	**D**	**M2**	**M1**	**M05**	**D**	**M2**	**M1**	**M05**	**D**	**M2**	**M1**	**M05**	**D**	**M2**	**M1**	**M05**
aloe	**90.5**	8.58	13.9	21.7	73.7	2.63	4.25	7.97	**79.1**	1.49	2.91	8.12	78.3	**0.26**	**0.68**	**2.87**
wood	**87.1**	16.6	25.3	37.1	73.7	0.57	1.72	5.27	75.4	0.92	2.21	9.15	**79.5**	**0.09**	**0.53**	**3.75**
village2	**97.5**	0.25	0.28	10.7	95.5	0.49	0.70	1.77	**96.9**	0.39	0.42	0.72	96.1	**0.01**	**0.02**	**0.19**
village1	**95.2**	4.79	17.7	45.6	90.3	1.18	2.22	7.97	**91.6**	0.78	1.33	3.17	89.9	**0.35**	**0.53**	**1.78**
cones	**89.8**	5.63	9.39	20.1	65.1	2.52	4.00	8.89	68.2	2.54	3.83	12.8	**70.9**	**1.22**	**2.22**	**7.72**
book	**88.8**	7.19	30.4	53.2	49.8	2.68	4.88	8.10	**83.4**	2.98	6.27	12.1	82.1	**0.25**	**1.85**	**4.52**
teddy	**90.7**	9.45	18.9	33.4	52.8	2.84	4.67	9.41	54.3	4.41	6.06	14.0	**58.2**	**1.27**	**2.29**	**7.25**
tsukuba	**96.8**	2.08	**2.33**	9.15	69.2	3.69	5.89	13.8	**87.6**	3.81	6.25	13.4	82.7	**1.47**	2.47	**7.88**



7 Discussion and conclusions

We have proposed a principled block-matching algorithm for stereo. Our work builds upon existing ideas of match validation to obtain a virtually parameterless algorithm for non dense block matching. We highlight the importance of detecting ambiguous matches, and propose two new match validation filters: the self-similarity and min-diff filters. We singled out the fronto-parallel assumption as a source of mismatches and proposed a new scheme with oriented windows to handle it. The oriented windows align with the 3D geometry of the scene and unlike adaptive windows permit to match correctly on slanted surfaces. The multi-scale and multi-window features of the proposed algorithm permit to deal with slanted surfaces and outliers, leading to higher matching densities. Compared to other stereo algorithms, the proposed algorithm yields excellent performances in terms of density and mismatch rates.

The natural continuation of the present work should, on one hand, improve the performance of parameter-

less rejection criteria, for example with *min-diff*, or the detection of dis-occlusions. On the other hand, should seek ways to improve the matching process. Exploring the combination of information from windows of various sizes. Or combining oriented windows with adaptive windows, which seems to be the best way to adequately handle the scene geometry. In the present work we only considered images with low noise levels, which allowed us to fix the window size, but the noise level is critical to establish the window size. These will be the object of future research.

Acknowledgements We thank CNES for providing the images in Figure 14.

References

1. Abeles, P.: Boofcv. http://boofcv.org/ (2012)
2. Arnold, R.: Automated stereo perception. Tech. rep., DTIC Document (1983)
3. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(4), 401–406 (1998). DOI http://dx.doi.org/10.1109/34.677269
4. Blanchet, G., Buades, A., Coll, B., Morel, J., Rouge, B.: Fattening free block matching. Journal of Mathematical Imaging and Vision pp. 1–13 (2011)
5. Bleyer, M., Rhemann, C., Rother, C.: PatchMatch stereo - stereo matching with slanted support windows. In: Procedings of the British Machine Vision Conference 2011, pp. 14.1–14.11. British Machine Vision Association (2011). DOI 10.5244/c.25.14
6. Bobick, A., Intille, S.: Large occlusion stereo. International Journal of Computer Vision **33**, 181–200 (1999)
7. Cech, J., Šára, R.: Efficient Sampling of Disparity Space for Fast And Accurate Matching. In: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, pp. 1–8. IEEE (2007). DOI 10.1109/CVPR.2007.383355
8. Cochran, S., Medioni, G.: 3-D surface description from binocular stereo. Pattern Analysis and Machine Intelligence, IEEE Transactions on **14**(10), 981–994 (1992). DOI 10.1109/34.159902
9. Fua, P.: A parallel stereo algorithm that produces dense depth maps and preserves image features. Machine Vision and Applications **6**(1), 35–49–49 (1993). DOI 10.1007/BF01212430
10. Fusiello, A., Roberto, V., Trucco, E.: Symmetric stereo with multiple windowing. International Journal of Pattern Recognition and Artificial Intelligence **14**(8), 1053–1066 (2000)
11. Gruen, A.W.: Adaptive least squares correlation: a powerful image matching technique. South African Journal of Photogrammetry, Remote Sensing and Cartography **14**(3), 175–187 (1985)
12. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press, ISBN: 0521540518 (2004)
13. Hirschmüller, H., Innocent, P., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. International Journal of Computer Vision **47**(1-3), 229–246 (2002)
14. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(9), 1582–1599 (2009). DOI 10.1109/TPAMI.2008.221
15. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiment. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(9), 920–932 (1994)
16. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 2, pp. 508–515. IEEE (2001)
17. Lotti, J., Giraudon, G.: Correlation algorithm with adaptive window for aerial image in stereo vision. In Image and Signal Processing for Remote Sensing **1**, 2315–10 (1994)
18. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
19. Manduchi, R., Tomasi, C.: Distinctiveness maps for image matching. Proceedings of the International Conference on Image Analysis and Processing pp. 26–31 (1999)
20. Moisan, L.: Paires stéréo simulées. personal communication (2010)
21. Patricio, M., Cabestaing, F., Colot, O., Bonnet, P.: A similarity-based adaptive neighborhood method for correlation-based stereo matching. In: International Conference on Image Processing, vol. 2, pp. 1341–1344 (2004)
22. Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.: Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In: Computer Vision–ECCV 2010, pp. 510–523. Springer (2010)
23. Sabater, N., Morel, J., Almansa, A.: How Accurate Can Block Matches Be in Stereo Vision? SIAM Journal on Imaging Sciences **4**(1), 472–500 (2011). DOI 10.1137/100797849
24. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47**(1), 7–42 (2002)
25. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition, CVPR'03, pp. 195–202. IEEE Computer Society, Washington, DC, USA (2003)
26. Szeliski, R., Scharstein, D.: Sampling the disparity space image. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(3), 419–425 (2004)
27. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the Sixth International Conference on Computer Vision, vol. 846. Citeseer (1998)
28. Veksler, O.: Fast variable window for stereo correspondence using integral images. IEEE Computer Society Conference on Computer Vision and Pattern Recognition **1**, 556–561 (2003). DOI 10.1109/CVPR.2003.1211403
29. Wang, L., Liao, M., Gong, M., Yang, R., Nister, D.: High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission, pp. 798–805 (2006)
30. Yaroslavsky, L., Eden, M.: Fundamentals of digital optics (2003)
31. Yoon, K., Kweon, S.: Adaptive support-weight approach for correspondence search. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(4), 650–656 (2006). DOI 10.1109/TPAMI.2006.70